ACCIDENT SEVERITY PREDICTION OF ZONGULDAK DISTRICT UNDERGROUND COAL MINES BY MACHINE LEARNING TECHNIQUES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MERVE ÖZDEMİR AYDIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MINING ENGINEERING

AUGUST 2022

Approval of the thesis:

ACCIDENT SEVERITY PREDICTION OF ZONGULDAK DISTRICT UNDERGROUND COAL MINES BY MACHINE LEARNING TECHNIQUES

submitted by **MERVE ÖZDEMİR AYDIN** in partial fulfillment of the requirements for the degree of **Master of Science** i**n Mining Engineering, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Naci Emre Altun
Head of the Department, **Mining Engineering, METU** ________________

Prof. Dr. Celal Karpuz
Supervisor, **Mining Engineering, METU** ________________

Examining Committee Members:

Assoc.Prof. Dr. Mehmet Ali Hindistan
Mining Engineering Department, Hacettepe University ________________

Prof. Dr. Celal Karpuz
Mining Engineering Department, METU ________________

Assoc. Prof. Dr. Mustafa Erkayaoğlu
Mining Engineering Department, METU ________________

Date: 15.08.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Last name : Merve Özdemir Aydın

Signature :

ABSTRACT

ACCIDENT SEVERITY PREDICTION OF ZONGULDAK DISTRICT UNDERGROUND COAL MINES BY MACHINE LEARNING TECHNIQUES

Özdemir Aydın, Merve
Master of Science, Mining Engineering
Supervisor : Prof. Dr. Celal Karpuz

August 2022, 116 pages

Underground coal mining is considered among the most dangerous sectors in the world due to the accidents. Thus, this study aims to build an accident severity prediction model for underground coal mines by using decision tree, support vector machine, and neural network algorithms. Defining the severity of accidents will provide an effective way of preventing risks that will cause serious accidents. This study also aims to fill the gap in the literature related to designing accident severity prediction models for underground coal mining for safety management.

In the study, 8406 underground accident data covering two years period of time, and eleven variables (dimensions), which are shift, day of the accident, job, education, type of accident, reason of the accident, location of the accident, severity of the accident, age, seniority, affected body part, collected by the Turkish Hard Coal Enterprise of Amasra, Armutçuk, Karadon, Kozlu, and Üzülmez district were used to build an accident severity prediction model. Before applying the machine learning algorithms, principal component analysis was applied to reduce the dimensions and express the data with fewer variables that are meaningful and easier to explain. Principal component analysis provided that 81.82% (cumulative variance percent) of the data could be interpreted with the seven components. By using these seven variables, accident severity prediction models were built applying decision tree, support vector machine, and neural network algorithms. The decision tree model has the accuracy 78.5%, support vector machine model has the accuracy 79.2%, and neural network model has the accuracy 78.5%. As a result, it was decided that the accident severity estimation model that gives the most accurate prediction results is the support vector machines for this data set. Based on trained prediction model results, the dominant correct classification accident severity type is slightly injured.

Keywords: Underground Coal Mine, Principal Component Analysis, Decision Tree, Support Vector Machine, Neural Network

ÖZ

ZONGULDAK BÖLGESİ YERALTI KÖMÜR MADENLERİNİN MAKİNE ÖĞRENMESİ TEKNİKLERİ İLE KAZA ŞİDDETİ TAHMİNİ

Özdemir Aydın, Merve
Yüksek Lisans, Maden Mühendisliği
Tez Yöneticisi: Prof. Dr. Celal Karpuz

Ağustos 2022, 116 sayfa

Yer altı kömür madenciliği, kazalar nedeniyle dünyanın en tehlikeli sektörleri arasında yer almaktadır. Bu nedenle, bu çalışma karar ağacı, destek vektör makinesi ve sinir ağı algoritmalarını kullanarak yeraltı kömür madenleri için bir kaza şiddeti tahmin modeli oluşturmayı amaçlamaktadır. Kazaların ciddiyetinin tanımlanması, ciddi kazalara neden olacak risklerin önlenmesinde etkili bir yol sağlayacaktır. Bu çalışma aynı zamanda iş güvenliği yönetimi için yeraltı kömür madenleri için kaza şiddeti tahmin modellerinin tasarlanması ile ilgili literatürdeki boşluğu doldurmayı amaçlamaktadır.

Bu çalışmada, kaza şiddeti tahmin modeli oluşturmak amacıyla Turkiye Taş Kömürü İşletmesi'nin Amasra, Armutçuk, Karadon, Kozlu, Üzülmez bölgelerine ait vardiya, kaza günü, meslek, eğitim, kazanın türü, kazanın nedeni, kazanın yeri, kazanın şiddeti, yaş, kıdem, etkilenen vücut bölümü gibi bilgilerden oluşan 11 değişken ve iki yıllık zaman dilimini kapsayan 8406 adet yeraltı kaza verisi kullanılmıştır. Makine öğrenimi algoritmalarını uygulamadan önce, boyutları azaltmak ve verileri anlamlı ve açıklanması daha kolay olan daha az değişkenle ifade etmek için temel bileşenler analizi uygulanmıştır. Temel bileşenler analizi verilerin %81,82'inin (kümülatif varyans yüzdesi) yedi bileşenle yorumlanabileceği sonucunu sağlamıştır. Bu yedi değişken kullanılarak, karar ağacı, destek vektör makinesi ve sinir ağı algoritmaları uygulanarak tahmin modelleri oluşturulmuştur. Karar ağacı modeli %78,5, destek vektör makine modeli %79,2 ve sinir ağı modeli %78,5 doğruluğa sahiptir. Sonuç olarak, en doğru tahmin sonuçlarını veren kaza şiddeti tahmin modelinin bu veri seti için destek vektör makineleri olduğuna karar verilmiştir. Eğitilmiş tahmin modeli sonuçlarına göre, baskın olan doğru sınıflandırılmış kaza şiddeti hafif yaralanmadır.

Anahtar Kelimeler: Yeraltı Kömürü Madeni, Temel Bileşenler Analizi, Karar Ağacı, Destek Vektör Makinesi, Sinir Ağı

To my beloved husband ~Mehmet Fatih Aydın~

ACKNOWLEDGMENTS

First, I would like to express my sincere gratitude to my thesis advisor Prof. Dr. Celal Karpuz for his valuable discussion, encouragement, support, and guidance on this research. I also present my special thanks to the examining committee members, Assoc. Prof. Dr. Mustafa Erkayaoğlu and Assoc. Prof. Dr. Mehmet Ali Hindistan for serving on the M.Sc. thesis committee.

I would also like to thank special thanks to Bayram Arı (Vice President of General Directorate of Mining and Petroleum Affairs), İsmail Onsekiz (Head of Department of Tender and Project Development), Leyla Sipahioğlu (Coordinator of Department of Tender and Project Development) and my workmates for their supports during my M.Sc. program.

I owe great thanks to my extended family Belma Engin, Ayşenur Bağcılar, Sezer Bağcılar, Belma Özkan, Umut Özkan, and Mehmet Emre Özdemirhan for their support, and absolute belief in me.

I would like to express my sincere gratitude to my mother İlkin Özdemir, my sister Hümeyra Kolancı and rest of my family for their endless love, encouragements and support for my whole life.

Finally and most importantly, I would like to special thank my dearest husband Mehmet Fatih Aydın, for his patience and continuous support. Your pep-talks gave me the strength every time. Thank you for being in my life and by my side.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	
THCE	Turkish Hard Coal Enterprise
PCA	Principal Component Analysis
DT	Decision Tree
SVM	Support Vector Machine
NN	Neural Network
ILO	International Labour Organization
SSI	Social Security Institution

LIST OF SYMBOLS

SYMBOLS

p	number of variables
n	number of samples
x_{ij}	measured data
$\overline{X_{\iota\jmath}}$	mean value of data
δ_j	the standard deviation
λ	eigenvalue
v	eigenvector
R	correlation matrix
r_{jk}	correlation coefficient between x_j and x_k
p_j	probability of class j in a node
I	the identity matrix

CHAPTER 1

INTRODUCTION

1.1 General Remarks

Energy, as one of the most important inputs to economic growth and industrialization, is a necessary component of modern life. Therefore, the demand for energy in the global markets is constantly increasing. Coal, one of the fossil energy sources, is an essential source of energy due to its widespread presence in the world, its production and the presence of visible coal reserves in terms of price stability compared to other fossil fuels.

According to the information provided by Republic of Turkey Ministry of Energy and Natural Resources (2022), a total of 116.7 million tonnes of coal were consumed in Turkiye in 2021, including 37.3 million tonnes of hard coal, 73.6 million tonnes of lignite and asphaltite, and 5.8 million tonnes of hard coal coke. Moreover, Turkiye's average coal (hard coal, hard coal coke, lignite, and asphaltite) consumption between 2016 and 2021 was approximately 110 million tonnes. Figure 1.1 represents Turkiye's coal consumption by years.

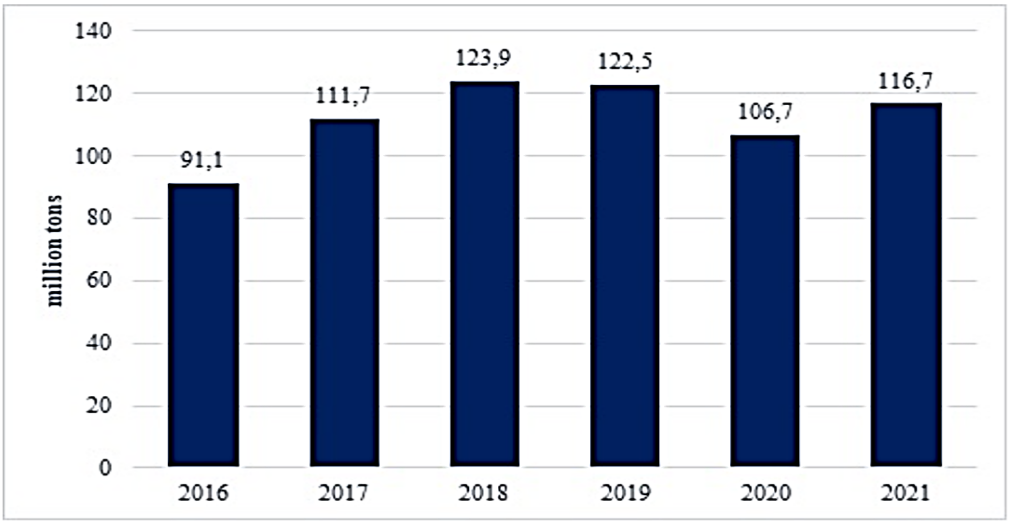


Figure 1.1 Turkiye's Coal Consumption Amounts by Years ("Coal", 2022)

The largest share in the consumption of hard coal, lignite-asphaltite belonged to thermal power plants with 52,9% and 81,7%, respectively. As of March 2022, in Turkiye, there are a total of 67 coal-fired power plants, including 1 asphaltite, 47 lignite, 4 hard coal and 15 imported coal-fired power plants ("Coal", 2022).

In parallel with the rise of the world's population and living standards, these consumption rates are also increasing. Moreover, the number of employees is increasing with increased consumption and increased production to reduce foreign dependency, and this makes occupational health and safety more important.

1.2 Problem Statement

Coal mining has many risks, whether surface or underground, making it exceptional in the area of occupational health and safety. The main activity in which occupational health and safety problems arise is the production process. The production process covers the main activities such as excavation, support, transportation and other activities such as establishment of electricity, compressed air networks, installation, operation, communication and signaling systems, and maintenance and repair of various machinery and equipment. During these processes, health and safety problems arise from both the nature of the work and the specific conditions of the mining activities. Thus, it is essential to precisely predict the severity of work-related accidents for safety management and control.

1.3 Scope and Objectives of the Study

The scope of this study is 8406 underground accident data belonging to Turkish Hard Coal Enterprise Amasra, Armutçuk, Karadon, Kozlu, Üzülmez district and covers the period of March 2008 and December 2010. The main objective is to build an accident severity prediction model for underground coal mines. There were eleven variables (dimensions) in the accident data set. Thus, the first aim was dimension reduction while preserving as much information as possible. During the literature survey, it was seen that there were not many comparative studies on which type of analysis would be better to analyze coal mine accident data and predict the severity of the accident. Thus, comparing the decision tree, support vector machine, and neural network algorithms, it was aimed to find the optimum model for accident severity prediction model for underground coal mines.

1.4 Research Methodology

The research methodology consists of the steps summarized as:

- Literature research,
- Preparation of data set for the algorithms,
- Analyzing each variable by using Microsoft Excel,
- Applying principal component analysis for dimension reduction by using R Studio,
- Building a prediction model by using decision tree algorithm by using MATLAB,
- Building a prediction model by using support vector machine algorithm by using MATLAB,

- Building a prediction model by using neural network algorithm by using MATLAB,
- Comparing the prediction models.

1.5 Outline of the Thesis

This thesis study includes five chapters and three appendices. Chapter 1 provides general information about the thesis covering general remarks, problem statement, scope and objectives of the study, research methodology, and outline of the study.

The literature survey of the study is presented in Chapter 2. In this section, an introduction, a general overview of applied methods, and previous accident forecasting studies in underground coal mines are explained.

Chapter 3 outlines the study area and the data set. In this chapter, a brief information is given about the study area and the variables in the data set are explained.

Chapter 4 covers analyzing data and building the models. The models created using the principal component analysis results, decision tree, support vector machine, and neutral network algorithms are described in detail in this section. Models with the highest accuracy are obtained by changing parameters for each algorithm.

The results of the principal component analysis and the trained prediction models and discussions are stated in Chapter 5.

Finally, the thesis ends with main outcomes of this study and some recommendations for future researches are presented in Chapter 6.

Appendix A presents the codes for principal component analyses. The scatter plots with respect to predictors are presented in Appendix B. Appendix C and Appendix D provides the summaries and the validation confusion matrix of the trained models, respectively.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

Occupational accidents that are encountered all over the world are a major problem that affects the entire community both financially and spiritually. According to the most recent global estimates from 2017, 2.78 million employees die each year as a result of work-related accidents and diseases, while hundreds of millions more suffer from non-fatal work-related injuries and diseases that are both temporary and permanent (ILO, 2022). These work-related deaths and diseases vary from sector to sector. For a long time, the mining industry has been regarded as one of the most dangerous industries in the world, with enormous health and safety risks (Lööw and Nygren, 2019). The data of the Social Security Institution in Turkiye supports this view. According to the Social Security Institution data in Turkiye, between 2010 and 2019, 2,360,472 insured occupational accidents resulted in 13,852 fatalities. In the same period, 115,950 insured workers had a work-related accident in the mining industry. A total of 1,042 miners have lost their lives as a result of these work-related accidents. According to these statistics, the mining industry accounts for 4.91 percent of all insured business accidents and 7.52 percent of all fatal accidents. Given that the average unregistered employment rate in Turkiye for the mining and quarrying activity code is 6.62 percent from 2010 to 2019, these percentages are lower than the actual value (SSI, 2020).

The rates of death due to work-related accidents in the activities of the mining sector are dominated by coal extraction activities. Between 2010 and 2020, according to SSI data, 55 percent of the deaths due to work-related accidents in the mining sector occurred in coal mines.

There is also a huge difference between hard coal mines and lignite mines when looking at the distribution of death rates due to work-related accidents in coal mines. In Figure 2.1, the number of deaths per million tonnes of hard coal and lignite in Turkiye are presented. It is seen that there are also more deaths in hard coal mines than in lignite mines per million tonnes of production.

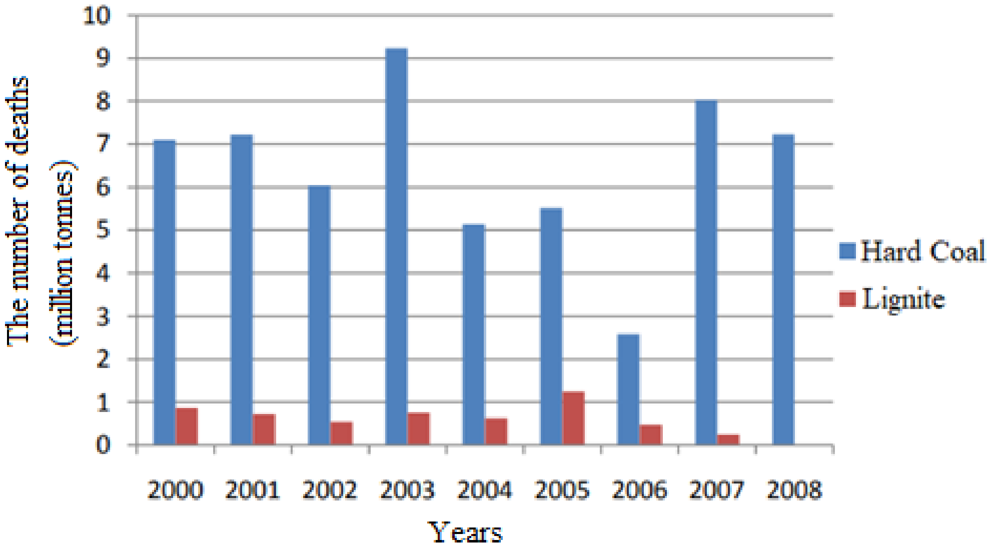


Figure 2.1 The number of deaths per million tonnes of hard coal and lignite in Turkiye (Arslanhan & Cunedioglu, 2010)

Turkiye ranks higher in the world rankings with these death rates in coal mines. Accidents in the United States and China, two of the world's largest coal producers, show that mortality rates are lower than in Turkiye (Arslanhan & Cunedioglu, 2010). Given the death number per million tonnes of coal production for 2008 in Table 2.1, Turkiye's death number is 5.7 times that of China, which is the world’s top producer, and 361 times that of the United States. According to the study by Arslanhan and Cunedioglu (2010), significant drops in death numbers have been achieved with the renewal of technology in coal mines in the United States, especially in the 1970s, and the reconstruction of mines in China, especially in 2004. However, the worst-case scenario for Turkiye continues. Because, according to the statistics of Social Security Institution in 2020, 36442 workers in 448 workplaces in Turkiye work in the coal and lignite industries, and 23 of the 100 workers working in the coal and lignite industries had a work-related accident in 2020.

Table 2.1 The number of deaths per hard coal million tonnes (Arslanhan & Cunedioglu, 2010)

Year	Turkiye	China	U. S
2000	7.10	4.08	0.03
2001	7.22	4.11	0.02
2002	6.04	3.98	0.04
2003	9.23	4.06	0.04
2004	5.14	3.03	0.03
2005	5.51	2.72	0.01
2006	2.59	2.00	0.06
2007	8.02	1.50	0.04
2008	7.22	1.27	0.02

Turkiye needs more policies about occupational health and safety to change this worst-case scenario. To develop policies about occupational health and safety, timely, relevant, and accurate data and statistics are essential. Moreover, comprehensive and high-quality statistics and data analysis are necessary to support decision-making and inform the development of policies for improving occupational safety and health and to evaluate their effectiveness in reducing and preventing occupational accidents.

2.2 General Overview of Applied Methods

In this study, principal component analysis (PCA), decision tree (DT), support vector machine (SVM), and neural network (NN) methods were used to estimate the severity of mining accidents and their performances were compared. The following sections provide a literature review on properties of these machine learning algorithms.

2.2.1 Principal Component Analysis

In data science studies, it may be necessary to work with a large number of variables. This situation, excessive training time, brings along various problems such as overfitting and multicollinearity. The prepared models will need to work in optimum time and with optimum performance. To overcome these problems, dimensionality reduction methods can be used. In dimensionality reduction, the number of variables is reduced by creating new variables that are a combination of existing variables. Thus, all the features in the dataset are somehow still present, but the number of variables is reduced.

One of the most frequently used multivariate data analysis and dimensionality reduction techniques is principal component analysis, which is also called “Hotteling transform” or “Karhunen-leove (KL) Method” (Chandra Paul et al., 2013).

The main purpose of this analysis is that it is based on the identification of new variables with fewer independent linear components that allow them to decompose without loss of information.

Principal component analysis (PCA) has become a basic tool in modern data analysis since it is a simple, non-parametric method for extracting meaningful information from complicated data sets and provides a simple method for reducing a complex data set to a lower dimension and revealing the sometimes hidden, simplified structure that lies beneath it (Shlens, 2014).

Moreover, PCA is also a size reduction method that calculates the least number of non-correlated variables from highly correlated data. It is an orthogonal projection of data onto lower-dimension linear space that maximizes variance of projected data (purple line, Figure 2.2), minimizes mean squared distance between data points and their projections (the blue segments, Figure 2.2) (R. Greiner & B. Póczos, 2009). PCA finds the best “subspace” that captures as much data variance as possible. Figure 2.2 shows that two-dimensional data $x = [x_1, x_2]^T$ projected onto a one-dimensional linear manifold (affine subspace) with direction u1 (principal component). Red points are the original data and green points are the projected data.

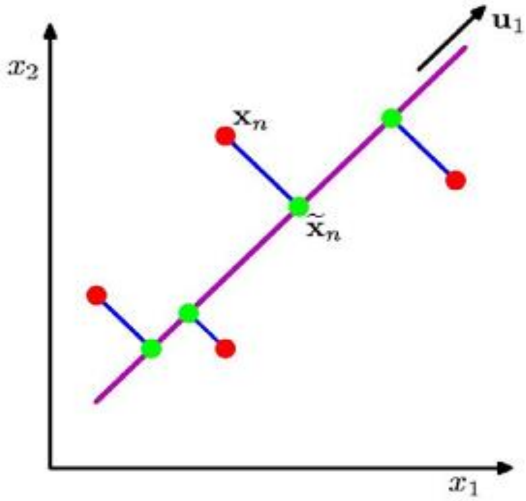


Figure 2.2 Orthogonal projection of data onto lower-dimension (Bishop, 2006)

As Greiner and Póczos point out, vectors are originated from the center of mass. First principal component points in the direction of the largest variance. Each subsequent principal component is orthogonal to the previous ones, and points in the directions of the largest variance of the residual subspace (2009). That is, PCA centers the data at the origin and rotates the axes (Alpaydın, 2010).

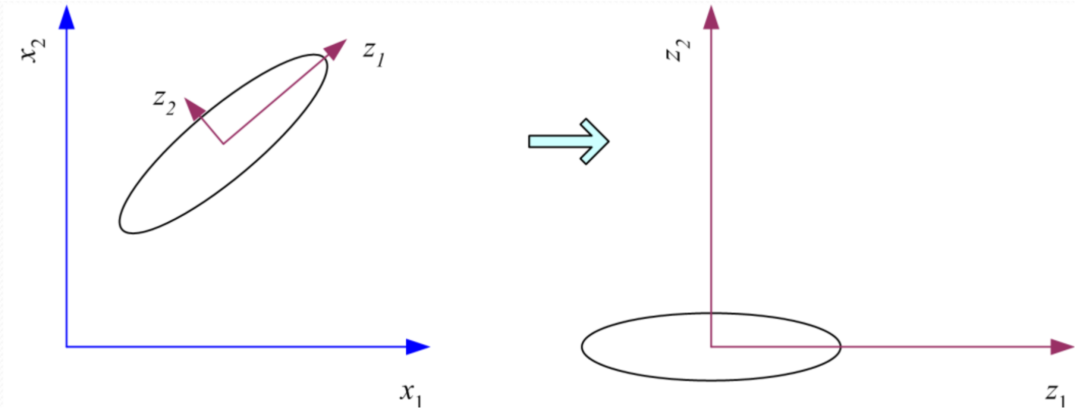


Figure 2.3 Rotation the axes (Alpaydın, 2010)

The steps to follow for principal component analysis are normalizing the data, calculating the correlation matrix, eigenvalues and eigenvectors, choosing components and forming a feature vector, forming principal components.

Normalizing the data

It makes direct comparison unusable if different columns of numeric data have very different ranges. Normalization is a method of bringing all data into a comparable range to make comparisons more meaningful. After normalization, all data in the matrix are within the same range where the mean is zero and the variance is one.

Normalization (standardization) is done by subtracting the respective means from the numbers in the respective column, and then dividing them by standard deviation. The equation of normalization is presented in Equation 3.1, where p is the columns, n is the rows, $\overline{x_j}$ is the average of jth element (Equation 3.2), and δ_j is the standard deviation, xij is the measured data, i is the index for the variable i = 1,2 …, n, and j is the index for the sample number and j = 1,2, …, p.

$$X_s = \begin{pmatrix} (x_{11} - \overline{x_1})/\delta_1 & \cdots & (x_{1p} - \overline{x_p})/\delta_p \\ \vdots & \ddots & \vdots \\ (x_{n1} - \overline{x_1})/\delta_1 & \cdots & (x_{np} - \overline{x_p})/\delta_p \end{pmatrix} \tag{3.1}$$

where

$$\overline{x_j} = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \text{ , } \forall j \tag{3.2}$$

$$\delta_j = \sqrt[2]{\frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \overline{x_j})^2} \text{ , } \forall j \tag{3.3}$$

Calculating the correlation matrix

The covariance provides information about how a variable change with other variables and is always measured between two dimensions (Konak, 2006). If the dataset has 2-dimensions, this will result in a 2x2 covariance matrix (Equation 3.4). In Equation 3.4, var and cov correspond to variance and covariance, respectively.

$$Matrix(Covariance)\begin{bmatrix} Var[X_1] & Cov[X_1, X_2 \\ Cov[X_2, X_1] & Var[X_2] \end{bmatrix} \quad (3.4)$$

If the value of one of the variables increases while the value of the other increases, or if one decreases and the other decreases, the covariance value between the two variables is positive. If the value of one of the variables increases and the value of the other decreases or the value of one decreases and the value of the other increases, the covariance value becomes negative. If there is no relation between variables, the covariance value is zero (Alpar, 2003).

When the data has different scales job, education, and others, the correlation matrix should be used since the variables are standardized by their standard deviation so the total variance is equal to one. In other words, the use of the correlation matrix is equivalent to standardizing each of the variables. Correlation matrices are calculated by using the Equations 3.5 and 3.6 (Atalay, 2019).

$$R = \frac{1}{n-1} X_s^T X_s = \begin{pmatrix} 1 & \cdots & r_{1p} \\ \vdots & \ddots & \vdots \\ r_{p1} & \cdots & 1 \end{pmatrix} \quad (3.5)$$

$$r_{ij} = \frac{\delta_{jk}}{\delta_j \delta_k} = \frac{\sum_{i=1}^{n}(x_{ij} - \overline{x_j})(x_{ik} - \overline{x_k})}{\sqrt{\sum_{i=1}^{n}(x_{ij} - \overline{x_j})^2}\sqrt{\sum_{i=1}^{n}(x_{ik} - \overline{x_k})^2}}, \forall i, j \quad (3.6)$$

where r_{ij} is the correlation coefficient between x_j and x_k.

Calculating the eigenvalues and eigenvectors

The directions of the axis with the largest variance, which we call principal components, are the eigenvectors of the covariance matrix. And eigenvalues are simply the coefficients attached to eigenvectors, which give the amount of variance carried in each principal component. By ranking the eigenvectors in order of their eigenvalues, highest to lowest, the principal components in order of significance has been obtained.

The eigenvalues and eigenvectors are calculated from the covariance matrix. λ is an eigenvalue for a matrix A if it is a solution of the characteristic equation:

$$\det(\lambda I - A) = 0 \tag{3.7}$$

Where, I is the identity matrix of the same dimension as A which is a required condition for the matrix subtraction as well in this case and 'det' is the determinant of the matrix. For each eigenvalue λ, a corresponding eigenvector v, can be found by solving the equation 3.8.

$$(\lambda I - A)\, v = 0 \tag{3.8}$$

Choosing components and forming a feature vector

The eigenvalues are ordered from largest to smallest so that it gives us the components in order of significance. The eigenvector corresponding to the highest eigenvalue is the first principal component of the dataset. The second highest eigenvalue is the second principal component, and so forth. Once the eigenvalues are sorted, the number of eigenvalues to proceed the analysis is determined.

A graphical representation known as a scree plot, Kaiser Rule and proportion of variance explained are the three most common methods for selecting the number of components.

A scree plot is a plot of the number of principal components versus the eigenvalues. The value at the point where the elbow shape starts in the scree plot shows the optimal number of components (Cattell, 1966). In Figure 2.4, the elbow shape starts at the third component number, so; three principal components are enough to describe the data.

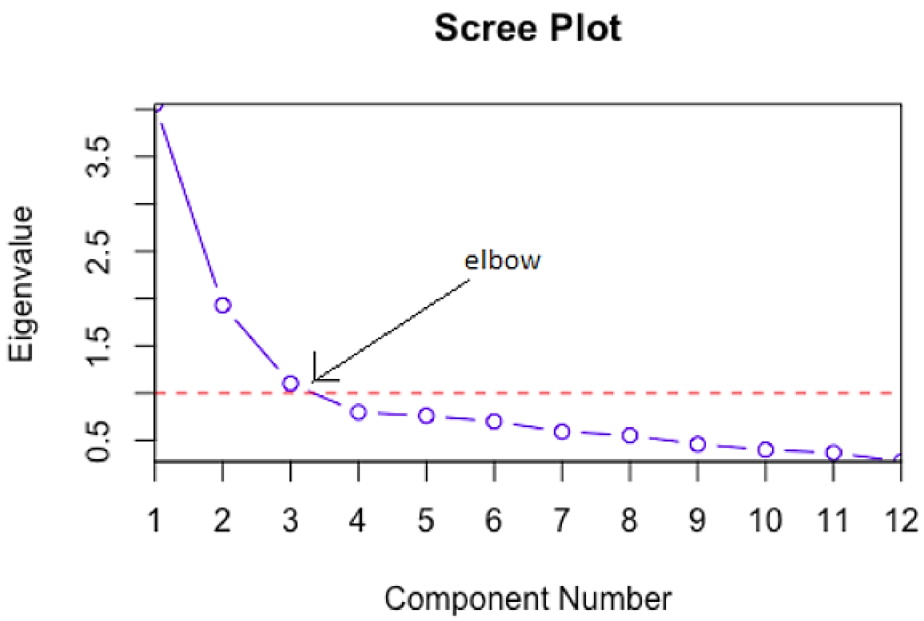


Figure 2.4 Sample Scree Plot

The Kaiser rule is the second option for selecting the number of components. According to this rule, the principal components whose eigenvalues are above 1.0 describe the data. (Kaiser, 1960).

The percentage of variance attributable to each of the specified components is the explained variance ratio (Lindgren, 2020). The proportion of variance explained is based on the rule of holding enough factors to take into account 90% of the variation (sometimes 80%) (Alpaydın, 2010). On the other hand, the number of components can be chosen by adding the explained variance ratio of each component until reaching a total of around 0.8 or 80% to avoid overfitting. (Lindgren, 2020). There is no rule of thumb for this option.

After selecting the number of components, a feature vector, which is a matrix of eigenvectors, is formed.

Forming Principal Components

The final step in PCA is forming principal components. The aim is to reorient the data from the original axis to the ones represented by the principal components using the feature vector created by the eigenvectors of the covariance matrix. To form the principal components, the transpose of the feature vector is taken and left-multiplied with the transpose of the scaled version of the original dataset. In equation 3.10, New Data is the matrix of the principal components, Feature Vector is the matrix of the eigenvectors, and Scaled Data is the scaled version of the original dataset. The superscript 'T' represents a transpose of a matrix, which is formed by changing rows for columns. In particular, a 2x3 matrix has a transpose of size 3x2.

$$\text{New Data} = \text{Feature Vektor}^{\mathrm{T}} \text{ x Scaled Data}^{\mathrm{T}} \qquad (3.9)$$

2.2.2 Decision Tree

Decision trees are a type of predictive learning algorithm that is simple and effective. Decision trees can be used for classification and predictive purposes.

Decision tree classification is a classification method that creates a model in the form of a tree structure, consisting of decision nodes and leaf nodes by property and goal (Russell & Norvig, 2003).

The decision tree algorithm is developed by dividing the dataset into smaller and even smaller parts. A decision node can contain one or more branches. The leaf node represents a classification or decision. The first node is called the root node. This top decision node in a tree corresponds to the best determinant. It follows the decisions in the tree from the root node down to a leaf node to predict a response. A decision tree can consist of both categorical and numerical data. Classification trees give nominal responses, although regression trees give numeric responses.

The training data is used to construct the tree. In Figure 2.5 the top node is the root node. Yes and No are the branches that are connecting nodes, showing the flow from decisions to outcomes. Each observation is classified by means of nodes. As the number of nodes increases, the complexity of the model also increases. The bottom nodes are the leaf nodes and the possible answers.

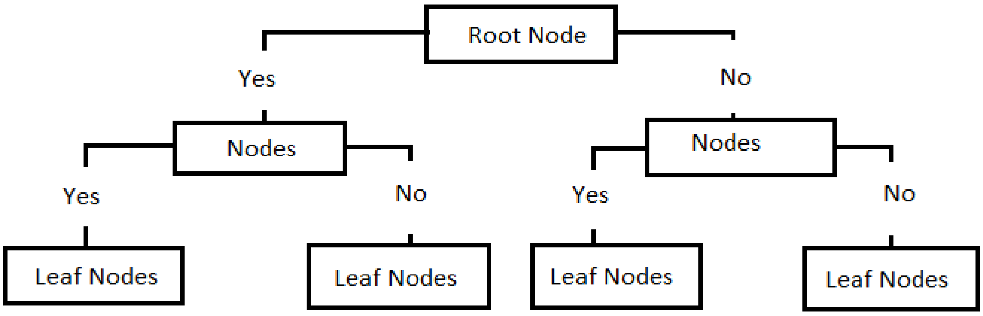


Figure 2.5 Decision Tree

Nodes are the building blocks of a tree. The root node to be selected should describe the dataset as much as possible and nodes are chosen in order to obtain the best possible feature split. A decision tree splits nodes into sub-nodes to make decisions. For that purpose, the splitting criteria are used to measure the quality of a split. Entropy (Quinlan, 1986) and gini-index (Breiman et al., 1998) are used for an optimum split of the features.

The basic idea of entropy is to measure the disorder of the features according to the target variable. The feature with less entropy chose the optimum split.

The entropy is calculated using the 3.10 Equation, where p_j is the probability of class j in a node, and $log_2(p_j)$ is the logarithm to the base 2 of the p_j.

$$Entropy = -\sum_j p_j log_2(p_j) \tag{3.10}$$

If the probability of the two classes is the same, entropy gets its maximum value as 1. When the entropy is equal to 0, a node is pure.

Entropy typically changes when we use a node in a decision tree to divide training samples into smaller subgroups. Information gain is a measure of this change in entropy. The decrease in entropy after a dataset is split on an attribute is the information gain.

Another criterion for an optimum split of the features, the Gini index (index) or Gini coefficient, is a statistical criterion developed by Italian statistician Corrado Gini in 1912 (Ceriani and Verme, 2012). When a dataset is randomly labeled, the Gini impurity estimates the frequency that any element will be mislabeled.

The Gini impurity is calculated using Equation 3.11, where p_j is the probability of class j in a node.

$$Gini\ Index = 1 - \sum_j p_j^2 \tag{3.11}$$

When all elements in the node have a single unique class, Gini Index gets its minimum value of 0. This means that there will be no further splitting of this node. Thus, the features chose a lower Gini Index for the optimum split.

There is no big difference between Gini and entropy. While entropy tends to build a more balanced tree, the Gini is prone to splitting the nodes whose frequency is high.

Overfitting Problems in Decision Tree

Overfitting is an important issue for decision tree models and many other predictive models. Errors and noise in training examples and coincidental regularities can cause overfitting.

Pruning is the approach to avoiding overfitting in building decision trees. Pruning is to remove predictive variables in branches that do not contribute well to the correct classification rate of the decision tree. Pre-pruning and post-pruning are the type of pruning.

Pre-pruning is to stop the tree from growing early. In this process, the tree is pruned before perfectly classifying the training set. It is a step-by-step process of branching by taking predictive variables one at a time without any classification.

Post-pruning is a process to remove the branches, which do not contribute to the model, from a completed decision tree.

If the model is overfit, decreasing the max_depth hyperparameter can prevent overfitting.

2.2.3 Support Vector Machine

Support vector machine is a supervised machine learning algorithm that can be used for linear and nonlinear classification and regression problems.

The basic idea behind the support vector machine is to divide and conquer. Firstly, the problem is transferred into a set of binary classification tasks. The first class is called “yes” and the second class is called “no”. If the decision-making variable is “yes” in the problem, then that decision is made. If the decision-making variable is "no" in the problem, then it is decided which two classes to be and again that class is divided into two classes, which are called "yes" and "no". If the problem is divided into two classes again, it will solve the problem, which is at which point it will be divided.

The support vector machine is a boundary that best separates the classes. The important terms to define this boundary are the support vectors, hyperplane and margin. Support vectors are the data points that are closest to the decision plane and the most difficult to classify (Berwick, 2003). Support vectors are coordinates of observation only and have an impact on the final decision boundary. The hyperplane is the decision plane that divides the data having different classes. In one dimensional space, the hyperplane is a point. In two-dimensional space, the hyperplane is a line and the hyperplane is a surface in three-dimensional space. The last important term in support vector machine, the margin (Figure 2.6), is the distance between the data points of both classes.

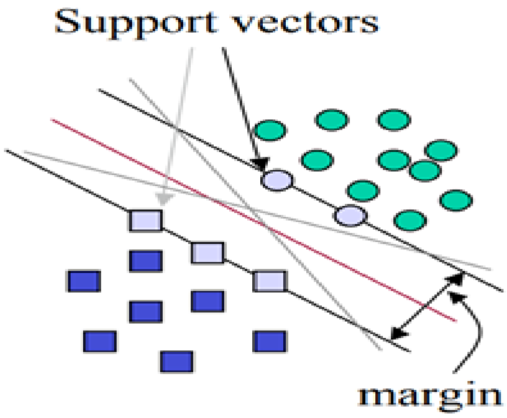


Figure 2.6 Support vectors and margin (Berwick, 2003)

Moreover, in an N (the number of characteristics) dimensional space, the support vector machine algorithm finds a hyperplane that clearly classifies the data points. These data can be linearly or non-linearly separable. Hence, SVM method performs the classification using a linear or nonlinear function.

Figure 2.7 shows that the data are separated by a line. In this case, the data are linearly separable. On the other hand, Figure 2.8 is an example of non-linearly separable data, which means that we cannot find a line to separate the data.

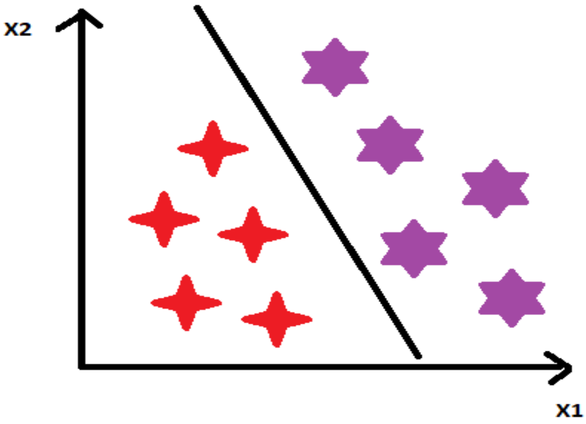


Figure 2.7 Linearly separable data

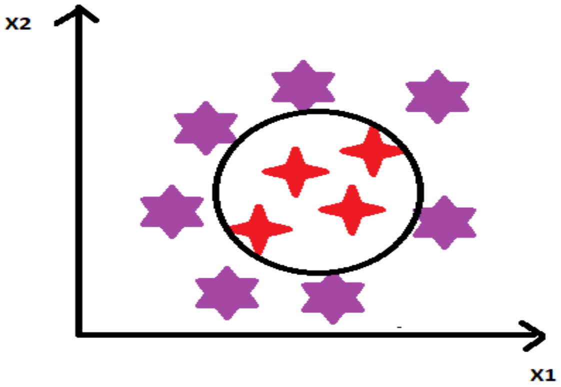


Figure 2.8 Non-linearly separable data

There are many possible hyperplanes that could be chosen to divide the linearly separable data points into two classes. The objective is to find a hyperplane with the largest margin (Figure 2.9). In that, future data points can be classified with more certainty as the margin increases.

Suppose there are only 2 classes and it is desired to learn where the arbitrary selected x data will remain in the plane. The straight line drawn from the origin to the data x is called the x vector, and $\bar{x}$ is the length of the x vector to the hyperplane. A vector (w) which is perpendicular to the hyperplane (H) is shown in Figure 2.9, and $\bar{w}$ represents the perpendicular distance from origin to the hyperplane.

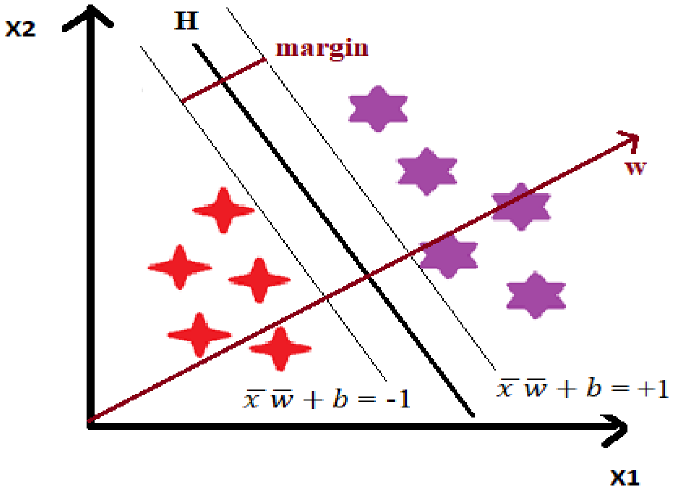


Figure 2.9 Vector w, margin and hyperplane

The points above and below the hyperplane correspond to the following inequalities 3.12 and 3.13, respectively.

$$\bar{x_\imath}\bar{w} + b > 0,\ \ y_i \geq\ +1 \tag{3.12}$$

$$\bar{x_\imath}\bar{w} + b < 0,\ \ y_i \leq +1 \tag{3.13}$$

Above inequalities are combined in equation 3.14;

$$y_i(\bar{x_\imath}\bar{w} + b) - 1 \geq 0 \tag{3.14}$$

The model learns by determining w and b values. New samples are classified by determining the value of y satisfying the inequality after computing w and b using the training set.

Margin is calculated by using Equation 3.15. The margin gets a higher value for the smaller value of ||w||.

$$m = \frac{2}{||\bar{w}||} \tag{3.15}$$

If there are some noise and outliers in the dataset, the given equation, which is called hard margin SVM, cannot tolerate them and fail to find the optimization. By adding a slack variable zeta to the constraints of the optimization problem, it is possible to satisfy the constraint even if some outliers do not meet the original constraint.

When data is characteristically non-linearly separable, the method, which is called kernel trick or method, is used to deal with this kind of problem. If the data is transformed from one space to another, a hyperplane can be found to separate the data.

The kernel trick works by adding nonlinear functions of the original variables until there are enough dimensions to separate the classes. The linear, radial (gaussian) and polynomial kernel are the most popular kernels.

2.2.4 Neural Network

In the discipline of machine learning, neural networks simulate the function of the human brain, allowing computer systems to identify patterns and solve common problems (“Artificial Intelligence”, 2020). Neural networks were inspired by human understanding of the biology of our brains and all those interconnections between neurons.

There is a very complex neural network in the body. The neuron is an electrically stimulating cell that transmits and processes information in the nervous system. These neurons transmit the electric signal from dendrites to the ends through the axons. Signals from neurons are transmitted to the brain along the nervous system (Von Bartheld et al., 2016).

An artificial neural network works using this process. A copy of the biological neural network is made into an artificial model, and each neuron layer is linked to the neurons on the next layer (Thakur et al., 2021).

Artificial neural networks consist of node layers that contain an input layer, one or more hidden layers, and an output layer. A typical example of a neural network with an input layer, 2 hidden layers, and an output layer is presented in Figure 2.10. Each circle is called as a “node” corresponding to a neuron.

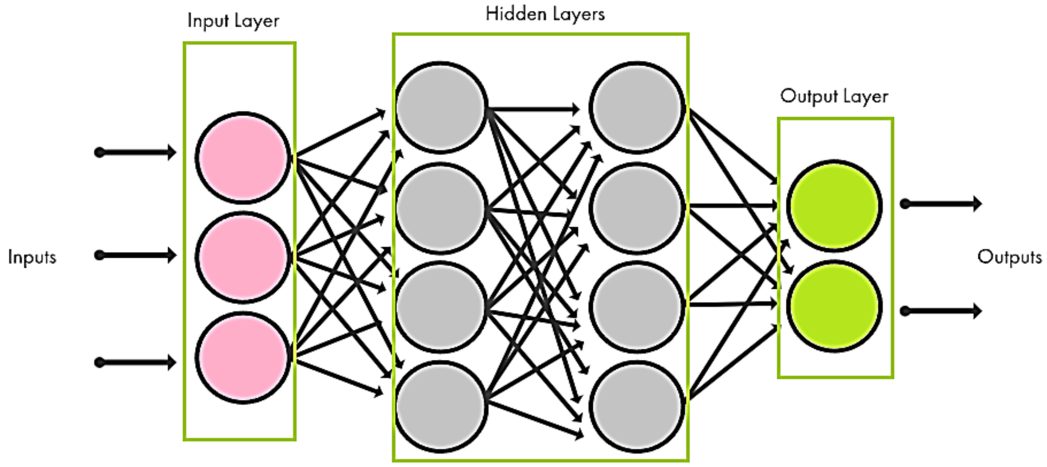


Figure 2.10 Neural Network Architecture

Information is transmitted to the network from the input layer.

If the network consists of a single layer, it is called a single layer artificial neural network. The perceptron, which has a single neuron, is the oldest and simplest form of a neural network. If the network consists of many neurons and hidden layers, it is called a multilayer artificial neural network. The non-linearity of the output can be increased by adding layers.

The layer between the input layer and the output layer is called the hidden layer. Information from the input layer is processed in hidden layers and sent to the output layer. They do not interact directly with input or output data. It is the layer where all the computation is done. Figure 2.11 shows the weights, net input function and activation function. The grey circle is the activation node. Each node connects to each node from the next layer, and each connection line has a specific weight (w). Weights are assigned after an input layer is specified. These weights play an important role in determining the importance of any given variable. All inputs (x_1, x_2…) are multiplied by their respective weights (w_1, w_2…). Then, all multiplication results for a neuron and a bias (threshold) are summed for that neuron. The activation function is shifted to the left or right using bias. This calculation, which is in Equation 3.16, is the net input function.

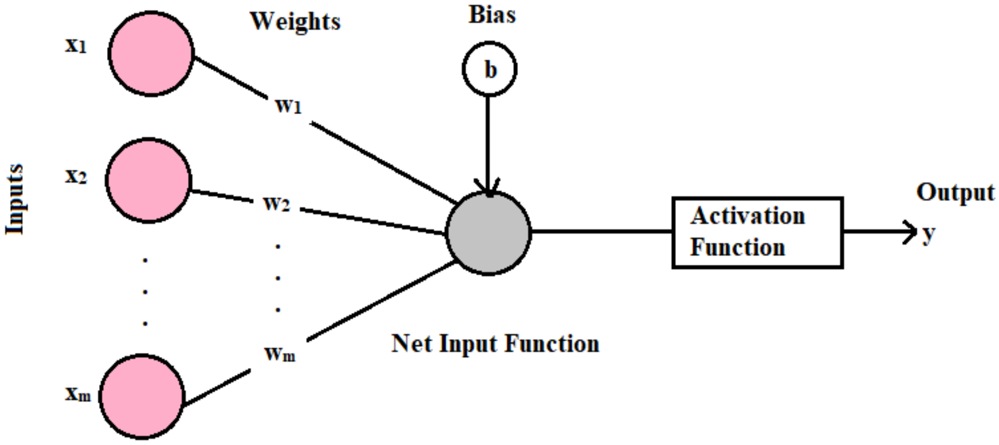


Figure 2.11 Weights, net input function and activation function

$$Net\ Input\ Function = z_i = \sum_{i=1}^{m} w_i x_i + bias \qquad (3.16)$$

If the total value for a neuron exceeds a given threshold value, the neuron fires and data pass to the next neuron in the network.

Activation function is used to introduce non-linearity in the model. There are many activation functions such as sigmoid, tanh, ReLU, leaky Relu, etc. (Figure 2.12). The multilayer artificial neural network uses sigmoid function as an activation function that often makes the error minimum (Oztemel, 2003). The Sigmoid transfer function receives values from its net input function and generates outputs value between 0 and 1, while tanh function generates outputs value between -1 and 1 (Akbari et al., 2014).

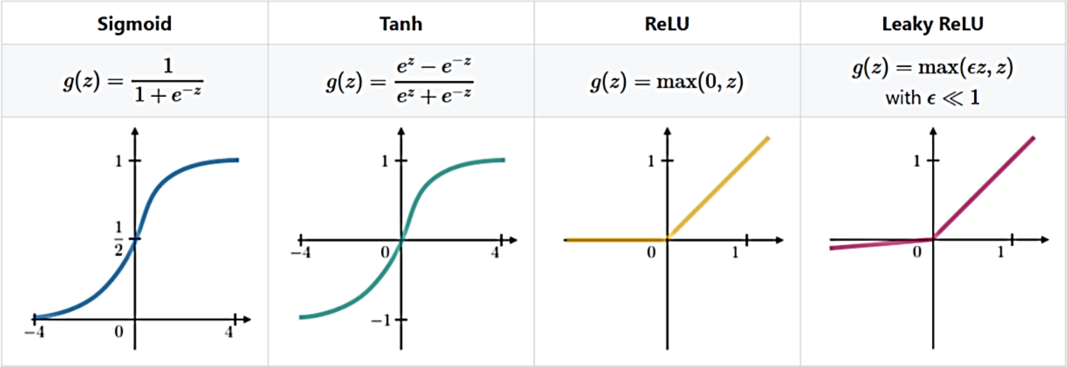


Figure 2.12 Most common used activation function graphs (Nalborczyk, 2021)

The output value obtained by applying the activation function and each neural net has a single output. The output produced may be the input of another neuron.

In particular, the transformation task between input and output is important to adapt a system. Because input and output layers can only transmit data, the number of hidden layers and the number of neurons in each hidden layer determine the calculation capability of an artificial neural network. An error method can be used for determining these numbers (Gomes et al., 2004).

The objective function is the mean square error function in an artificial neural network optimization. Thus, the value of the parameters (weights) that minimize the error when mapping inputs to outputs is found using an optimization algorithm.

Optimization algorithms can be divided into two categories as one-dimensional optimization and multi-dimensional optimization algorithms.

Some of the functions that are used for one-dimensional optimization are Convex Unimodal Functions, Non-Convex Unimodal Functions, Multimodal Functions, Discontinuous Functions (Non-Smooth), and Noisy Functions. One-dimensional functions that receive a single input value give a single evaluation of the input. Inputs to the function on the x-axis and outputs of the function on the y-axis are given, and so both inputs and outputs are visualized on a single plot.

Gradient descent, backpropagation algorithm, Newton method, conjugate gradient, quasi-Newton method, and Levenberg-Marquardt algorithm are the most commonly used functions for multi-dimensional optimization algorithms.

The gradient descent algorithm is one of the most popular and simplest optimization algorithms. Gradient descent aims to reach the global minimum value, starting with randomly imported variables.

The backpropagation algorithm is an extension of the gradient-based delta learning rule, and a local optimization technique based on the steepest gradient method. The error is calculated and checked whether the error is minimized. If the error is large, then the parameters (weights) are updated. The error is then checked again. The process is repeated until the error reaches the minimum. When the error reaches a minimum, it can feed some inputs to the model and get the output. The optimization algorithm stops, as seen in Figure 2.13, when the output is true. This process is called Backpropagation.

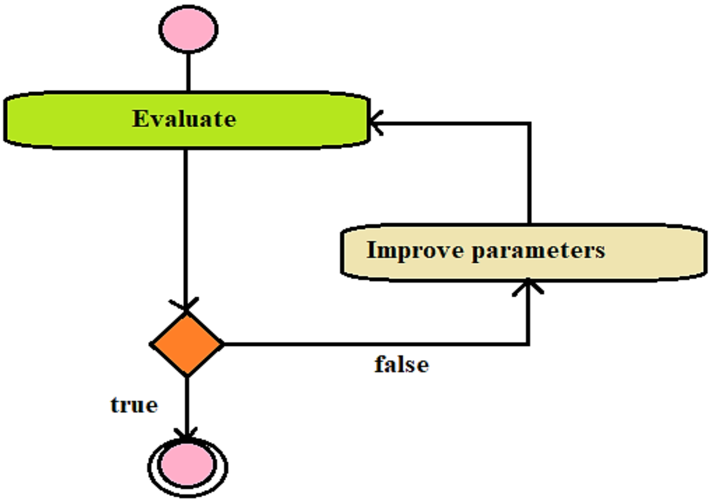


Figure 2.13 Training process of backpropagation

In the backpropagation algorithm, each iteration consists of three phases, feed-forward, backpropagation and the adjustment of the weights (update).

Another multi-dimensional optimization function, Newton's method, uses the Hessian matrix. Thus, it is a second-order algorithm. The objective of this algorithm is to use the second derivatives of the loss function to find better training routes.

The quasi-Newton method is a type of Newton's method, but this algorithm does not use the second derivatives of the loss function. The quasi-Newton method, on the other hand, uses just gradient information to approximate the inverse Hessian at each iteration of the algorithm.

The conjugate gradient algorithm is a line search method that is performed with conjugate directions. This is why it usually converges faster than the steepest descent method. This method can be considered an intermediate method between gradient descent and Newton method. The difference between this and Newton method is that this algorithm does not require the Hessian matrix.

The Levenberg-Marquardt (LM) algorithm is the most frequently used backpropagation algorithm, as it combines the speed of the Gauss-Newton optimization method and the stability of the steepest descent method in minimizing the sum-squared errors of the output results. (Suratgar et al., 2005). To perform this method, the loss index must be in the form of a sum of squares. It requires both the gradient and the Jacobian matrix of the loss index. For small data sets, the Levenberg-Marquardt algorithm can be used due to its high speed and precision.

2.3 Accident Forecasting Studies in Underground Coal Mines

A literature survey was conducted for the studies, which were conducted by using principal component analysis, decision trees, support vector machines, and neural network algorithms to prevent work-related accidents in underground coal mines. During the literature survey it is seen that the principal component analysis is commonly used by researchers for dimension reduction. The decision tree, support vector machine, and neural network are mainly used for classification purposes to determine the causes of accidents. However, there weren't many comparative studies on which type of analysis would be better to analyze coal mine accident data and predict the severity of the accident.

The work face gas emission prediction model was developed by Ning et al. (2009) using a support vector machine (SVM) model based on data statistics of a mine work face gas emission to prevent work-related accidents. The outcomes were accurate, demonstrating that the model's face gas prediction is viable and useful.

Ruilin and Lowndes (2010) used Chinese coal mines’ statistics for prediction of coal and gas outbursts. According to this study, the combined fault tree analysis and artificial neural network model may offer a credible alternative way to predict the possible risk of coal and gas outbursts. The model was used by Hong et al. (2010) for a gas warning system. They concluded that it offers very good features in gas extraction, analysis, and judgement. However, they state that it is needed to do extensive research because early warning systems are a deep and major problem.

Carnero and Pedregal (2010) have developed an accident prediction model by analyzing data of light injury, serious injury, and deadly work accidents in Spain and identifying work accidents with these severity levels. Multivariate Unobserved Components models were used to deal with the irregular sampling interval of the data and forecast occupational accidents for different levels of severity.

Sanchez et al. (2011) used the support vector machine method to predict work-related accidents. Before applying the method, semi-parametric principal component analysis was used for dimensional reduction. Because of unsatisfactory results, another dimensional reduction method, which was multivariate adaptive regression splines, was applied and obtained good results. The results of this methods were selected as input for support vector machine model. This SVM technique made classification according to worker's working conditions. As a result, they observed that a support vector machine model does not overfit the experimental data and performs better than back-propagation neural network models.

Nenonen (2013) analyzed the statistical database for slipping, stumbling, and fall work-related accidents in Finland between 2006-2007 using data mining methods, and it has been concluded on the consequences of accidents, whether accidents are actually caused by these factors. In the study, the data was analyzed using decision tree and association rules. As a result of this study, data mining methods were shown to be effective.

Alaeddinoğlu et al. (2015) trained the artificial neural network model with the results of the risk assessment in the past and offered this method to help the expert person decide the consequences that may arise from the potential risks.

Sanmiquel et al. (2015) applied a clustering algorithm to the dataset containing the description of the mining accident reasons. The Quinlan algorithm, which is used in data mining as a decision tree, was applied to the dataset and the causes of accidents were classified based on the feature value. Results were obtained that could help develop appropriate prevention policies to reduce injuries and deaths.

In the study by Chen et al. (2015), the stability of mine tailings dam was analyzed based on principal component analysis and neural network. They agreed that, before the neural network analysis, preprocessing the original sample with the principal component analysis can significantly improve training speed and precision, and the model is feasible in the analysis of the stability of mine tailings dam.

Xiangzhong Meng, Peng Lu and Baolei Wang developed Coal Mine Safety Warning System Based on Principal Component Method and Neural Network to prevent the accidents (2017). They stated that using PCA to extract data can effectively reduce data, eliminate interference and improve the efficiency and accuracy of neural network recognition.

Ye Zhang et al. (2022) propesed a back propagation neural network prediction method based on primary component analysis and deep confidence network optimization for water inrush in order to provide an effective risk assessment of water inrush for coal mine safety production. As a result of this study, the principal component analysis-deep belief network model is able to eliminate the defects in standard feature selection algorithms and successfully filter out missing and noisy data to provide a more trustworthy water inrush accident evaluation model.

Wu et al. (2022) stated that principal component analysis is a dimension reduction methodology that can be useful for identifying significant variables or components and can be utilized effectively in hazard, risk, and emergency assessment. Moreover, because it was sensitive to outliers, the PCA approach could be used to construct prediction and forecasting systems.

CHAPTER 3

STUDY AREA AND DATA

3.1 Study Area

In the late 18th century, the imports of hard coal that had not been in the country to meet the needs of industrial branches began. On November 8, 1829, the discovery of hard coal outcrop along Viran Creek in Eregli by Uzun Mehmet, who was a bluejacket, constitutes the basis for today's coal business. The production of coal in Zonguldak coal basin began in 1848 (Turkish Hard Coal Enterprise, 2022).

In accordance with the general industry and energy policy of the state, Turkish Hard Coal Enterprise (THCE) was established in 1983 to contribute to the domestic economy by optimizing the reserves of hard coal and meeting the country's requirements for hard coal. However, the year of the establishment is considered 1848 because it inherits the coal mining process, which was considered to have started in 1848 at the Zonguldak basin. The production of hard coal, which was about 2 to 2.5 million tonnes/year in recent years, has been in the five establishments. Four of the establishments (Armutcuk, Kozlu, Karadon, and Uzulmez) are located within the Zonguldak province, and one (Amasra) is within the province of Bartın. Moreover, the concession area (Figure 3.1), including these establishments, is 6,885 km2 by the Council of Ministers’ decision dated 14/04/2000 and numbered 2000/525 (THCE, 2022).

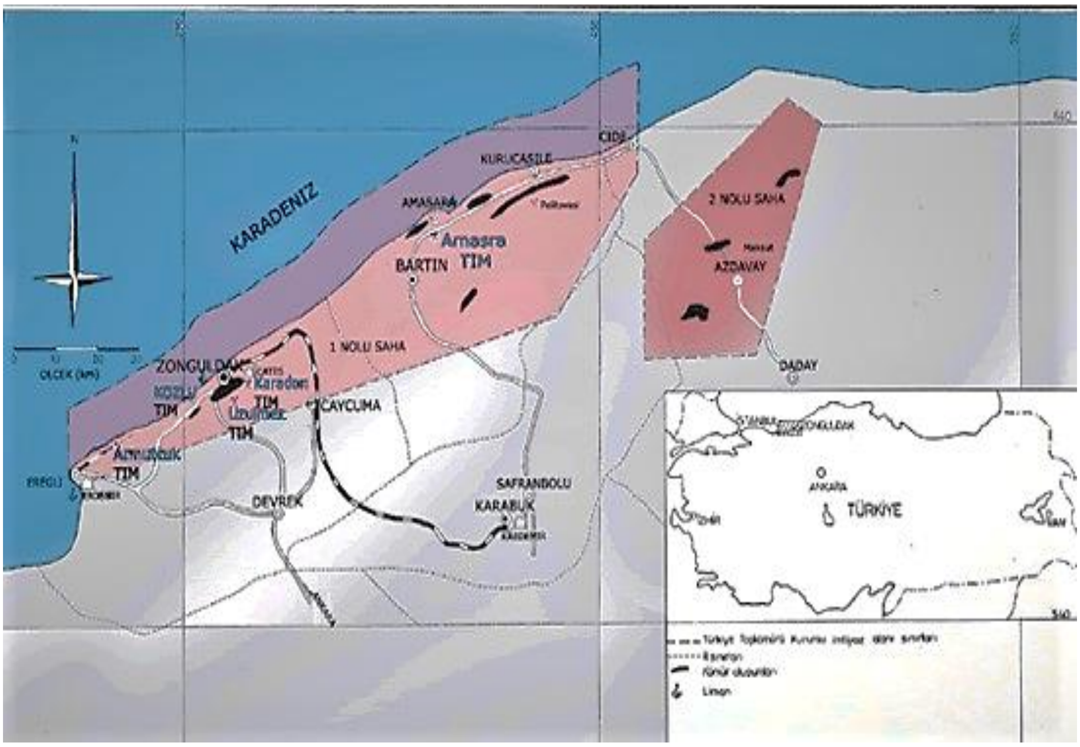


Figure 3.1 The license area of Turkish Hard Coal Enterprise (THCE, 2022)

The total geological reserve, determined at a depth of 1200 m in the reserve search conducted in the basin so far, is 1.511 billion tonnes and approximately 48% of this is considered proven reserve (2021 Hard Coal Sector Report, 2022). Table 3.1 shows the total hard coal reserve amounts of TTK and the hard coal reserve amounts of Armutçuk, Kozlu, Üzülmez, Karadon, and Amasra in 2022.

Table 3.1 Turkiye's hard coal reserves in 2022 (tonnes) (“2021 Hard Coal Sector Report”, 2022)

RESERVE	**Armutçuk**	**Kozlu**	**Üzülmez**	**Karadon**	**Amasra**		**TTK**
					A	**B**	
Ready	1.907.524	2.421.222	328.414	3.154.507	330.000		8.141.667
Proven	6.719.800	62.715.504	132.559.492	127.643.082	4.897.000	395.954.757	730.489.635
Possible	14.407.491	40.539.000	94.342,00	159.162,00	7.690,00	151.161.950	467.302.441
Probable	7.883.164	47.975.000	74.020.000	117.034.000	56.619.859	2.192.919	305.724.942
TOTAL	**30.917.979**	**153.650.726**	**301.249.906**	**406.993.589**	**69.536.859**	**549.309.626**	**1.511.658.685**

The maximum run of mine coal production in the history of the basin was 8.5 million tonnes in 1974, and the saleable production was 5 million tonnes in 1967 and 1974. Today in Turkiye, the production of hard coal is carried out in Zonguldak hard coal basin by Turkish Hard Coal Enterprise, by private sector companies that work with a royalty method at Turkish Hard Coal Enterprise 's concession site, and also by the companies that Turkish Hard Coal Enterprise Institution transfers licenses to. Table 3.2 illustrates the amount of hard coal produced by Turkish Hard Coal Enterprise Institution and the private sector for years (“2021 Hard Coal Sector Report”, 2022). According to the table, the minimum production was in 2020. The lowest production is expected in 2020 due to the COVID-19 pandemic.

Table 3.2 The amount of hard coal production (tonnes) (“2021 Hard Coal Sector Report”, 2022)

Years	**Turkish Hard Coal Enterprise**	**Private Sector**	**Total**
2010	1.708.844	883.074	2.591.918
2011	1.592.515	1.026.732	2.619.247
2012	1.457.098	835.157	2.292.255
2013	1.366.509	549.332	1.915.841
2014	1.300.154	488.187	1.788.341
2015	948.573	486.309	1.434.882
2016	911.002	404.968	1.315.970
2017	823.042	411.212	1.234.254
2018	686.142	415.442	1.101.584
2019	734.316	472.432	1.206.748
2022	712.689	352.862	1.065.551
2021	870.018	365.043	1.235.061

Moreover, the complex geological structure of the Zonguldak coal mining basin makes production difficult with fully mechanized systems, and the production of hard coal is mainly carried out in a labor-intensive way that depends on human power. However, in recent years, production with mechanized and semi-mechanized systems that meets the requirements of the basin has been successful (“2021 Hard Coal Sector Report”, 2022).

3.2 Data

For this study, 8406 underground accident data belonging to Turkish Hard Coal Enterprise were analyzed. The obtained data covers the districts of Amasra, Armutçuk, Karadon, Kozlu, Üzülmez, and the dates between March 2008 and December 2010.

As a first step, variables are determined for accident data set. The data set had eleven variables (dimensions) that are shift, day, job, education, type of accident, reason of the accident, location of the accident, severity of the accident, age, seniority, affected body part. Each dimension had several categories that are numeric and categorical. Variables and their categories are shown in table 3.3, and the variables specified for the creation of the accident data table are described.

Table 3.3 Variables of the data set

Code	**Shift**	**Code**	**Day**	**Code**	**Job**	**Code**	**Education**	**Code**	**Type of Accident**
1	First	1	Monday	1	Blaster	1	Primary School	1	Bump, Break
2	Second	2	Tuesday	2	Chainman	2	Secondary School	2	Electrical
3	Third	3	Wednesday	3	Driller	3	High School	3	Falling rocks
		4	Thursday	4	Duties Man	4	University	4	Gas poisoning or suffocating
		5	Friday	5	Electrical Electronics Worker	5	Unknown	5	Gas/dust explosion
		6	Saturday	6	Environmental Worker			6	Ground support
		7	Sunday	7	Ground Support Worker			7	Hand tools
				8	Haulage Worker			8	Inrush
				9	Machinist			9	Manual Handling
				10	Maintenance and Repair Worker			10	Material Handling and Usage
				11	Mechanization and Press Worker			11	Mechanical
				12	Miner			12	Miscellaneous injury
				13	Mining Engineer			13	Slipping, Falling, Tripping, Ankle Sprain
				14	Mining Technician				
				15	Development Worker				
				16	Production Worker				
				17	Pump Worker				
				18	Service Man				
				19	Washery worker				
				20	Welding				

Table 3.3 (continued)

Code	Reason of the Accident	Code	Location of the Accident	Code	Accident Severity	Age	Seniority	Code	Affected Body Part
1	Coal transfer	1	Inclined Shaft	1	Death	21-55	0-31	1	Arm
2	Personal Mistake	2	Footwall	2	Seriously Wounded			2	Brest
3	Equipment error	3	Gallery	3	Injured			3	Calf
4	Geological conditions	4	Roadways (development)	4	Slightly injured			4	Dorsi
5	Locomotives	5	Miscellaneous					5	Face
6	Transportation vehicles	6	Ground support					6	Foot Finger
7	Other	7	Production areas					7	Hand Finger
		8	Transportation					8	Head
		9	Pump					9	Knee
		10	Shaft					10	Leg
		11	Tippling					11	Neck
								12	Shoulder
								13	Waist
								14	Arm
								15	Brest

- **Descriptive statistics of the dataset**

The variable “shift” was divided into subtitles as first, second, third. The first, second, and third shifts cover the working time from 08:00 to 16:00, from 16:00 to 24:00, and from 24:00 to 08:00, respectively. While there were 4523 accidents on the first shift, 2061 and 1822 accidents occurred on the second and the third shift, respectively.

Another variable is the days of the week. It was seen that there were almost the same number of accidents every day, except on Sunday. The total number of accidents on Sundays is 10. Monday is in first place with 1542 accidents (18.34%).

Moreover, in the data set, the variable which is called "job" shows the professions of those affected by the accidents. The variable job was divided into twenty subtitles. When the number of accidents according to jobs is analyzed, production workers have the highest share of the accidents. The production workers were affected from 80.24% of the total accidents. Figure 3.2 shows the distribution of the jobs with respect to the total accidents.

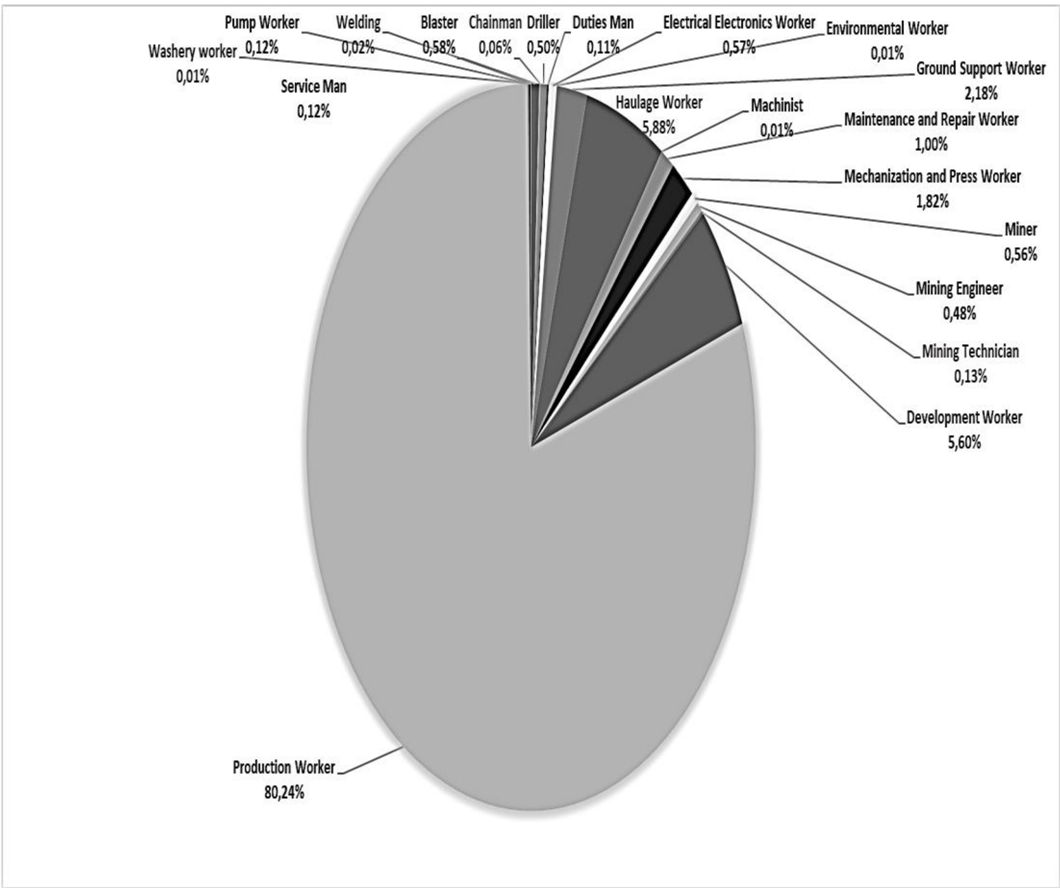


Figure 3.2 The distribution of the jobs with respect to the total accidents

The education level of those affected by the accidents is another variable. The education level was divided into five categories, which are primary, secondary, high school, university, and unknown. 4561 (54.26%) of the observations have primary school education, and this category has the highest percentage of accidents.

The type of accident consists of thirteen sub-categories as bump-break, electrical, falling rocks, gas poisoning or suffocating, gas/dust explosion, ground support, hand tools, inrush, manual handling, material handling and usage, mechanical, miscellaneous injury, slipping-falling-tripping-ankle sprain. The most common type of accident is falling rock, with 88.52%.

Accidents' reasons were geological conditions, personal mistake, equipment error, coal transfer, locomotives, vehicles and others. According to this variable, personal mistake has the highest rate. There were 3870 accidents (46.04% of the total) due to personal mistakes.

Inclined shaft, footwall, gallery, roadways (development), ground support, transportation, production areas, pump, shaft, tippling and miscellaneous areas are common accident sites. When considering the number of accidents, the production area appears to be the most dangerous area, with 4420 accident data (52.58%) out of 8406.

Another variable is the severity of each accident. The prediction models were built to predict this variable.

The age of the workers who had accidents is also another variable in the data set. The age range of workers who subjected to accidents is in between 21 and 55.

One further variable that needs to be evaluated in terms of work-related accidents is seniority. Seniority represents the employee's year of experience at the time of the accident. According to the data set, workers with 0–7 years of seniority are more likely to be in accidents; they account for 72.22% of accidents.

The last variable is the affected body part. This variable shows where the body was injured in the accident. The hand finger is the most injured body part (29.26%) as a result of accidents. Figure 3.3 shows the number of accidents with respect to affected body parts.

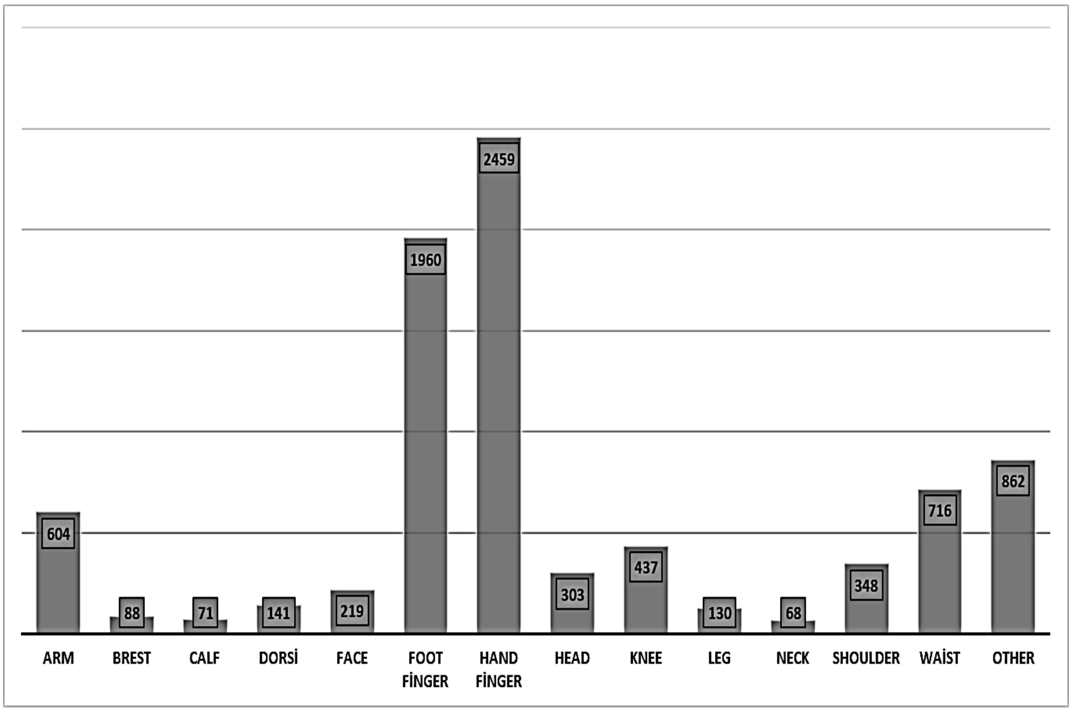


Figure 3.3 The number of accidents with respect to affected body parts

CHAPTER 4

ANALYSING DATA AND BUILDING THE MODELS

4.1 Applying Principal Component Analysis

This section describes the study carried out to omit meaningless variables from the raw data set of 8406 accidents and 11 variables. In other words, principal component analysis (PCA) was applied for dimension reduction of the data set.

Principal component analysis (PCA) was performed using R Studio, which is free and open-source software for data science. The data set had both numerical and categorical data. Thus, before the analysis, all categorical data was converted into numerical data. To import the data into program the spreadsheet file was used and the input data was a vector. To convert these data vectors into numeric values, the factor method was used to convert the input, which was a vector, into the factor, which is a data structure that is used to classify data. Then the numeric method was used to convert the factor into numeric. Moreover, the factor numbers such as age and seniority were converted first into a character vector and then into a numeric value.

The second step is to prepare the data set for analysis is standardization. As it was mentioned in the Section 2.2.1, normalization or standartization is a method of bringing all data into a comparable range to make comparisons more meaningful. After normalization, all data in the matrix are within the same range where the mean is zero and the variance is one. Thus, standardization was applied by using scale function, and after this process, the data was ready to be analyzed.

The output of the correlation matrix resulting from the PCA applied to accident data using the R software is given in Table 4.1. The correlation matrix given in Table 4.1 shows the eigenvalues, the variance percent and the cumulative variance percent. As it is mentioned in Chapter 2, eigenvalues are simply the coefficients attached to eigenvectors, which give the amount of variance carried in each principal component. Thus, in Table 4.1, eigenvalues show the variances. Because of having the data set with eleven variables, there are eleven eigenvalues and eigenvectors. The components, in other words, dimensions, are listed in order of importance by the eigenvalues, which are ordered from largest to smallest in Table 4.1. The first principal component (first dimension) explains 21.76% of the variation in the data, while the second principal component (second dimension) explains 14.59% of the variation. The summation of these two corresponds to 36.35% of the variation. The number of components can be chosen by adding the explained variance percent of each component until reaching a total of around 80% to avoid overfitting. (Lindgren, 2020). Therefore, 81.82% of the data can be interpreted with the seven components given in Table 4.1.

Table 4.1 The output of the correlation matrix

	Eigenvalue	**Explained Variance Percent**	**Cumulative Variance Percent**
1st Principal Component	2.3935705	21.759732	21.75973
2nd Principal Component	1.6053013	14.593648	36.35338
3rd Principal Component	1.0980707	9.982561	46.33584
4th Principal Component	1.0055105	9.141004	55.47685
5th Principal Component	0.9917678	9.016071	64.49292

Table 4.1 (continued)

	Eigenvalue	**Explained Variance Percent**	**Cumulative Variance Percent**
6th Principal Component	0.9723008	8.839099	73.33201
7th Principal Component	0.9338560	8.489600	81.82161
8th Principal Component	0.8608942	7.826311	89.64793
9th Principal Component	0.5377797	4.888906	94.53683
10th Principal Component	0.4606229	4.187481	98.72431
11th Principal Component	0.14032256	1.275687	100.00000

Looking at a Scree Plot, which is the plot of explained variance percent ordered from largest to smallest, is another method to determine the number of principal components. The optimal number of components is selected as the value at the point where the elbow form begins in the scree plot. Figure 4.1, scree plot, shows the percentage of explained variance by each principal component, and Figure 4.2 shows the cumulative sum. From the scree plot, it is seen that the elbow shape starts at the third component. However, the sum of the percentages of explained variance of the first three components explains 46.34% of the total data. This is not enough for reaching high prediction accuracy. Therefore, seven components identified by the previous method were chosen instead of three components.

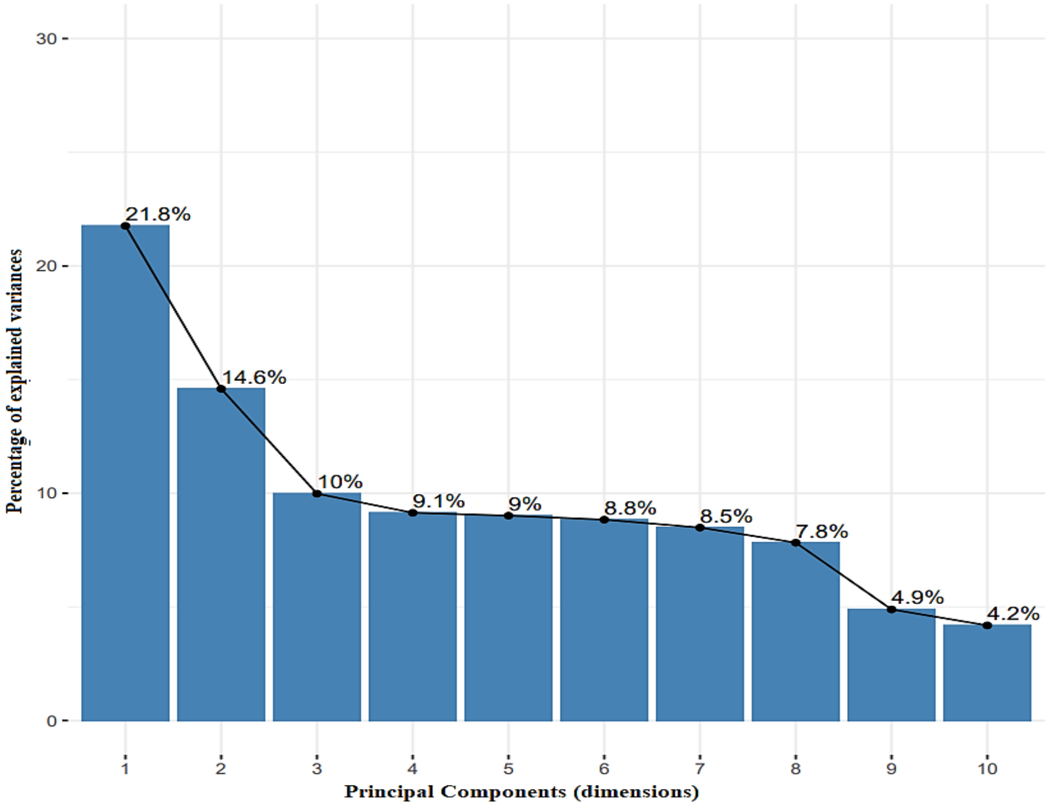


Figure 4.1 The scree plot of the percentage of explained variance with respect to principal components

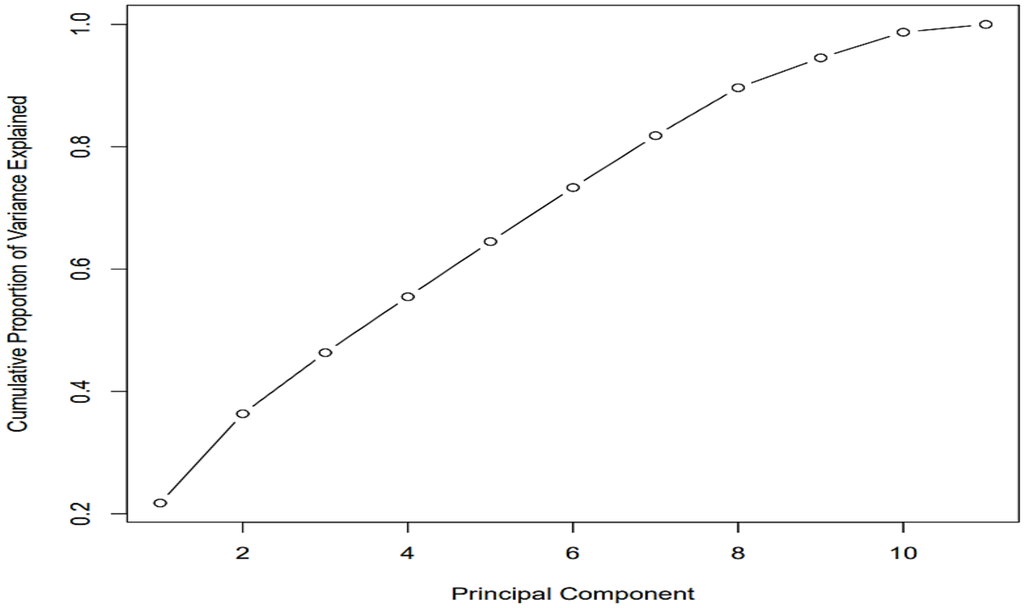


Figure 4.2 Cumulative proportion of explained variance with respect to principal components

After determining the number of components, the PCA results are visualized. All plots are shown on the plane of the first two components because the first principal component shows the most variation, and the second principal component shows the second most. Firstly, the score plots of the first two principal components are plotted. On the x and y axes, these scores are referred to as the first and second principal components, respectively. Figure 4.3 is the score plot that is individuals factor map of severity of accidents, and Figure 4.4 is the score plot that is individuals factor map of job. In the maps, the points are the projections of each data point along the directions with the largest variance, which are the first and second principal components. These PCA plots shows clusters of samples based on their similarity. It can be seen that PCA performed not too well in Figure 4.3. Because clusters are clearly not separate from each other. But it performed better in Figure 4.4.

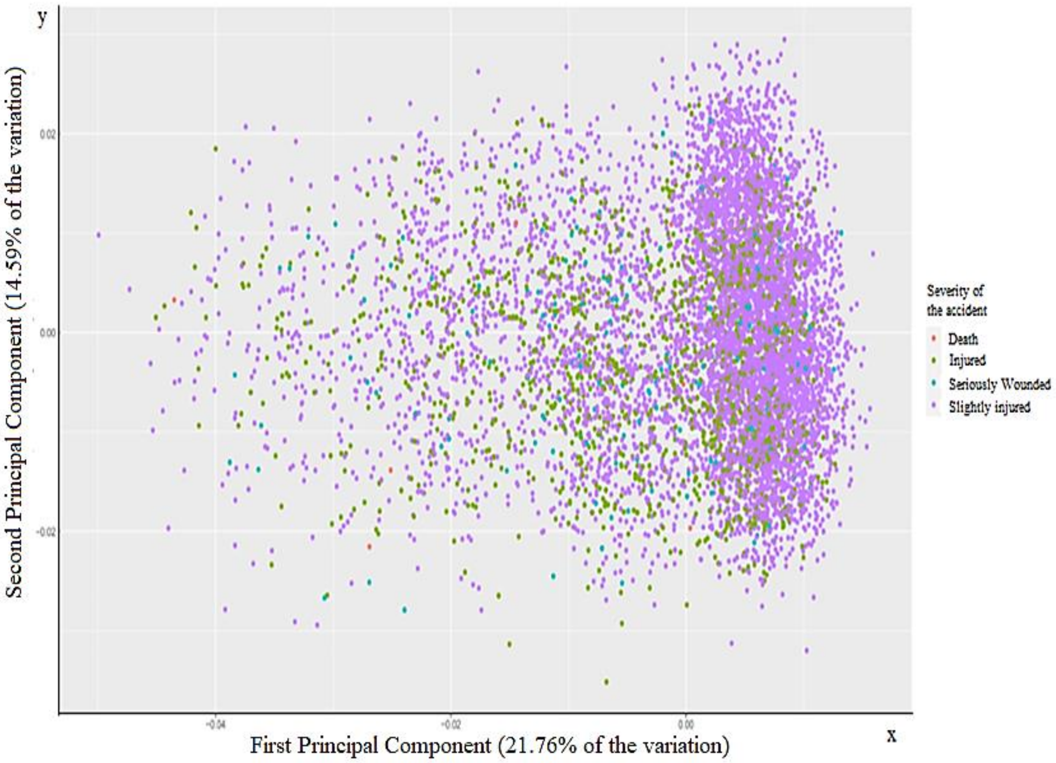


Figure 4.3 Individuals factor map of severity of the accident

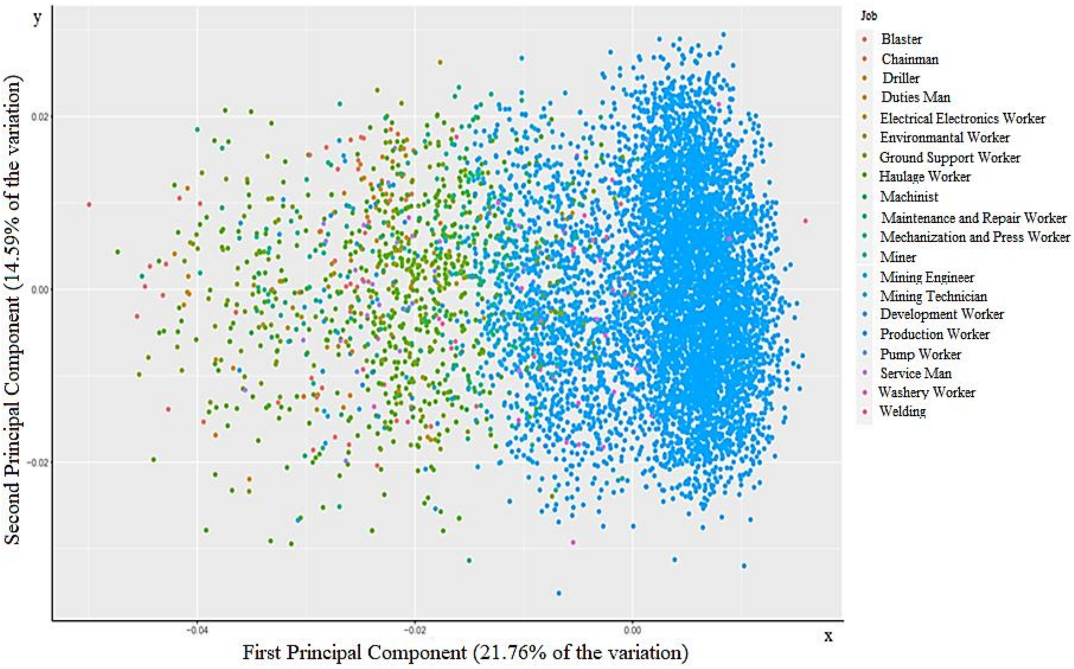


Figure 4.4 Individuals factor map of job

Another way of visualizing PCA results is the correlation circle. The correlation circle shows the relationships between all variables. The quality of representation of the variables on factor map is called cos^2. A high cos^2 indicates a good representation of the variable on the principal components. In this case the variable positioned close to the circumference of the correlation circle. A low cos^2 indicates that the variable is not perfectly represented by the principal components. In this case the variable is close to the center of the circle. Figure 4.5 shows the correlation circle of the PCA results. The radius of the circle is 1. First principal component is represented on the horizontal axis, and second principal component is represented on the vertical axis. Inside the circle, each arrow, which has different lengths, indicates a variable. The angle between arrows (variables) shows how well the variables are correlated on the factorial plane. The angle is small if the representation of the two variables on the factorial plane is positively correlated. For example, according to the circle age and seniority are positively correlated on this factorial plane, and their arrows are longer than others. This means that these variables well explain the variance of the data on the factorial plane. The job variable, according to the circle, is negatively correlated with age and seniority variables since they are positioned on opposite sides of the plot origin.

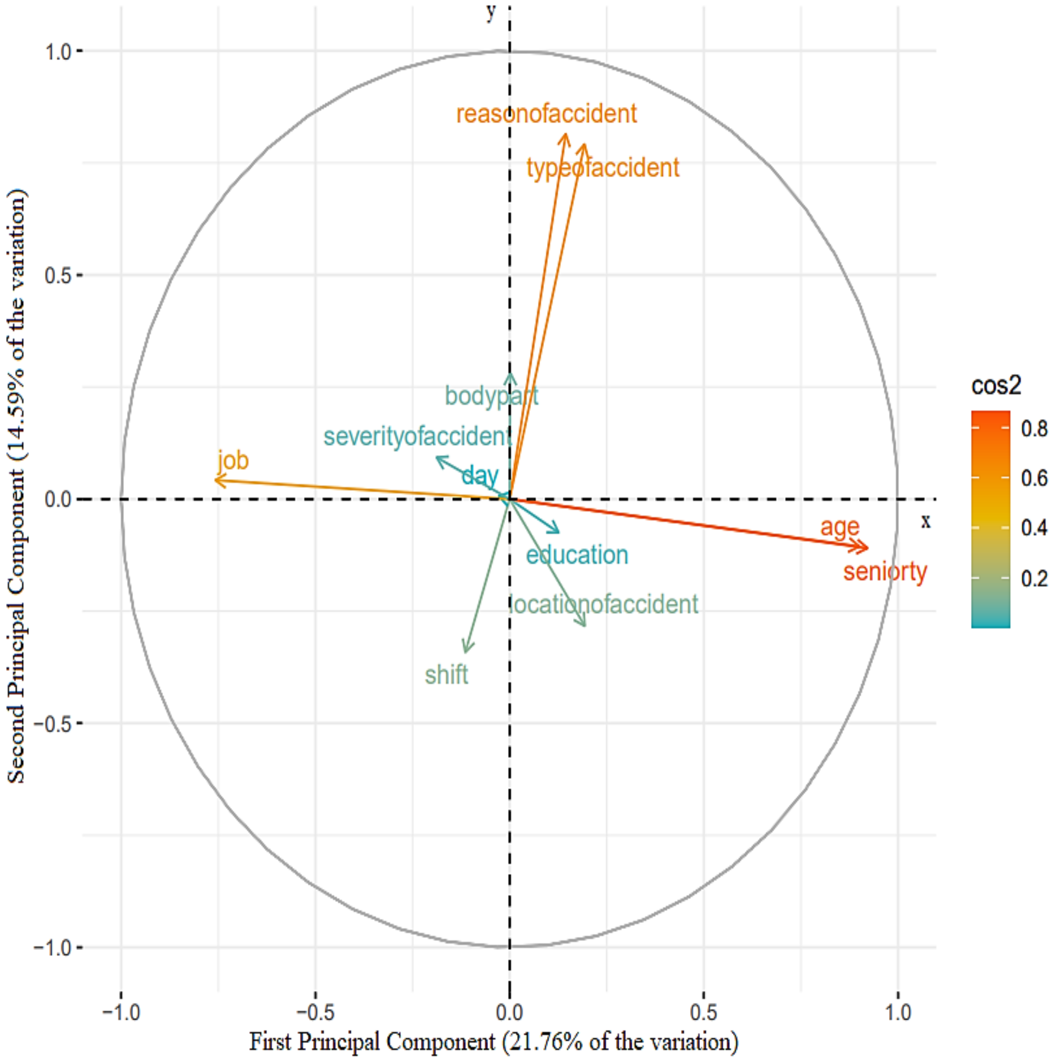


Figure 4.5 Correlation circle of PCA results

Moreover, $\cos^2$ bar plot is shown in Figure 4.6. From the bar plot, the variable day has the lowest $\cos^2$ value. This variable is not perfectly represented by the PCs. Seniority, age, reason of the accident, type of accident and job have high $\cos^2$ values.

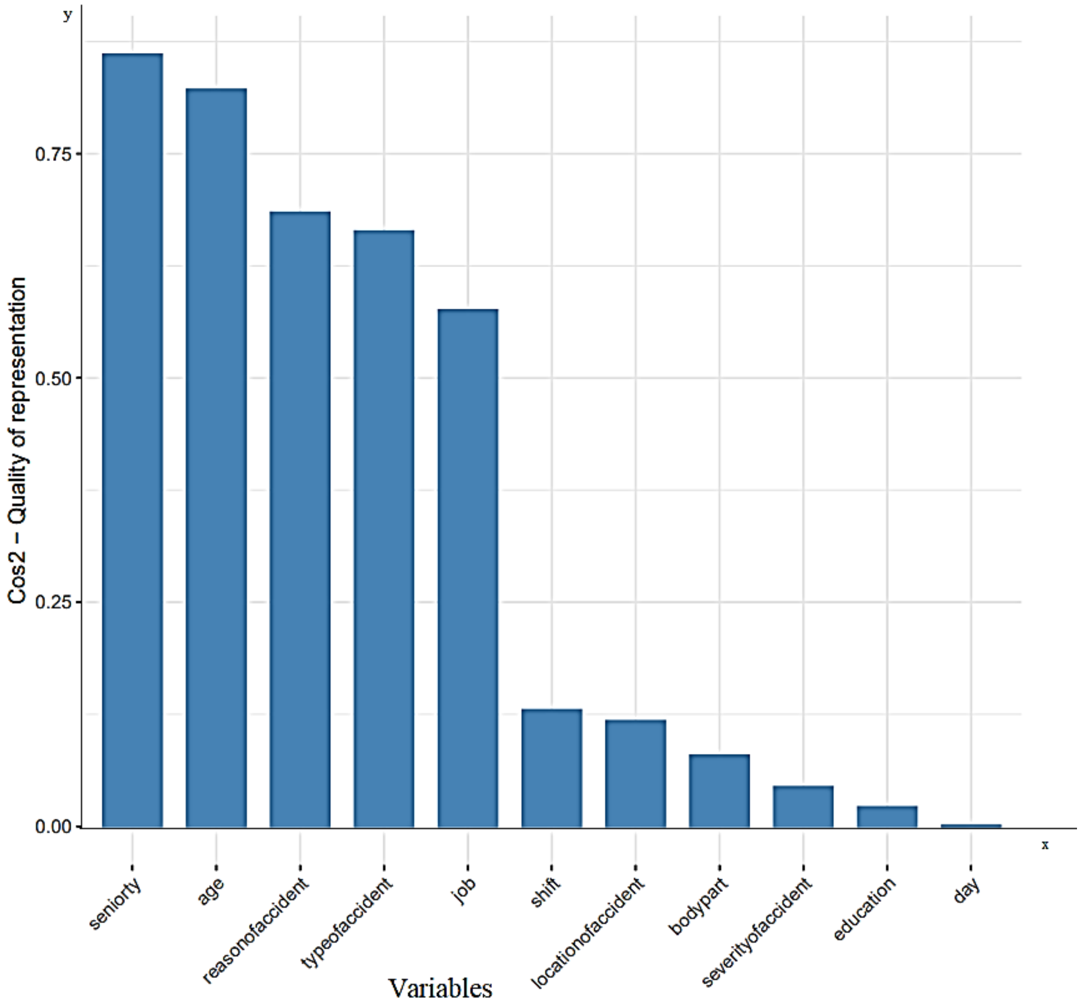


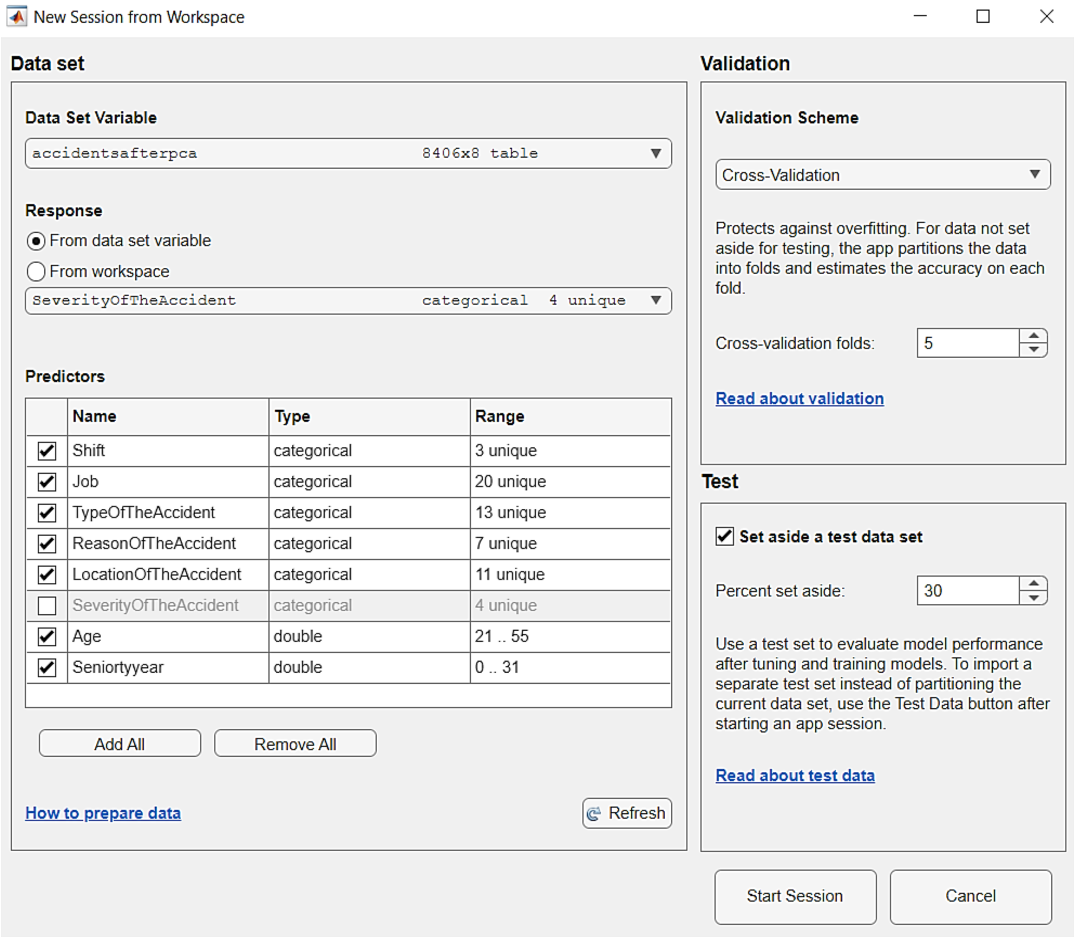
Figure 4.6 Bar plot of $\cos^2$, quality of representation of variables

4.2 Building Decision Tree Algorithm

Among the different data mining techniques, the decision tree approach was chosen since it performs classification without requiring much computation, it is able to handle both continuous and categorical variables, and is easy to interpret. After obtaining PCA results, the decision tree model was built with respect to eight components. This is eight because one of these variables, "severity of the accident" was selected for the prediction output. MATLAB, which is a programming and numeric computing platform that is used as a tool to analyze data, develop algorithms, and create models, was used to build the decision tree model.

As mentioned before, shift, job, type of accident, reason of accident, location of the accident, age, seniority and severity of the accidents were the variables in the data set. By using these variables, the model was built and the severity of the accidents were predicted.

Firstly, by using the "import data" section, the data set was introduced into the program. 30% of the data set was set aside as a test data set, while 70% of the data set was used to train the model. Splitting the data set as test and training data sets was done by using the splitting option in MATLAB to avoid controlling the data set. Then, classification learner section from the app tab was used to build the model. This time, the variables shift, job, type of the accident, reason of the accident, location of the accident, age, and seniority were used as predictors, and the variable seniority of the accident was used as a response (Figure 4.7). The next step is to decide the validation type. Cross-validation was selected to train the decision tree classifier. Because, the dataset must be divided to maximize learning and test result validity. But this is a difficult phenomenon. Cross-validation provides a bunch of techniques that divide the data in different ways. Moreover, it both protects the model against overfitting and provides a way to see the quality of the model. When the cross-validation fold number is selected as 5, the data is divided into 5 different subsets. Four subsets are used to train the data, and the final subset is left as test data. This process is repeated five times, such that each subset is used exactly once for validation. Although there isn't a rule, the most common choices for the cross-validation fold number are 5 or 10. The size gap between the training set and the resampling subsets decreases as fold number increases. The bias of the technique becomes smaller as this gap decreases (Kuhn & Johnson, 2013). Moreover, the size of the data determines the number of folds. The size is not very large because there are 8406 data in the dataset. Thus, for the model this value was selected as 5 (Figure 4.7).



Data set

Data Set Variable

accidentsafterpca 8406x8 table

Response

● From data set variable
○ From workspace

SeverityOfTheAccident categorical 4 unique

Predictors

	Name	Type	Range
☑	Shift	categorical	3 unique
☑	Job	categorical	20 unique
☑	TypeOfTheAccident	categorical	13 unique
☑	ReasonOfTheAccident	categorical	7 unique
☑	LocationOfTheAccident	categorical	11 unique
□	SeverityOfTheAccident	categorical	4 unique
☑	Age	double	21 .. 55
☑	Seniortyyear	double	0 .. 31

Validation

Validation Scheme

Cross-Validation

Protects against overfitting. For data not set aside for testing, the app partitions the data into folds and estimates the accuracy on each fold.

Cross-validation folds: 5

Test

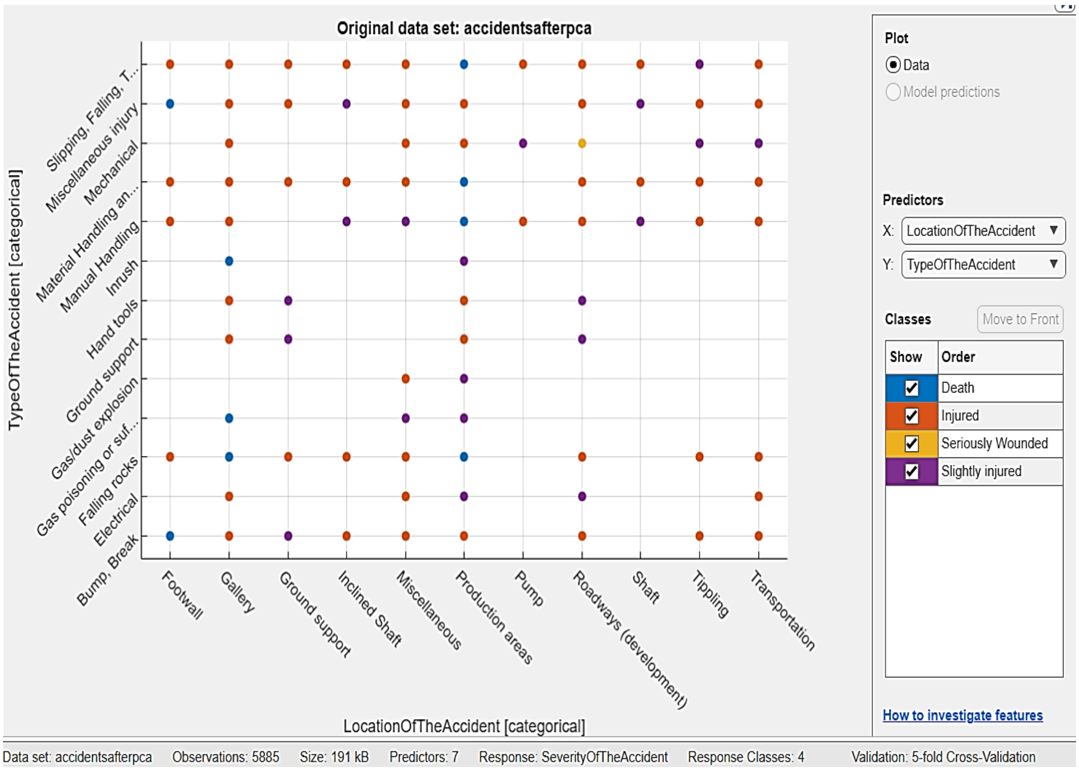
☑ Set aside a test data set

Percent set aside: 30

Use a test set to evaluate model performance after tuning and training models. To import a separate test set instead of partitioning the current data set, use the Test Data button after starting an app session.

Figure 4.7 Session information for the classification learner

Then, in order to find predictors that effectively separate classes, different pairs of predictors are plotted on the scatter plot of the original data. For example, on the scatter plot in Figure 4.8, x axes correspond to location of the accidents and y axes correspond to the type of the accidents. The points on the scatter plot show the severity of the accidents by classes. As it is seen from Figure 4.8, the classes were not well separated. Thus, it is difficult to decide which predictors are not useful for separating out classes. When the scatter plot, which was drawn with respect to other predictors, was plotted, there was no class to be removed. Therefore, training was continued with all classes. The scatter plots with respect to other predictors is presented in Appendix B.



Plot

● Data

○ Model predictions

Predictors

X: LocationOfTheAccident ▼

Y: TypeOfTheAccident ▼

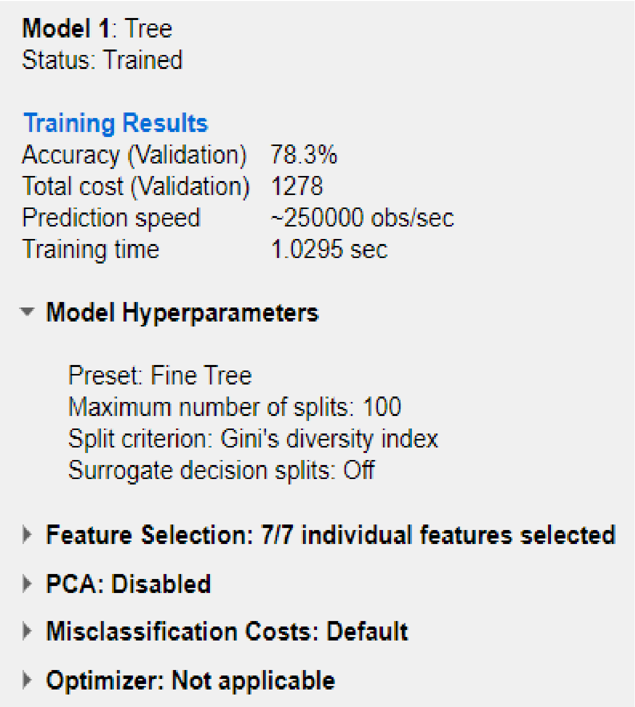
Classes

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure 4.8 Scatter plot of original data (location of accident versus type of accident)

After determining to train the model with all classes, the model was built by using the fine tree method, whose maximum number of splits is 100. The larger number of splits means that the model has more flexibility. Thus, the maximum number of splits was selected to start training the model. Gini’s diversity index was used as a split criterion. The data set had no missing values, so surrogate splits were not used. Summary of the trained model is presented in Figure 4.9. According to the training results, the accuracy of the trained model, which is the percentage of observations that are correctly classified, is 78.3%. The total cost of validation, which is the total misclassification, is 1278 out of 5885 observations. In other words, the trained model made the classification of 1278 out of 5885 observations according to the severity of the accident incorrectly.



Model 1: Tree
Status: Trained

Training Results

```
Accuracy (Validation)   78.3%
Total cost (Validation) 1278
Prediction speed        ~250000 obs/sec
Training time           1.0295 sec
```

▾ **Model Hyperparameters**

Preset: Fine Tree
Maximum number of splits: 100
Split criterion: Gini's diversity index
Surrogate decision splits: Off

▸ **Feature Selection: 7/7 individual features selected**

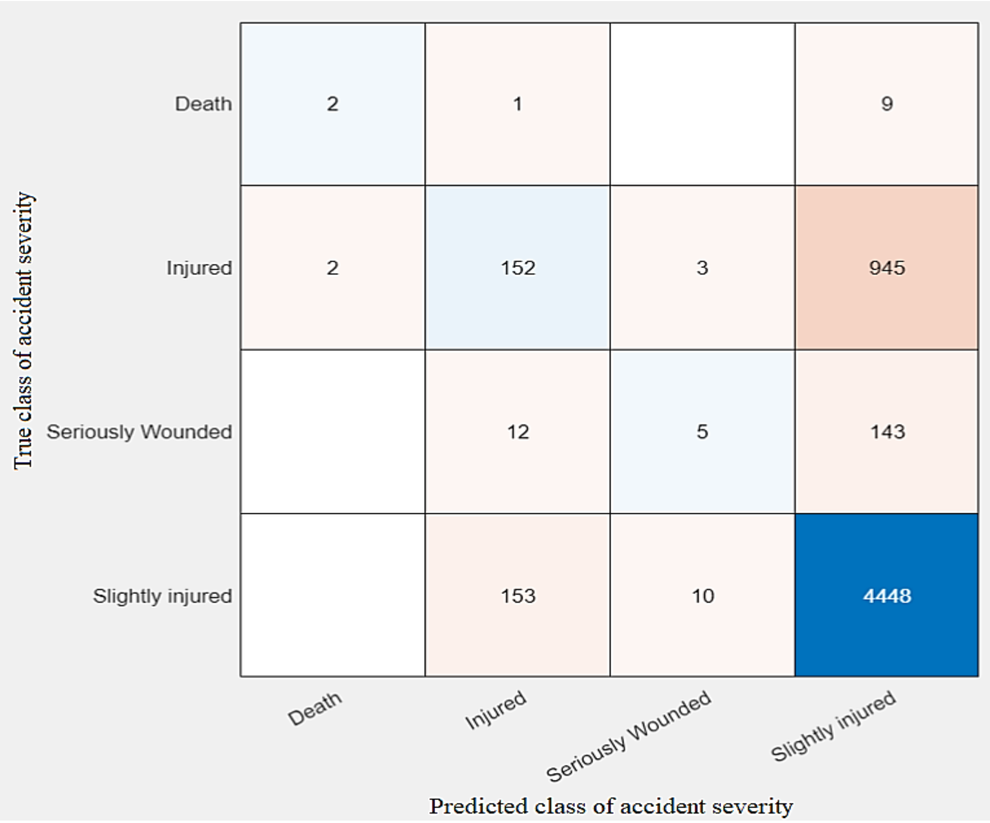
▸ **PCA: Disabled**

▸ **Misclassification Costs: Default**

▸ **Optimizer: Not applicable**

Figure 4.9 Summary of the DT model 1

Figure 4.10 shows the validation confusion matrix, which presents the performance of the currently chosen classifier in each class. In reddish cells, the true class and the predicted class do not match, whereas in bluish cells, the true class and the predicted class match well. For example, Figure 4.10 shows that 4448 samples in the dark blue cell were classified as truly. However, 945 data points are misclassified. These data points were classified as slightly injured rather than injured.



	Death	Injured	Seriously Wounded	Slightly injured
Death	2	1		9
Injured	2	152	3	945
Seriously Wounded		12	5	143
Slightly injured		153	10	4448

Figure 4.10 The validation confusion matrix of DT Model 1 (number of splits:100)

The model depth is controlled by changing the maximum number of splits. As the depth of the tree changes, the accuracy changes, too. For that reason, the second model was built by changing the number of splits to 75. The accuracy of the model did not change (78.3%). Figure C1 in Appendix C shows the results, and Figure D1 in Appendix D shows the validation confusion matrix of DT model 2. The diagonal sum of bluish cells is 4607. Namely, this model correctly classified 4607 data points.

Moreover, the models were trained by changing the number of splits to 50, 25, and 20 (DT model 3, DT model 4, and DT model 5, respectively). Figure C2, Figure C3 and Figure C4 in Appendix C show the summary results of these models. DT model 3 has the accuracy 78.2%, DT model 4 has the accuracy 78.5%, DT model 5 has the accuracy 78.4%. Trained models with a lower number of splits were not built because of decreasing accuracy.

The validation confusion matrix of DT models 3, 4, and 5 indicates that the numbers of correctly classified data points are 4604, 4618, and 4613, respectively (Figures D2, 5.3, D3). As a result, DT model 4 is the best trained model among other trained decision tree models due to its high accuracy and correct classification rate.

A visualization of the decision tree model is presented in Figure 4.11. The codes of each variable of the trained model, which were in Figure 4.11, were already given in Table 3.3 (variables of the data set) of Chapter 3. Each blue triangle in the decision tree model (Figure 4.11) corresponds to a node and a rule. The leaves, which are illustrated as blue dots, show the predicted severity of the accidents. According to this trained tree model, the first rule (top triangle) is to check whether its seniority is smaller than 3.5 or not. For example, if the seniority of the person affected by the accident is less than 3.5, and the location of the accident is a gallery, the type of accident is falling rocks, and reason of accident is personal mistakes, the severity of the accident will be predicted as "slightly injured" based on this trained decision tree model.

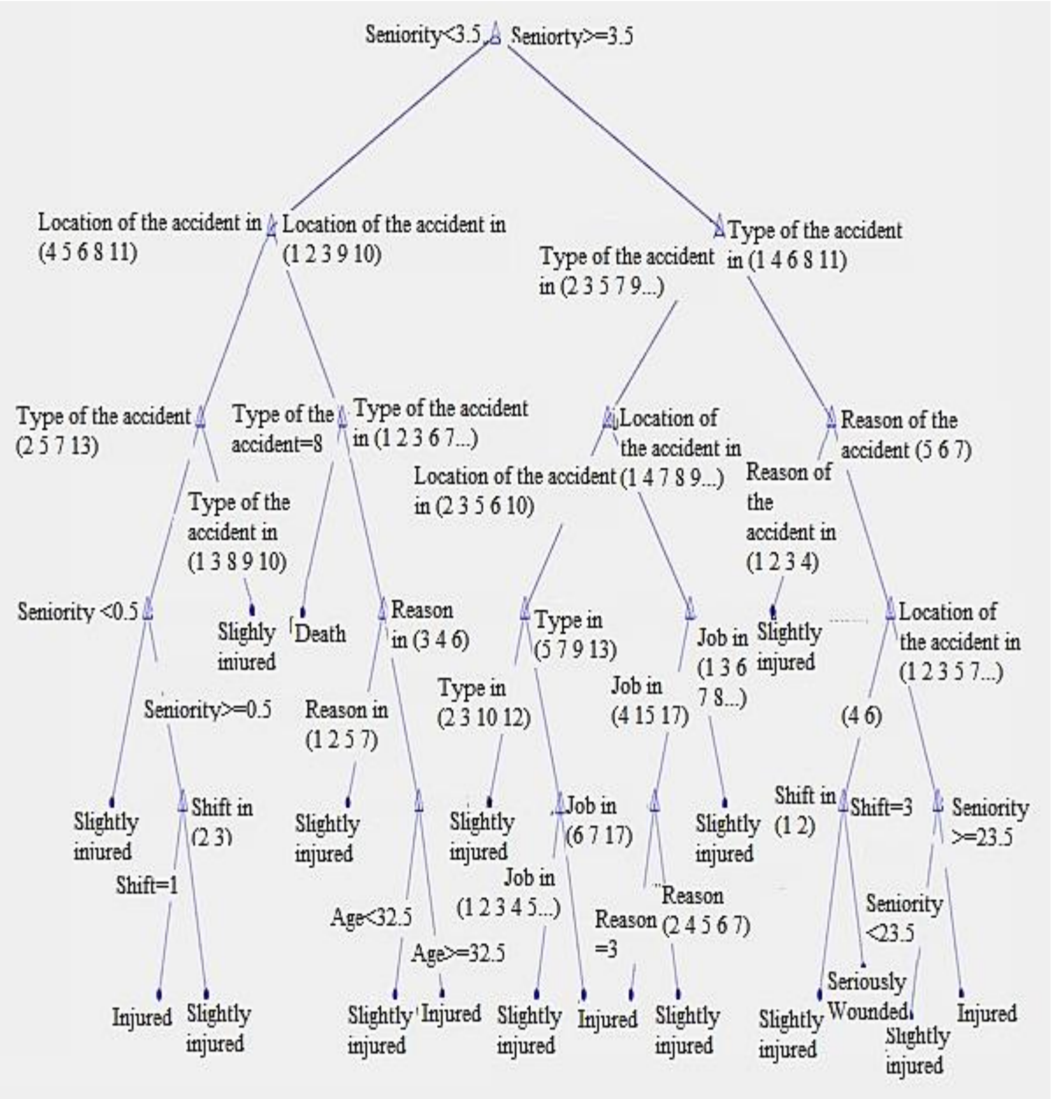


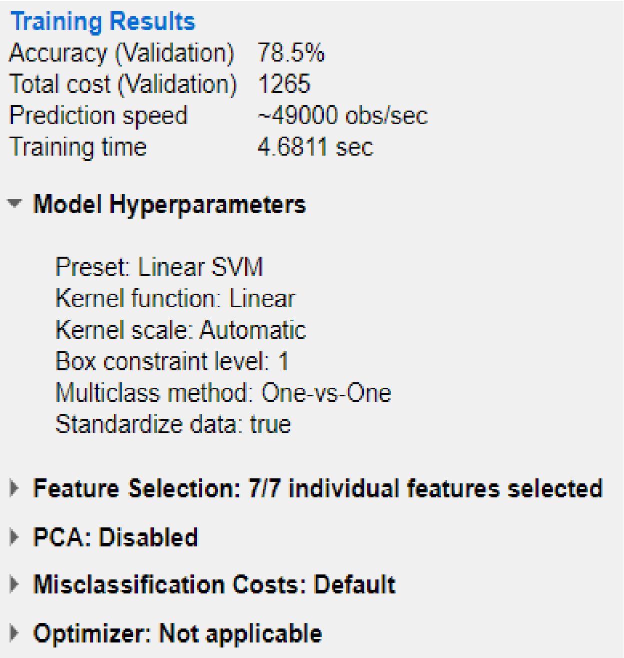
Figure 4.11 Decision tree model for accident severity prediction

4.3 Building Support Vector Machine Algorithm

In recent years, one of the most successful machine learning algorithms developed for solving classification problems is support vector machines. Because support vector machines are optimized-based, classification performance is more successful than other techniques in terms of compute complexity and usability (Nitze et al., 2012). Thus, after building a trained prediction model based on the decision tree algorithm, another prediction model was built by using the support vector machine algorithm. For the SVM trained model, again the MATLAB program, 5-fold cross validation, and the same training data used for the decision tree algorithm, were used to make a comparison.

Firstly, for the prediction model, the hyperparameters, which are kernel function, box constraint level, kernel scale mode, multiclass method, and standardize data, were decided. The types of the first hyperparameter, the kernel function, are linear, gaussian (coarse, medium, fine), quadratic, and cubic. The Kernel function provides a means of transforming the input dataset to a higher dimensional space. Individual models were designed for each function. The second parameter, box constraint level, is a parameter that regulates the maximum penalty applied to observations that violate the margin and works to prevent overfitting (MATLAB Help Center, 2022). The number of support vectors can be decreased by increasing the box constraint level. Moreover, by changing the other parameter, the kernel scale, model flexibility can be decreased. The learning method can be selected by using the multiclass method option. So, with this option, it's decided whether the model learns to distinguish one class from the other or whether it learns to distinguish one class from all others. If variables have different scales, the standardize data option should be selected to improve the fit.

As mentioned before, using the default parameters of MATLAB for the support vector machine algorithm, six models, which are linear, quadratic, cubic, fine gaussian, medium gaussian, and coarse gaussian, were built. Training results and hyperparameters of linear model are shown in Figures 4.12, and training results and hyperparameters of other support vector machine trained models are appended to Figures C5, C6, C7, C8, and C9 in Appendix C. According to the results, the fine gaussian support vector machine trained model has the highest accuracy, with a value of 78.9%.



Training Results
Accuracy (Validation) 78.5%
Total cost (Validation) 1265
Prediction speed ~49000 obs/sec
Training time 4.6811 sec

Model Hyperparameters

Preset: Linear SVM
Kernel function: Linear
Kernel scale: Automatic
Box constraint level: 1
Multiclass method: One-vs-One
Standardize data: true

Feature Selection: 7/7 individual features selected

PCA: Disabled

Misclassification Costs: Default

Optimizer: Not applicable

Figure 4.12 Summary of the linear SVM model

When the validation confusion matrix constructed in terms of predicted class versus true class of accident severity for the linear SVM model, which is presented in Figure 4.13, it is seen that the numbers of correctly classified data points are 4620. The validation confusion matrixes of the other SVM trained models are illustrated in Figures D4, D5, D6, and D7 in Appendix D and Figure 5.4 in Chapter 5. Quadratic, cubic, fine gaussian, medium gaussian, and coarse gaussian SVM trained models gave 4584, 4514, 4642, 4626, and 4607 true outputs, respectively. As a result, the fine gaussian support vector machine model is the best trained model among other support vector machine models due to its high accuracy (78.9%) and correct classification rate.

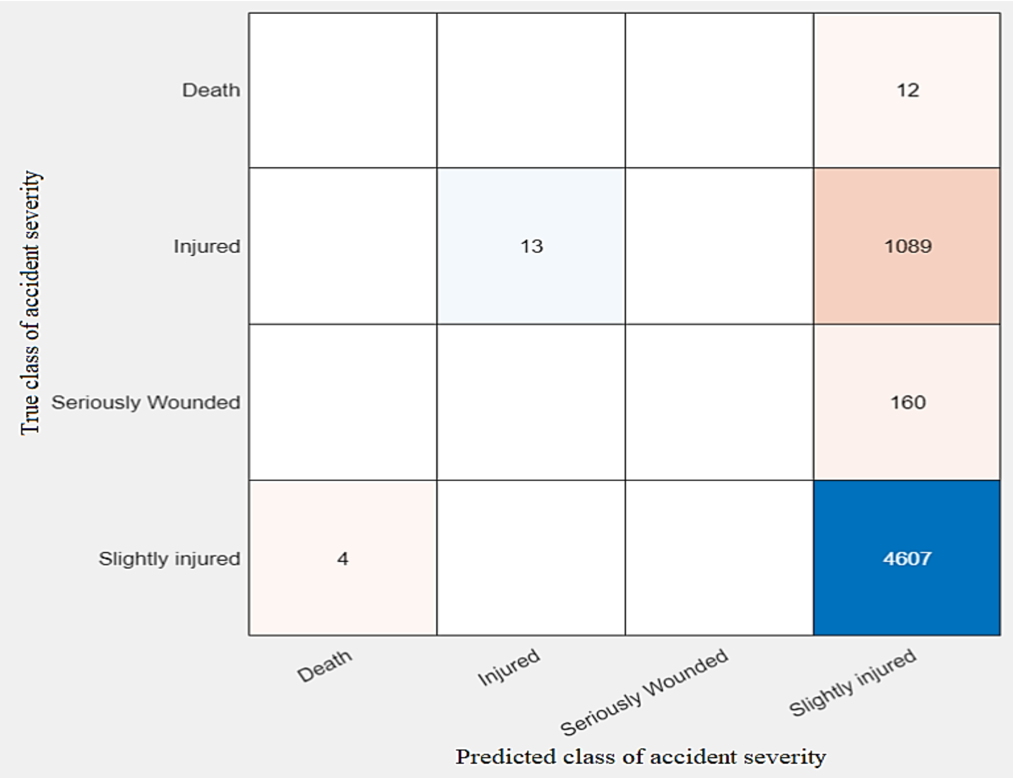


Figure 4.13 Validation confusion matrix of linear SVM model

After determining the best trained SVM model which provides the maximum accuracy for the given dataset, different kernel scales, box constraint levels, and multiclass methods were applied. The purpose of doing this is to get a better accuracy result by changing the hyperparameters of the fine gaussian model decided for the dataset. Hence, the first kernel scales were changed. When the accuracy was found at its maximum value, which was 79.1%, box constraint levels were changed. When the accuracy of the trained model began to remain unchanged, the multiclass methods were modified. The accuracies of these new trained fine gaussian SVM models and fine gaussian trained model (default parameters) are shown in Table 5.3 in Chapter 5 (Results and Discussion). Finally, the highest accuracy rate, 79.2%, was found with the hyperparameters, which were 1.5 kernel scale, 1 box constraint level, and one-vs-all multiclass methods.

According to the validation confusion matrix of fine gaussian model 7, 4659 samples were classified truly.

4.4 Building Neutral Networks

The last built prediction model is the artificial neural networks. This algorithm was chosen because it is non-linear and can be designed with input and output mappings (Haykin, 1999). For the trained model, again the MATLAB program, 5-fold cross validation (same as the decision tree algorithm and support vector machine algorithm), and the same training data that was used for the decision tree algorithm and support vector machine algorithm were used to make a comparison.

The classifier types in MATLAB Classification Learner tab are narrow neural network, medium neural network, wide neural network, bilayered neural network, and trilayered neural network. Hence, five trained models were designed by using all these classifier types, which are feedforward, fully connected neural networks for classification. The neural network classifiers have a layer structure like in Figure 4.14. A connection to the network input is made by the neural network's first fully connected layer, and there is a connection from the previous layer to each subsequent layer. The input is multiplied by a weight matrix in each fully connected layer, and a bias vector is then added. Each fully connected layer is followed by an activation function. The output of the network is produced by the final fully connected layer and the subsequent softmax activation function (Help Center, 2022).

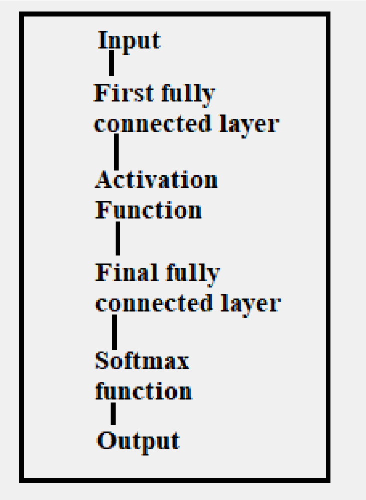
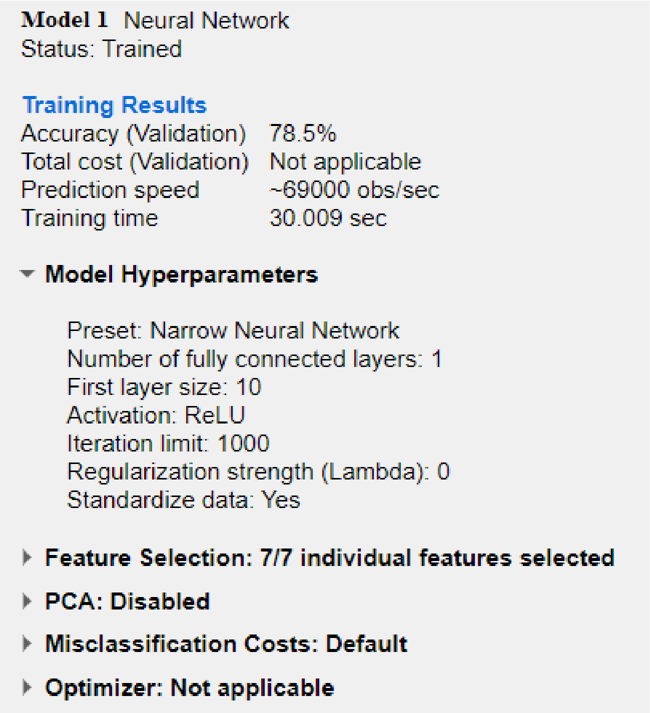


Figure 4.14 Structure of the neural network classifiers

While designing the neural network models, internal parameters play an important role in the performance of the model. These hyperparameters in MATLAB for the neural network models are the number of fully connected layers, layer sizes (first, second, and third), activation types, iteration limit, regularization strength (lambda), and standardize data. As number of fully connected layers in the neural network increases, model flexibility increases. The maximum number of fully connected layers is three in the program. The size of each fully connected layer, which is the number of outputs in the layer, can be changed by using the layer size option. If the model is created by a neural network with multiple fully connected layers, the layer sizes should be specified with decreasing sizes. Activation function is used for fully connected layer. ReLU, Tanh, None, and Sigmoid are the activation functions to be selected in the program. Softmax function, which is one of the activation functions, is always used for the final fully connected (output) layer. The softmax activation function generates a vector of probability scores using a vector of the neural network's raw outputs as input. Another hyperparameter, iteration limit, which is the maximum number of training iterations, can be specified by using the iteration limit option. Moreover, in order to prevent overfitting, regularization imposes a penalty on increasing the magnitude of parameter values. Thus, regularization strength (lambda) should be specified in determining the best fit to the data. If the value of lambda is so high, there can be underfitting. As it is mentioned at the beginning of Chapter 4.2, overfitting and underfitting can be prevented by cross validation. Hence, at the beginning of the session, the cross-validation fold was selected as 5, and so the regularization strength option was selected as 0 for the models. The last hyperparameter, standardize data, was selected for the models.

By using the default parameters of MATLAB for the narrow neural network, medium neural network, wide neural network, bilayered neural network, and trilayered neural network algorithms, five trained models were built. Training results and hyperparameters of the narrow neural network model are shown in Figure 4.15, and the summaries of other four trained models are appended to Figures C10, C11, C12, and C13 in Appendix C. According to the results, the narrow neural network method has the highest accuracy, with a value of 78.5%.



Model 1 Neural Network
Status: Trained

Training Results
Accuracy (Validation) 78.5%
Total cost (Validation) Not applicable
Prediction speed ~69000 obs/sec
Training time 30.009 sec

▾ Model Hyperparameters

Preset: Narrow Neural Network
Number of fully connected layers: 1
First layer size: 10
Activation: ReLU
Iteration limit: 1000
Regularization strength (Lambda): 0
Standardize data: Yes

▸ Feature Selection: 7/7 individual features selected

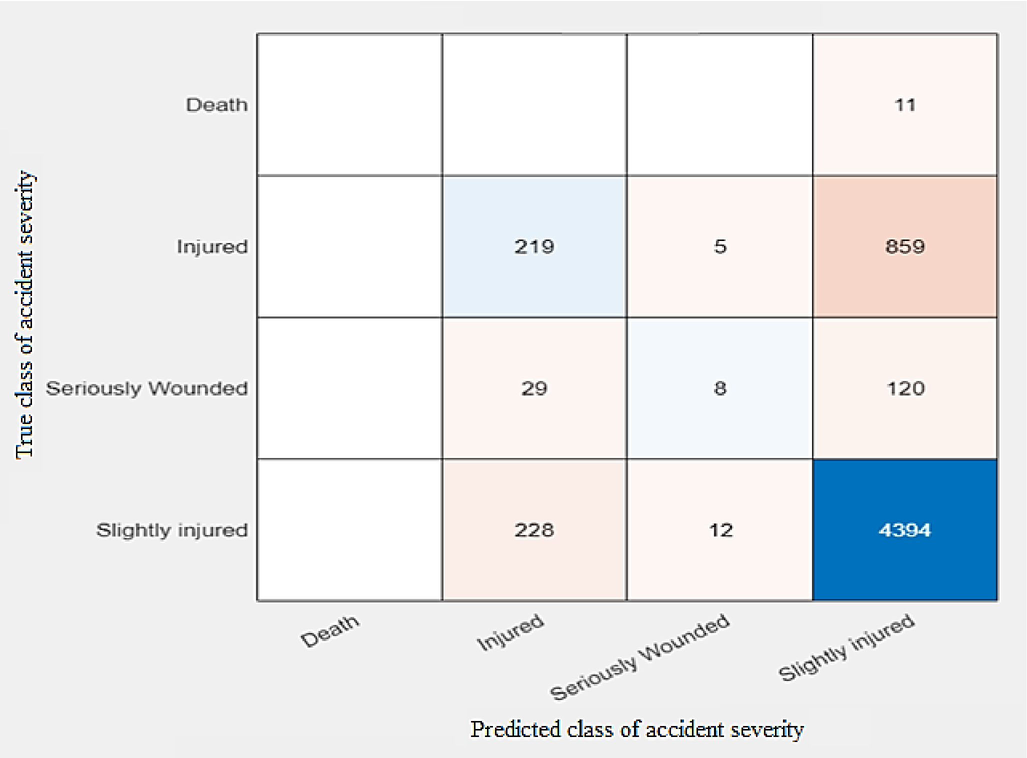
▸ PCA: Disabled

▸ Misclassification Costs: Default

▸ Optimizer: Not applicable

Figure 4.15 Summary of narrow neural network model

According to the validation confusion matrixes of the trained models, which are shown in Figures 4.16, D8, D9, D10, and D11, it is seen that the numbers of correctly classified data points are 4621, 4512, 4386, 4569, and 4521, respectively. As a result, the narrow neural network trained model is the best trained model among other neural network models due to its high accuracy and correct classification rate.



True class of accident severity	Death	Injured	Seriously Wounded	Slightly injured
Death				11
Injured		219	5	859
Seriously Wounded		29	8	120
Slightly injured		228	12	4394

Predicted class of accident severity

Figure 4.16 Validation Confusion Matrix of narrow neural network model

After determining the best neural network trained model (the narrow neural network) for the dataset, different numbers of fully connected layers, layer sizes, and activation functions were applied to further increase the current accuracy rate. Firstly, the first layer sizes were changed. But both increasing and decreasing the sizes caused the decrease in accuracy. Then, number of fully connected layers were changed. When the number of fully connected layer sizes was selected as 2 and 3, the first, second, and third layer sizes were given value as decreasing. But again, the accuracy of the model decreased. Finally, different activation functions and no activation function were used. The accuracy results were lower than for a narrow neural network trained model with MATLAB's default hyperparameters. The accuracies of these new trained narrow neural network models with changed parameters and the narrow network trained model (default hyperparameters) are shown in Chapter 5 (Results and Discussion). Finally, the highest accuracy rate, 78.5%, was found with the default hyperparameters, which were 1 fully connected layer, 10 outputs in the layer (layer size), and ReLU activation function.

CHAPTER 5

RESULTS AND DISCUSSION

In this study, 8406 underground accident data belonging to Turkish Hard Coal Enterprise Amasra, Armutçuk, Karadon, Kozlu, Üzülmez district were used to build an accident severity prediction model for underground coal mines by using decision tree, support vector machine, and neural network algorithms. The data covers the period between March 2008 and December 2010.

The main results drawn from this study can be listed as:

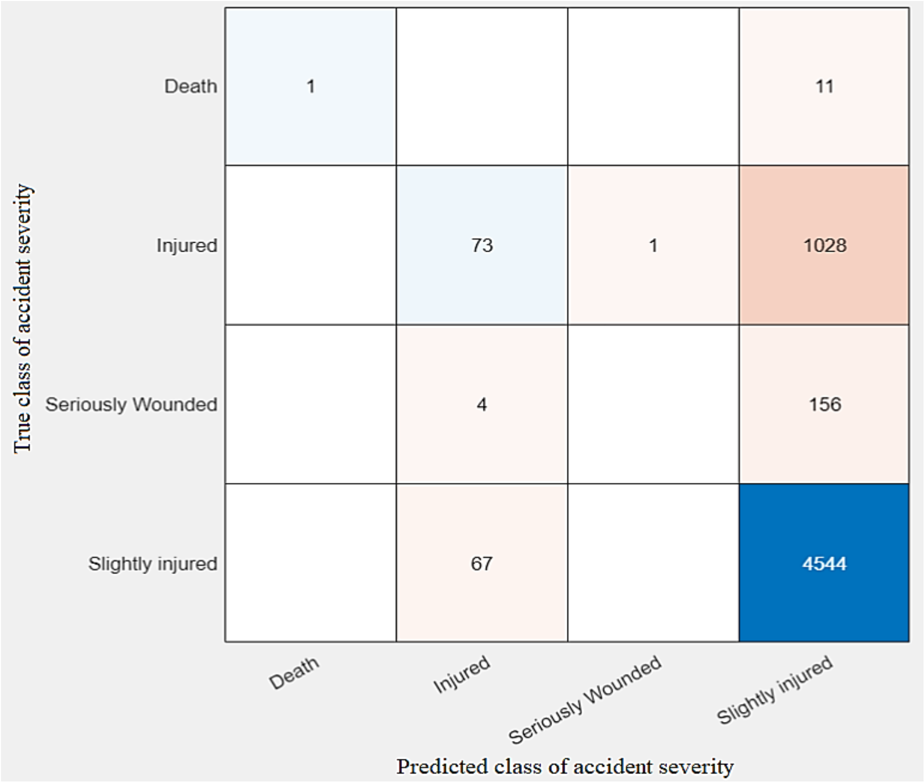
- The data set had eleven variables (dimensions) that are shift, day of the accident, job, education, type of the accident, reason of the accident, location of the accident, severity of the accident, age, seniority, affected body part. When the basic statistics of the data set were examined, primary school graduates (54.26% of the 8406 accidents) and production workers (80.25%), with experience ranging from 0 to 7 years (72.22%) and ages ranging from 27 to 30 years (35.82%), had the highest rate of encountering an accident in the production area (52.58%) in the first shift (08:00-16:00, 53.81%) on Mondays (18.34%) due to falling rock (88.52%) and personal mistakes (46.04%), and the most injured body parts were the hand fingers (29.26%) as a result of these accidents.
- After analyzing and preparing the data, principal component analysis was applied first. Since this analysis provides reduction of the dimensions, it helps to express the data in fewer variables that are meaningful and easier to explain. As a result of the analysis, 81.82% (cumulative variance percent) of the data can be interpreted with the seven components. Thus, the accident data set has been converted into a meaningful and reduced data set of 8406

accident events and seven variables, which are seniority, age, job, type of accident, reason of the accident, location of the accident, and shift.

- After the PCA results, the decision tree models were built with respect to eight components. This is eight because one of these variables, "severity," was selected for the prediction output. For this algorithm, the MATLAB program, 5-fold cross validation method was used. 30% (2521 data) of the total data was used to test the model, and 70% (5885 data) of the total data was used to train the model. By changing the number of splits, five decision tree models were designed. Table 5.1 shows the results of models build by decision tree algorithm. As a result, 25-splits decision tree model 4 is the best model among them due to its high accuracy (78.5%) and correct classification rate (Figure 5.1). After selecting the DT Model 4 as the best trained prediction model for the decision tree algorithm, the test data was imported into the program for the evaluation. When the trained prediction model was run with test data, which is the accuracy of a model on examples it hasn't seen, the accuracy of the test was 78.1% (Figure 5.2), and the number of correct classifications was 1969 out of 2521 observations.

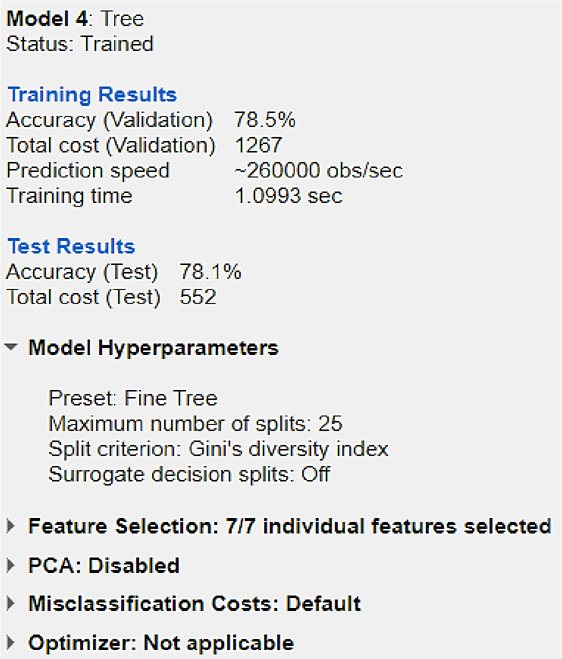
Table 5.1 Results of decision tree models

Model Name	**Number of splits**	**Accuracy of the Model (%)**	**The numbers of correctly classified data point out of 5885 observations**
DT Model 1	100	78.3	4607
DT Model 2	75	78.3	4607
DT Model 3	50	78.2	4604
DT Model 4	25	**78.5**	4618
DT Model 5	20	78.4	4613



	Death	Injured	Seriously Wounded	Slightly injured
Death	1			11
Injured		73	1	1028
Seriously Wounded		4		156
Slightly injured		67		4544

Figure 5.1 Validation confusion matrix of the trained DT model 4 (25-splits)



Model 4: Tree
Status: Trained

Training Results

Accuracy (Validation)	78.5%
Total cost (Validation)	1267
Prediction speed	~260000 obs/sec
Training time	1.0993 sec

Test Results

Accuracy (Test)	78.1%
Total cost (Test)	552

▼ **Model Hyperparameters**

Preset: Fine Tree
Maximum number of splits: 25
Split criterion: Gini's diversity index
Surrogate decision splits: Off

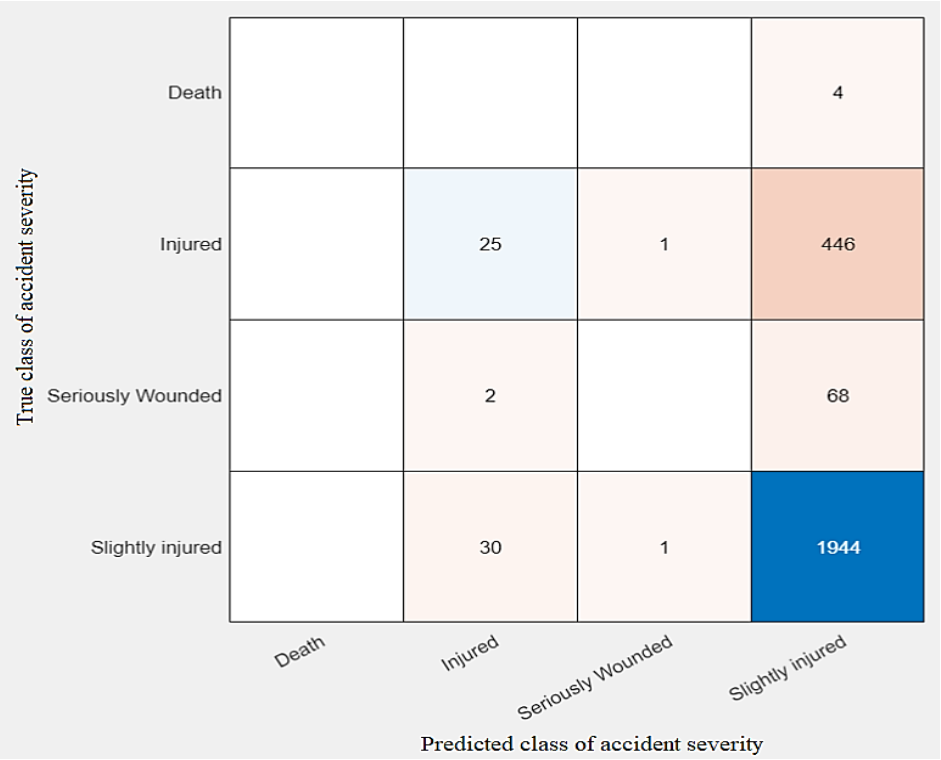
▶ **Feature Selection: 7/7 individual features selected**

▶ **PCA: Disabled**

▶ **Misclassification Costs: Default**

▶ **Optimizer: Not applicable**

Figure 5.2 Summary of the trained DT model 4 with test data



	Death	Injured	Seriously Wounded	Slightly injured
Death				4
Injured		25	1	446
Seriously Wounded		2		68
Slightly injured		30	1	1944

Figure 5.3 Test confusion matrix of the trained DT Model 4

- From the decision tree model, the prediction model first checks the seniority. If the seniority is less than 3.5 years, the next decision rule is to look at where the accident occurred. However, if the seniority is greater than or equal to 3.5 years, the second control point is the type of the accident. The tree divided the age check point into two. If the age of the person is less than 32.5, the severity of the accident will be slightly injured. If the person is older than or equal to 32.5 years old, the severity of the accident is estimated as injured.
- Another prediction model was built next by using the support vector machine algorithm. For this algorithm, the MATLAB program, 5-fold cross validation (same as the decision tree algorithm), and the same training data that was used for the decision tree algorithm were used to make a comparison. By using the default parameters of MATLAB for the support vector machine algorithm, six models, which are linear, quadratic, cubic, fine gaussian, medium gaussian, and coarse gaussian, were built. The box constraint level

was 1, the multiclass method was one-vs-one, and data standardization was true for all method types. According to the results (Table 5.2), the fine gaussian model has the highest accuracy with a value of 78.9% and correct classification rate (Figure 5.4).

Table 5.2 Results of support vector machine models

Model Name	**Type of Support Vector Machine Model**	**Kernel Function**	**Kernel Scale**	**Accuracy of the Model (%)**	**The numbers of correctly classified data point out of 5885 observations**
Linear Model	Linear	Linear	Automatic	78.5	4620
Quadratic Model	Quadratic	Quadratic	Automatic	77.9	4584
Cubic Model	Cubic	Cubic	Automatic	76.7	4514
Fine Gaussian Model	Fine Gaussian	Gaussian	0.66	**78.9**	4642
Medium Gaussian Model	Medium Gaussian	Gaussian	2.6	78.6	4626
Coarse Gaussian Model	Coarse Gaussian	Gaussian	11	78.3	4607

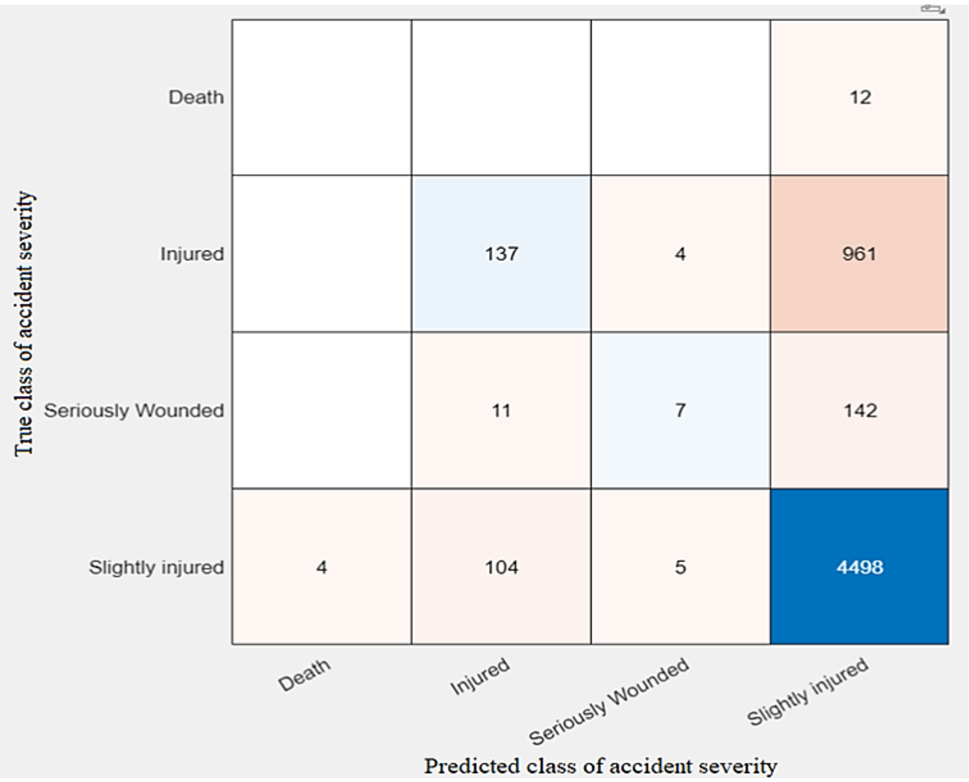
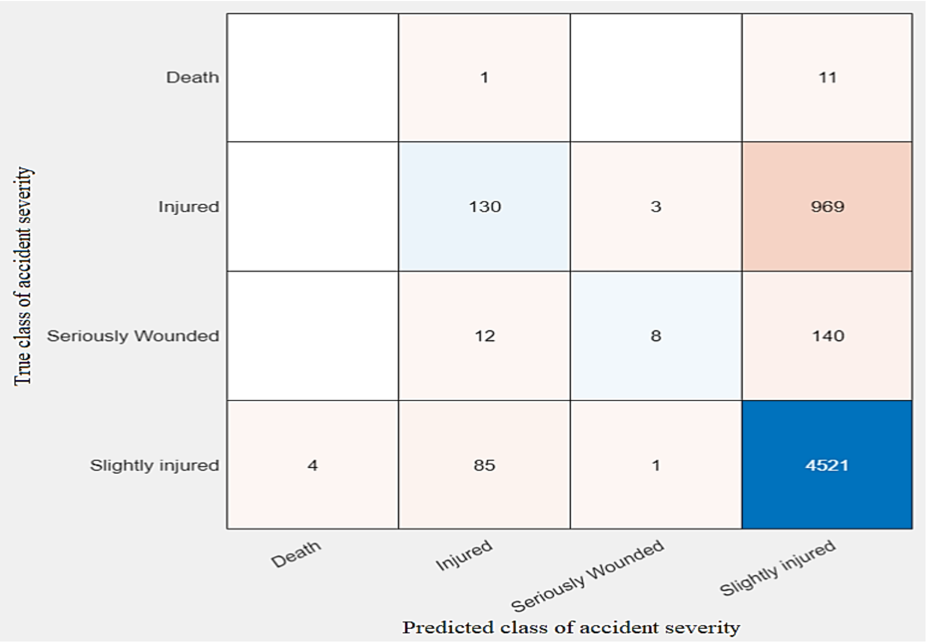


Figure 5.4 Validation confusion matrix of fine gaussian SVM model

After determining the best support vector machine model type for the dataset as fine gaussian, to increase the accuracy, different kernel scales, box constraint levels, and multiclass methods were applied. The highest accuracy rate (Table 5.3), 79.2%, was found with the hyperparameters, which were 1.5 kernel scale, 1 box constraint level, and one-vs-all multiclass methods. As a result, Fine Gaussian Model 7 is the best trained prediction model among others due to its high accuracy and correct classification rate (Figure 5.5).

Table 5.3 Results of the fine gaussian SVM models with different hyperparameters

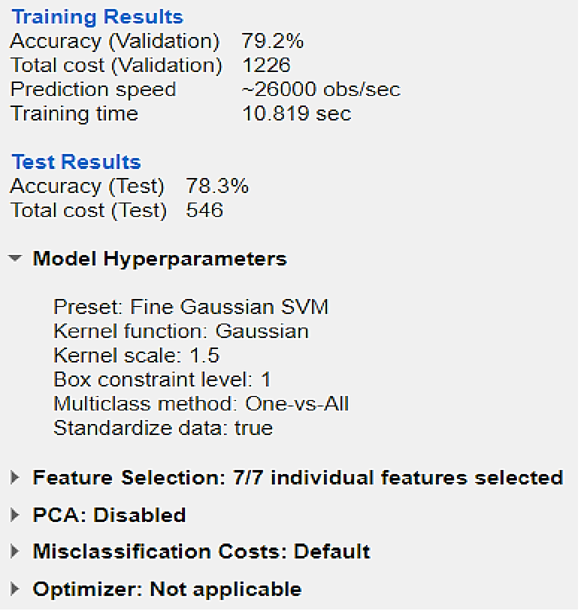
Model Name	**Kernel Scales**	**Box Constraint Levels**	**Multiclass Methods**	**Accuracy of the Model (%)**
Fine Gaussian Model	0.66	1	One-vs-One	78.9
Fine Gaussian Model 1	1	1	One-vs-One	79.0
Fine Gaussian Model 2	1.5	1	One-vs-One	79.1
Fine Gaussian Model 3	2	1	One-vs-One	78.8
Fine Gaussian Model 4	1.5	2	One-vs-One	78.8
Fine Gaussian Model 5	1.5	3	One-vs-One	78.9
Fine Gaussian Model 6	1.5	4	One-vs-One	78.9
Fine Gaussian Model 7	1.5	1	One-vs-All	**79.2**
Fine Gaussian Model 8	1.5	2	One-vs-All	78.8
Fine Gaussian Model 9	1.5	3	One-vs-All	78.7



True class of accident severity	Death	Injured	Seriously Wounded	Slightly injured
Death		1		11
Injured		130	3	969
Seriously Wounded		12	8	140
Slightly injured	4	85	1	4521

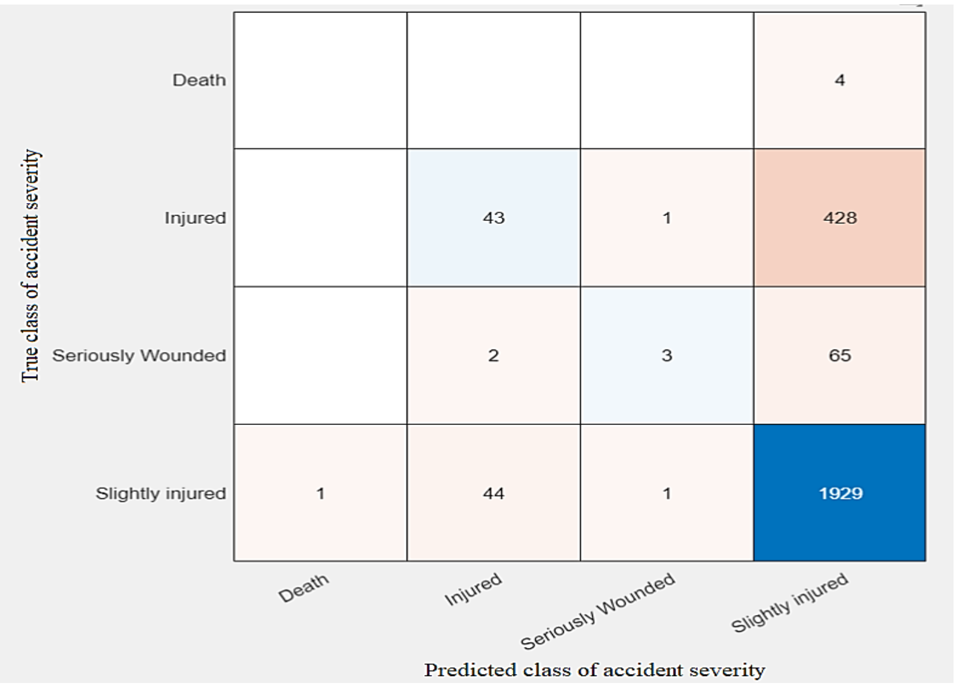
Figure 5.5 Validation confusion matrix of fine gaussian model 7

After selecting fine gaussian model 7 as the best trained prediction model for the support vector machine algorithm, the test data was imported into the program for the evaluation. When the trained prediction model was run with test data, the accuracy of the test was 78.3% (Figure 5.6), and the number of correct classifications was 1975 out of 2521 observations (Figure 5.7).



```
Training Results
Accuracy (Validation)   79.2%
Total cost (Validation) 1226
Prediction speed        ~26000 obs/sec
Training time           10.819 sec

Test Results
Accuracy (Test)   78.3%
Total cost (Test) 546

▾ Model Hyperparameters

    Preset: Fine Gaussian SVM
    Kernel function: Gaussian
    Kernel scale: 1.5
    Box constraint level: 1
    Multiclass method: One-vs-All
    Standardize data: true

▸ Feature Selection: 7/7 individual features selected
▸ PCA: Disabled
▸ Misclassification Costs: Default
▸ Optimizer: Not applicable
```

Figure 5.6 Summary of fine gaussian model 7 with test data



True class of accident severity	Death	Injured	Seriously Wounded	Slightly injured
Death				4
Injured		43	1	428
Seriously Wounded		2	3	65
Slightly injured	1	44	1	1929

Predicted class of accident severity

Figure 5.7 Test confusion matrix of fine gaussian model 7

- The last prediction model was built using artificial neural networks. For this algorithm, the MATLAB program, 5-fold cross validation (same as the decision tree algorithm and support vector machine algorithm), and the same training data that was used for the decision tree algorithm and support vector machine algorithm were used to make a comparison. By using the default parameters of MATLAB for the neural network algorithms, five models, which are narrow neural network, medium neural network, wide neural network, bilayered neural network, and trilayered neural network, were built. According to the results (Table 5.4), the narrow neural network method has the highest accuracy with a value of 78.5% and correct classification numbers (Figure 4.16).

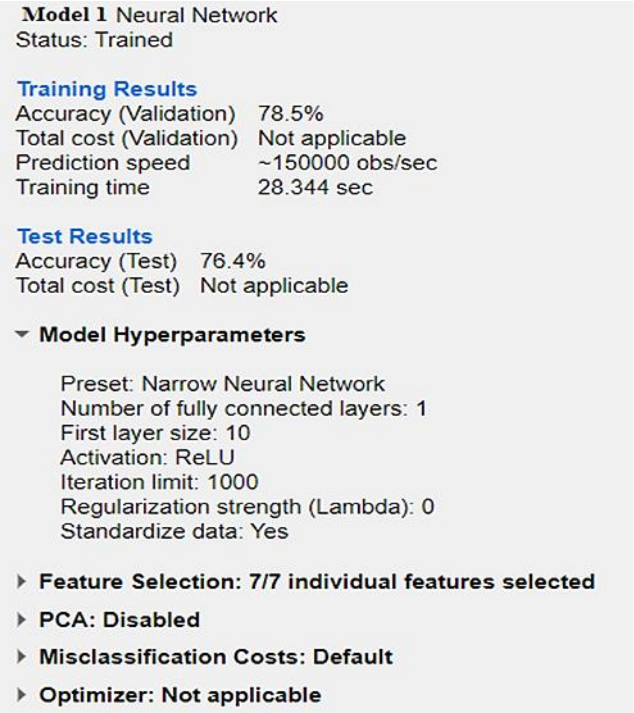
Table 5.4 Results of neural network models

Model Name	**Type of Neural Network Model**	**Number of Fully Connected Layers**	**First Layer Size**	**Second Layer Size**	**Third Layer Size**	**Accuracy of the Model (%)**
Narrow NN Model	Narrow NN	1	10	-	-	**78.5**
Medium NN Model	Medium NN	1	25	-	-	76.7
Wide NN Model	Wide NN	1	100	-	-	74.5
Bilayered NN Model	Bilayered NN	2	10	10	-	77.6
Trilayered NN Model	Trilayered NN	3	10	10	10	76.8

After determining the best neural network model type (Narrow NN Model) for the dataset, to increase the accuracy, different number of fully connected layers, layer sizes, and activation functions were applied (Table 5.5). As a result, the highest accuracy rate with a value of 78.5%, was found with the default hyperparameters of Narrow NN Model, which were 1 fully connected layer, 10 outputs in the layer (layer size), and ReLU activation function. Finally, the test data was imported into the program for the evaluation of this neural network trained prediction model. When the trained prediction model was run with test data, the accuracy of the test was 76.4% (Figure 5.9), and the number of correct classifications was 1921 out of 2521 observations.

Table 5.5 Results of the Narrow NN models with different hyperparameters

Models	**Number of Fully Connected Layers**	**First Layer Size**	**Second Layer Size**	**Third Layer Size**	**Activation Function**	**Accuracy of the Model (%)**
NNN Model (Default parameters)	1	10	-	-	ReLU	**78.5**
NNN Model 1	1	5	-	-	ReLU	78.0
NNN Model 2	1	20	-	-	ReLU	77.2
NNN Model 3	2	10	5	-	ReLU	77.0
NNN Model 4	2	6	3	-	ReLU	78.1
NNN Model 5	3	7	5	3	ReLU	78.2
NNN Model 6	1	10	-	-	Tanh	77.7
NNN Model 7	1	10	-	-	Sigmoid	77.1
NNN Model 8	1	10	-	-	None	78.2



```
Model 1 Neural Network
Status: Trained

Training Results
Accuracy (Validation)   78.5%
Total cost (Validation) Not applicable
Prediction speed        ~150000 obs/sec
Training time           28.344 sec

Test Results
Accuracy (Test)   76.4%
Total cost (Test) Not applicable

▾ Model Hyperparameters

    Preset: Narrow Neural Network
    Number of fully connected layers: 1
    First layer size: 10
    Activation: ReLU
    Iteration limit: 1000
    Regularization strength (Lambda): 0
    Standardize data: Yes

▸ Feature Selection: 7/7 individual features selected
▸ PCA: Disabled
▸ Misclassification Costs: Default
▸ Optimizer: Not applicable
```

Figure 5.8 Summary of the narrow neural network model with test data

- As it was mentioned, when this trained prediction model was run with test data, the accuracy of the test was 78.3%, and the number of correct classifications was 1975 out of 2521 observations (Figure 5.7).
- Based on this test result, the dominant correct classification severity type is slightly injured (Table 5.10). The trained prediction model correctly classified this class of test data by 78.4%. Moreover, the test results show that the trained prediction model makes the most accurate classification of accident severity with an accuracy rate of 89.82% at the location of the gallery. Moreover, as can be seen from the test results, the model remains weak in predicting deaths. This is because there are few examples of fatal accidents in the dataset, so the data is insufficient when training the model. Because the number of other accident outcomes is higher, the model is better at predicting other accident outcomes such as injured, slightly injured, and seriously wounded.

Table 5.6 Part of the Test Data Prediction Results

Accident Number	Shift	Job	Type of The Accident	Reason of The Accident	Location of The Accident	Age	Seniority	Severity of The Accident	Prediction Result
1	Third	Production Worker	Falling rocks	Coal transfer	Production area	29	1	Slightly injured	Slightly injured
2	First	Haulage Worker	Mechanical	Transportation vehicles	Gallery	39	13	Slightly injured	Slightly injured
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
78	First	Production Worker	Falling rocks	Personal mistake	Production area	23	1	Slightly injured	Slightly injured
79	Third	Production Worker	Falling rocks	Geological conditions	Gallery	29	1	Injured	Slightly injured
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
566	Second	Preparatory Worker	Falling rocks	Geological conditions	Gallery	37	10	Seriously Wounded	Slightly injured
567	Second	Maintenance and Repair Worker	Material Handling and Usage	Other	Gallery	32	10	Injured	Injured
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
1524	First	Production Worker	Slipping Falling Tripping Ankle Sprain	Personal mistake	Production area	26	4	Injured	Injured
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
2521	First	Production Worker	Material Handling and Usage	Personal mistake	Gallery	31	1	Slightly injured	Slightly injured

- The minimum classification error plots for the best trained prediction models selected for DT, SVM, and NN algorithms are shown in Figures 5.9, 5.10, and 5.11, respectively. In the plots, light blue dots show the estimated minimum classification errors, and observed minimum classification errors are represented as dark blue dots. Yellow point shows the minimum error rate. Although the trained DT (25 splits) and NN (narrow) models have the same accuracy rate, their observed minimum classification errors are different. Because the classification error plot shows the error rates of the training by iteration numbers sensitively. The trained NN model has a smaller observed minimum classification error (0.21490) than the trained DT model (0.21495). The trained DT model has the lowest classification error rate in the ninth iteration, while the trained NN model finds the lowest error rate in the third iteration. Among the three machine learning algorithms, the trained SVM (fine gaussian model 7) model has the lowest minimum classification error of 0.20798.

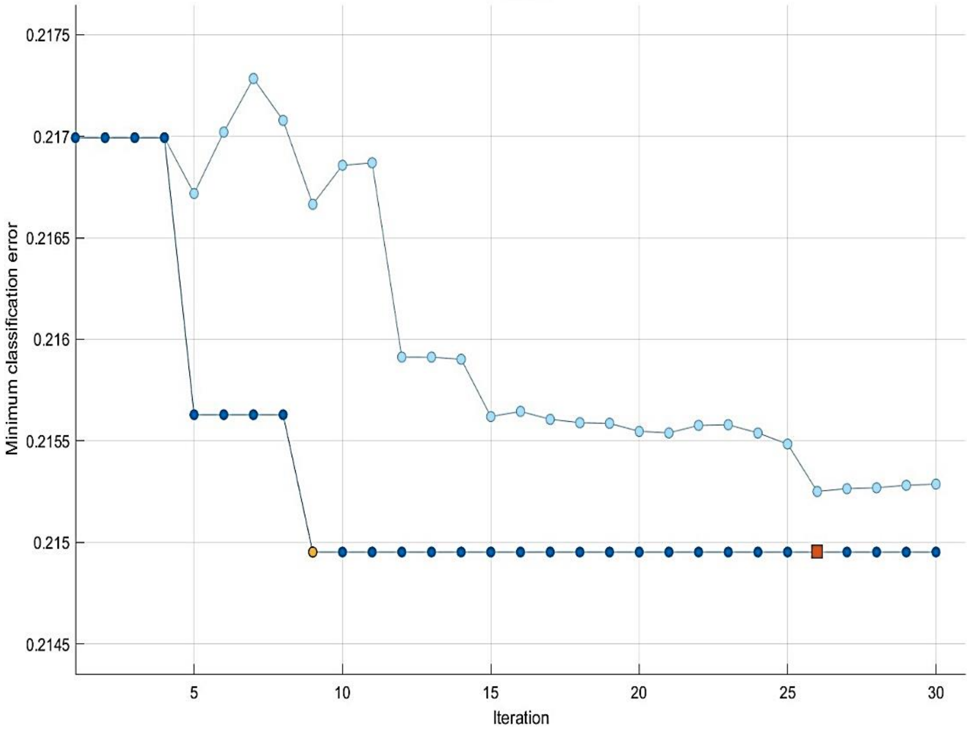


Figure 5.9 The minimum classification error plot of trained DT model

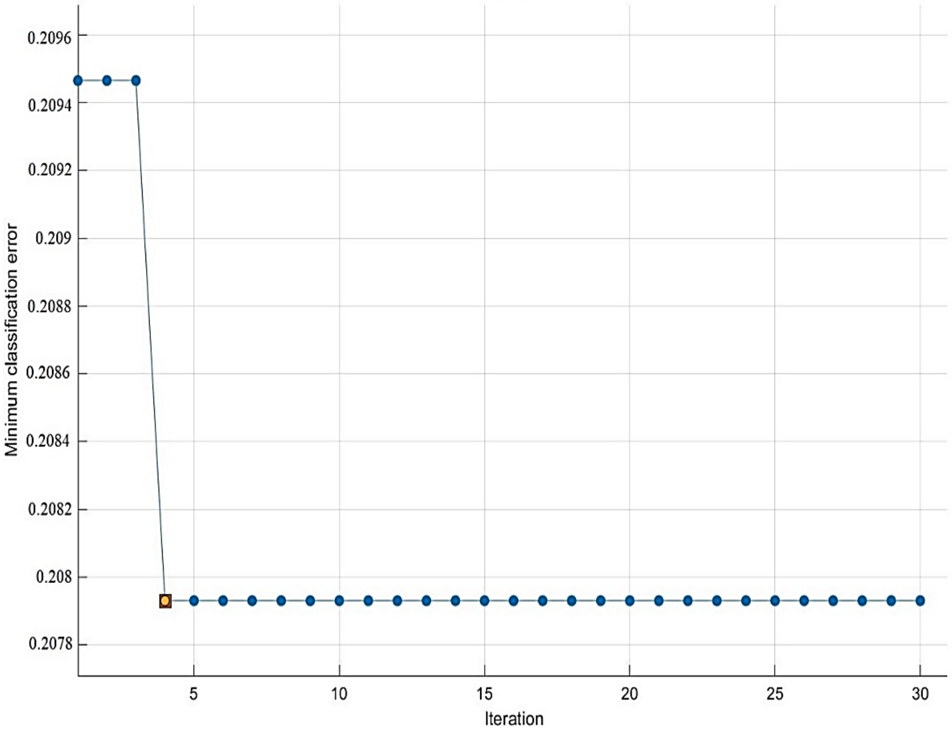


Figure 5.10 The minimum classification error plot of trained SVM model

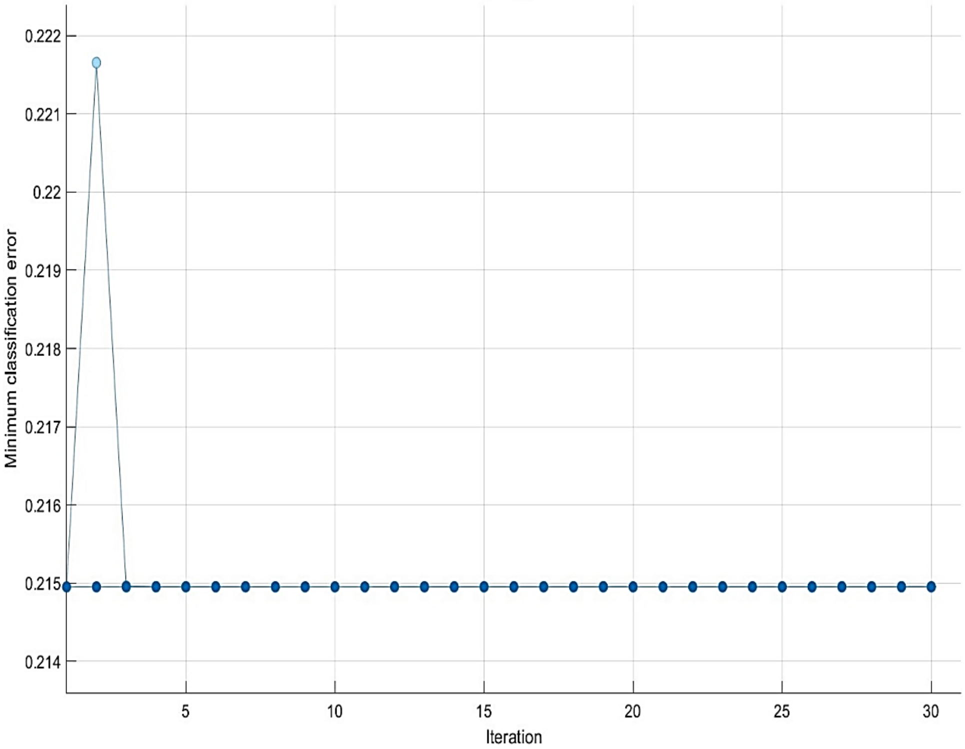
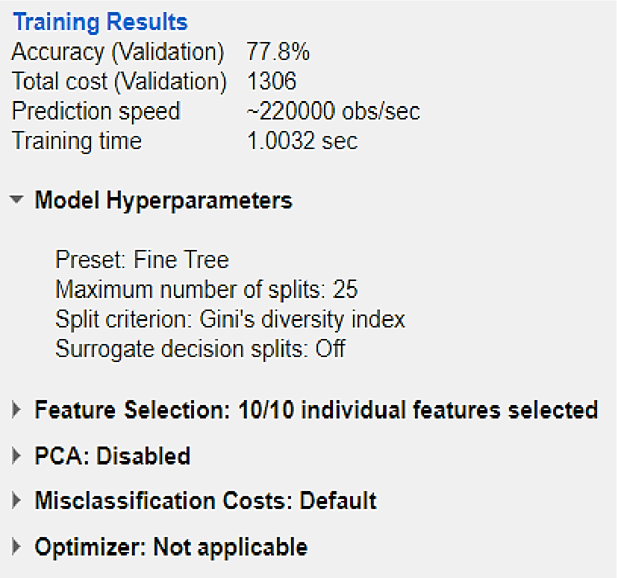


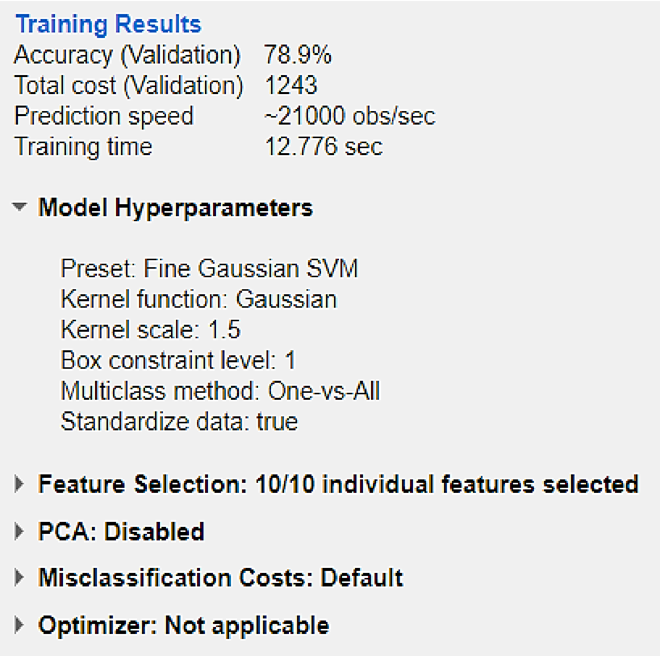
Figure 5.11 The minimum classification error plot of trained NN model

- Three machine learning analyses were conducted using raw data to see the impact of pre-analysis implementation of principal component analysis. 8406 accident data and eleven dimensions were used, and the hyperparameters that gave the best results in the decision tree (25-splits), support vector machine (fine gaussian model 7), and neural network (narrow neural network) analysis were selected. Figure 5.12, 5.13, and 5.14 show the training results of DT, SVM and NN prediction models without PCA. The accuracies of the DT, SVM, and NN trained prediction models are 77.8%, 78.9%, and 75.1%, respectively. These validation rates were 78.5%, 79.2%, and 78.5% for DT, SVM, and NN trained models with PCA, respectively. The accuracy rates of all three trained models are lower than the accuracy rates of the trained models designed after PCA is applied. This shows that principal component analysis is effective in increasing the prediction success of the trained model.



```
Training Results
Accuracy (Validation)   77.8%
Total cost (Validation) 1306
Prediction speed        ~220000 obs/sec
Training time           1.0032 sec

▼ Model Hyperparameters

    Preset: Fine Tree
    Maximum number of splits: 25
    Split criterion: Gini's diversity index
    Surrogate decision splits: Off

▶ Feature Selection: 10/10 individual features selected
▶ PCA: Disabled
▶ Misclassification Costs: Default
▶ Optimizer: Not applicable
```

Figure 5.12 Training results of DT without PCA



Training Results

```
Accuracy (Validation)   78.9%
Total cost (Validation) 1243
Prediction speed        ~21000 obs/sec
Training time           12.776 sec
```

▾ **Model Hyperparameters**

Preset: Fine Gaussian SVM
Kernel function: Gaussian
Kernel scale: 1.5
Box constraint level: 1
Multiclass method: One-vs-All
Standardize data: true

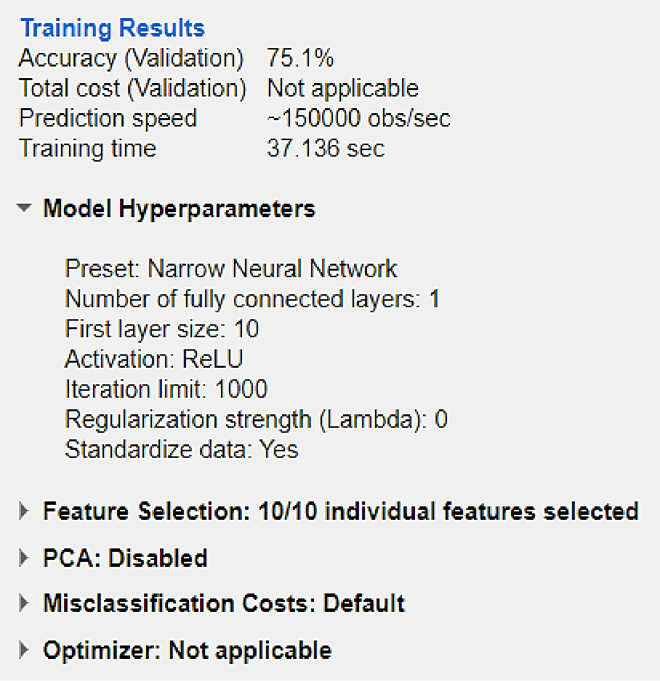
▸ **Feature Selection: 10/10 individual features selected**

▸ **PCA: Disabled**

▸ **Misclassification Costs: Default**

▸ **Optimizer: Not applicable**

Figure 5.13 Training results of SVM without PCA



Training Results

```
Accuracy (Validation)   75.1%
Total cost (Validation) Not applicable
Prediction speed        ~150000 obs/sec
Training time           37.136 sec
```

▾ **Model Hyperparameters**

Preset: Narrow Neural Network
Number of fully connected layers: 1
First layer size: 10
Activation: ReLU
Iteration limit: 1000
Regularization strength (Lambda): 0
Standardize data: Yes

▸ **Feature Selection: 10/10 individual features selected**

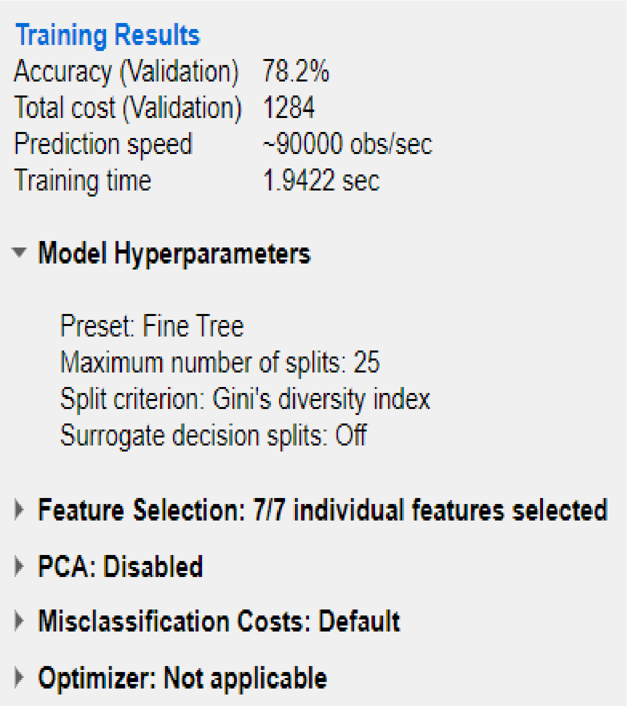
▸ **PCA: Disabled**

▸ **Misclassification Costs: Default**

▸ **Optimizer: Not applicable**

Figure 5.14 Training results of NN without PCA

- As it is mentioned in Chapter 4.2, although there isn't a rule, the most common choices for cross validation fold number are 5 or 10. Hence the size of the data set was not large this value was chosen as 5. The models were trained by selecting fold number as 10 to see how the accuracy rates of the trained models changed. Figures 5.15, 5.16, and 5.17 show the training results of 25 splits decision tree trained model, fine gaussian support vector machine trained model, and neural network trained model, respectively. Accuracy rates of these new trained models are 78.2%, 78.6%, and 77.2%. It was seen that the accuracy rates of all new training models whose fold numbers were 10 decreased.



Training Results

Accuracy (Validation) 78.2%

Total cost (Validation) 1284

Prediction speed ~90000 obs/sec

Training time 1.9422 sec

▾ Model Hyperparameters

Preset: Fine Tree

Maximum number of splits: 25

Split criterion: Gini's diversity index

Surrogate decision splits: Off

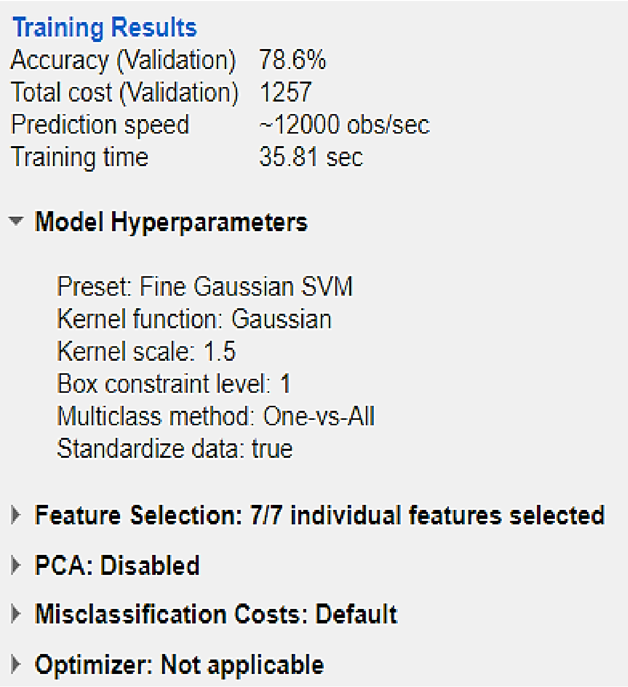
▸ Feature Selection: 7/7 individual features selected

▸ PCA: Disabled

▸ Misclassification Costs: Default

▸ Optimizer: Not applicable

Figure 5.15 Training results of DT trained model (25 splits) with 10-fold number



Training Results

Accuracy (Validation)	78.6%
Total cost (Validation)	1257
Prediction speed	~12000 obs/sec
Training time	35.81 sec

▼ **Model Hyperparameters**

Preset: Fine Gaussian SVM
Kernel function: Gaussian
Kernel scale: 1.5
Box constraint level: 1
Multiclass method: One-vs-All
Standardize data: true

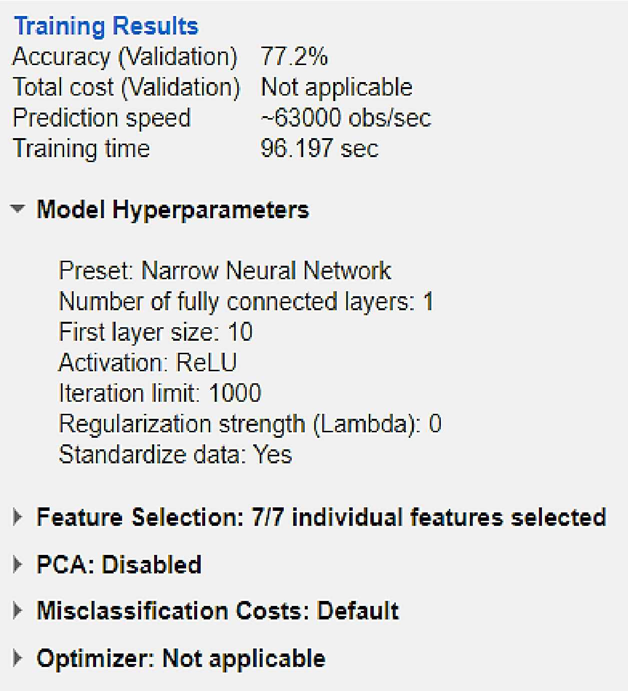
▶ **Feature Selection: 7/7 individual features selected**

▶ **PCA: Disabled**

▶ **Misclassification Costs: Default**

▶ **Optimizer: Not applicable**

Figure 5.16 Training results of fine gaussian SVM trained model with 10-fold number



Training Results

Accuracy (Validation)	77.2%
Total cost (Validation)	Not applicable
Prediction speed	~63000 obs/sec
Training time	96.197 sec

▼ **Model Hyperparameters**

Preset: Narrow Neural Network
Number of fully connected layers: 1
First layer size: 10
Activation: ReLU
Iteration limit: 1000
Regularization strength (Lambda): 0
Standardize data: Yes

▶ **Feature Selection: 7/7 individual features selected**

▶ **PCA: Disabled**

▶ **Misclassification Costs: Default**

▶ **Optimizer: Not applicable**

Figure 5.17 Training results of NN trained model with 10-fold number

CHAPTER 6

CONCLUSIONS AND RECOMMENDADITONS

Within the scope of this study, accident prediction models were created with decision trees, support vector machines, and neural networks, which are machine learning algorithms, by using the accident data of Turkish Hard Coal Enterprise Amasra, Armutçuk, Karadon, Kozlu, Üzülmez district.

The main conclusions drawn from this study can be listed as:

i. The used data covered the years 2008-2010 with variables such as shift, day, job, education, type of accident, reason of the accident, location of the accident, severity of the accident, age, seniority, affected body part. When the basic statistics of the data set were examined, primary school graduates (54.26% of the 8406 accidents) and production workers (80.25%), with experience ranging from 0 to 7 years (72.22%) and ages ranging from 27 to 30 years (35.82%), had the highest rate of encountering an accident in the production area (52.58%) in the first shift (08:00-16:00, 53.81%) on Mondays (18.34%) due to falling rock (88.52%) and personal mistakes (46.04%), and the most injured body parts were the hand fingers (29.26%) as a result of these accidents.

ii. The principal component analysis concluded that seven variables, which are seniority, age, job, type of accident, reason of the accident, location of the accident, and shift, were sufficient to be used in subsequent analyses. According to the result of the principal component analysis, the seniority, age, and job variables better explain the variance of the data on the factorial plane than other variables. This means that seniority, age, and job are important factors in work-related accident data.

iii. The accuracy of the trained DT, SVM, and NN models was increased by changing the hyperparameters of the algorithms.

iv. It was concluded that the trained prediction model that gave the highest accuracy rate (78.5%) in the decision tree algorithm was the 25-splits decision tree model 4.

v. In the algorithm of support vector machines, it was found that the trained prediction model that gave the highest accuracy rate (79.2%) was the fine gaussian model 7 with the hyperparameters, which were 1.5 kernel scale, 1 box constraint level, and one-vs-all multiclass methods.

vi. It was seen that the highest accuracy rate (78.5%) in the neural network algorithm was found with the default hyperparameters of narrow NN model, which were 1 fully connected layer, 10 outputs in the layer (layer size), and ReLU activation function.

vii. The accuracy results of three best-trained prediction machine learning algorithm models are 78.5% (DT), 79.2% (SVM), and 78.5% (NN). The best trained prediction model is determined as fine gaussian model 7, which is a support vector machine method with hyperparameters of 1.5 kernel scale, 1 box constraint level, one-vs-all multiclass, and it has the highest accuracy score with a value of 79.2%.

viii. According to the results obtained in the study, decision trees and neural networks, also showed close success with the algorithm for support vector machines.

ix. The dominant truly classified severity type for the three best trained prediction models is the slightly injured. The number of data correctly classified as slightly injured from 4611 slightly injured observations was 4544, 4521, and 4394 for decision tree, support vector machine, and neural network trained prediction models, respectively (Figure 5.1, 5.5, 4.16).

x. The test data, which comprised 30% of the total data set, were used as input to validate the trained prediction model. The accuracy of the test was 78.3%, and the number of correct classifications was 1975 out of 2521 observations.

The dominant correct classification severity type was slightly injured as a result of this test result. In addition, the test results revealed that the gallery was the location with the most accurate classification of accident severity, with an accuracy rate of 89.82 percent.

xi. The trained SVM (fine gaussian model 7) model was found to have the lowest observed minimum classification error (0.20798) as well as the highest accuracy rate among other trained models. However, the trained NN model has a smaller observed minimum classification error (0.21490) than the trained DT model (0.21495), although their accuracy rates were equal.

xii. It was seen that when the prediction models were built after applying PCA and reducing the number of variables, the accuracy rate increased and so the error rate decreased.

xiii. It was stated that when the cross-validation fold number was selected as 10, the accuracy of the trained models decreased. As an expected result, the error rates increased. The error rates of new trained DT, SVM, and NN models are 21.8%, 21.4%, and 22.8%, respectively. These error rates were 21.5%, 20.8%, and 21.5% for trained DT, SVM, and NN models with 5-fold number. It was found that the maximum error rate increased in the NN trained model.

Some recommendations for future studies can be listed as:

- It is obvious that it is very important to accurately and consistently record work-related accidents in order to get better results.
- Since each employee will have different working characteristics such as age, job, shift, working location, special occupational health training programs can be organized and occupational health and safety measures can be taken by using the results obtained from the prediction model.
- With the prediction model proposed as a result of this study, the possible severity of the accidents in the workplaces can be determined by determining the dangerous situations determined as a result of the audits to be carried out in the working places in coal mining. Depending on these results, the risk

levels of these possible accidents can be determined for risk analysis. For example, the Fine Kinney risk analysis method is an easy-to-use and widely used method. The severity of the damage or damage that the hazard will cause to people, the environment, and the workplace is one of the Fine Kinney risk analysis calculation parameters (Kinney and Wiruth, 1976). It was concluded that this prediction model, which was created by using the data of the workplace, is a method that can be used to determine this parameter. Thus, subjective judgment, which is generally included in the perception of risk, will be replaced by objective criteria. This is important in terms of preventing risks that will cause serious accidents.

- Model results should be controlled and compared after data from 2010 onwards has been obtained. For this reason, in the studies to be carried out with data sets belonging to later years, the comparison of the prediction results with the support vector machine prediction results and also by applying other advanced data mining methods will be research that will contribute to the future period.

- The trained prediction model predicts the severity of accidents that only have known inputs. However, there might be new inputs that there is no information about in the current data set. Thus, continuous training is very important for the prediction models to make adaptable, scalable, and accurate predictions.

REFERENCES

Akbari, M., Asadi, P., & Givi, M. (2014). Artificial neural network and optimization. *Advances in Friction-Stir Welding and Processing,* 543-599.

Alaeddinoğlu, M. F., Sincar, S., & Naralan, A. (2015). A decision support system developed for risk analysis and assessment in occupational health and safety (artificial neural network). *Journal of Cukurova University Faculty of Engineering and Architecture*, *30*(2), 275-291.

Alpar, R. (2003). Introduction to Applied Multivariate Statistical Methods 1. *Nobel Publication Distribution/Technical Series,* 410.

Alpaydın, E. (2010). Introduction to Machine Learning. *The MIT Press, 1*, 115.

Arslanhan, S., & Cünedioğlu, H.E. (2010). An assessment on work accidents in mines and their consequences. *Economic Policy Research Foundation of Turkey*.

Artificial Intelligence. (2020). Retrieved from https://www.ibm.com/cloud/learn/neural-networks

Atalay, V. (2019). *Chapter six: Dimensionality Reduction* [powerpoint slides]. Unpublished manuscript, CEng574, Middle East Technical University, Ankara, Turkiye.

Berwick, R. (2003). *An Idiot's guide to Support vector machines (SVMs)*. Retrieved from https://web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf.

BP (2021). *Statistical Review of World Energy, 70*, 48.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. *Boca Raton, Fla.: CRC Pres*.

Cattell, R. B. (1966). The scree plot test for the number of factors. *Multivariate Behavioral Research, 1*, 140-161.

Ceriani, L., & Verme, P. (2012). The origins of the gini index: extracts from variabilità e mutabilità (1912). *The Journal of Economic Inequality, 10*(3), 421-443.

Chen, J., ZHU, D., CHEN, Y., YE, A., & QIU, W. (2015). Analysis of the stability for mine tailings dam based on pca-bp neural network. *Gold Science and Technology*, (5), 47-52.

Coal. (2022). Republic of Turkey Ministry of Energy and Natural Resources. Retrieved 15 July 2022, from https://enerji.gov.tr/english-info-bank-natural-resources-coal

Cornero, M.C., & Pedregal, D.J. (2010). Modelling and forecasting occupational accidents of different severity levels in Spain. *Reliability Engineering and System Safety*, 95, 1134-1141.

Gomes, H.M., & Awruch, A.M. (2004). Comparison of response surface and neural network with other methods for structural reliability analysis, 2649–2667.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation* (2nd ed.). Prentice Hall.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, *24*, 417-441 and 498-520.

ILO. (2016). *Support Kit for Developing Occupational Safety and Health Legislation*. Retrieved from https://www.ilo.org/wcmsp5/groups/public/---ed_dialogue/---lab_admin/documents/publication/wcms_834142.pdf

Jollife, I.T. (2002). *Principal Component Analysis* (2nd ed.). New York: Springer-Verlag.

Kaiser, H. F. (1960). The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, *20*, 141-151.

Kinney, G.F., & Wiruth, A.D. (1976). Practical risk analysis for safety management, *NWC Technical publication 5865*, Naval Weapons Center, China Lake CA, USA.

Konak, E.S. (2006). *Computer aided facial recognition system design* [Master dissertation, Istanbul University]. Institute of Science and Technology, p58.

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling* (1st ed.). Springer.

Lindgren, I. (2020). *Citation: Dealing with Highly Dimensional Data using Principal Component Analysis (PCA)*. Retrieved from https://towardsdatascience.com/dealing-with-highly-dimensional-data-using-principal-component-analysis-pca-fea1ca817fe6

Lööw, J., & Nygren, M. (2019). Initiatives for increased safety in the Swedish mining industry: studying 30 years of improved accident rates. *Saf. Sci.,* 437-446. https://doi.org/10.1016/j.ssci.2019.04.043

Meng, X., Lu, P., & Wang, B. (2017). Coal mine safety warning system based on principal component method and neural network, *6th Data Driven Control and Learning Systems (DDCLS)*.

Nalborczyk, L. (2021). *A gentle introduction to deep learning in R using Keras*. Retrieved from https://www.barelysignificant.com/slides/vendredi_quanti_2021/vendredi_quantis#1

Nenonen, N. (2013). Analysing factors related to slipping, strumbling, and falling accidents at work: Application of data mining methods to Finnish occupational accidents and diseases statistics database. *Applied Ergonomics*, *44*, 215-224.

Ning, Y., Chen, X. (2009). Coal Face Gas Emission Prediction Based on Support Vector Machine. *2009 International Conference on Artificial Intelligence and Computational Intelligence*. Retrieved from https://ieeexplore.ieee.org/document/5375985

Nitze, I., Schulthess, U., & Asche, H. (2012). Comparison of machine learning algorithms random forest, artficial neural network and support vector machine to maximum likelihood for supervised crop type classification. *Proceedings of the 4th GEOBIA,* 35-40.

Oztemel, E. (2003). Artificial neural networks. *Istanbul Papatya Publishing*, *1,* 50.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, *2*(11), 559-572.

Peng, H., Li, L., & Li, Y. (2010). Model of gas warning system based on neural network expert. *International Conference on Computer Design and Applications*.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1,* 81-106.

Ruilin Z., & Lowndes I. (2010). The application of a coupled artificial neural network and fault tree analysis model to predict coal and gas outbursts. *International Journal of Coal Geology*, *84*, 141–152.

Russell, S. J., & Norvig, P. (2003). Artificial intelligence: a modern approach. *N.J.: Prentice Hall*.

Sánchez, S. (2011). Prediction of work-related accidents according to working conditions using support vector machines. *Applied Mathematics and Computation*, *218*(7), 3539-3552.

Sanmiquel, L., Rossell, J.M., & Vintro, C. (2015). Study of Spanish mining accidents using data mining techniques. *Safety Science*, *75,* 49-55.

Snoek, J., Larochelle, H., & Adams, R. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems ,25,* 2960-2968.

SSI. (2020). *Statistical annuals*. Retrieved from http://www.sgk.gov.tr/wps/portal/sgk/tr/ kurumsal/ istatistik/ sgk_istatistik_yilliklari

SSI. (2020). *Informal Employment Rate*. Retrieved from http://www.sgk.gov.tr/wps/portal/sgk/tr/calisan/kayitdisi_istihdam/kayitdisi_istihdam_oranlari

Suratgar, A., Tavakoli, M., & Hoseinabadi, A. (2005). Modified Levenberg-Marquardt method for neural networks training, *World Acad Sci Eng Technol, 1,* 1719–1721.

Thakur A., Sarkar D., Bharti V., & Kannan U. (2021). Development of in-core fuel management tool for AHWR using artificial neural networks. *Annals of Nuclear Energy, 150,* 107869. https://doi.org/10.1016/j.anucene.2020.107869

Turkish Hard Coal Enterprises. (2022). *History*. Retrieved from http://www.taskomuru.gov.tr/ (Last accessed on 18.06.2022).

Turkish Hard Coal Enterprises. (2022). *About us*. Retrieved from http://taskomuru.net/tr/hakkimizda/ (Last accessed on 18.06.2022).

Turkish Hard Coal Enterprises. (2022). *The License Areas*. Retrieved from http://taskomuru.net/tr/imtiyaz-alanlari/ (Last accessed on 18.06.2022).

Turkish Hard Coal Enterprises. (2022). *2021 Hard Coal Sector Report*. Retrieved from http://taskomuru.net/tr/whiseezu/2022/05/2021yilisektor.pdf

Von Bartheld, CS., Bahney, J., & Herculano-Houzel, S. (2016). The search for true numbers of neurons and glial cells in the human brain: A review of 150 years of cell counting. *J Comp Neurol, 524*(18), 3865–3895.

Wu, R. M., Zhang, Z., Yan, W., Fan, J., Gou, J., Liu, B., Gide, E., Soar, J., Shen, B., Fazal-e-Hasan, S., Liu, Z., Zhang, P., Wang, P., Cui, X., Peng, Z., & Wang, Y. (2022). A comparative analysis of the principal component analysis and entropy weight methods to establish the indexing measurement. *Plos One, 17*(1). https://doi.org/10.1371/journal.pone.0262261

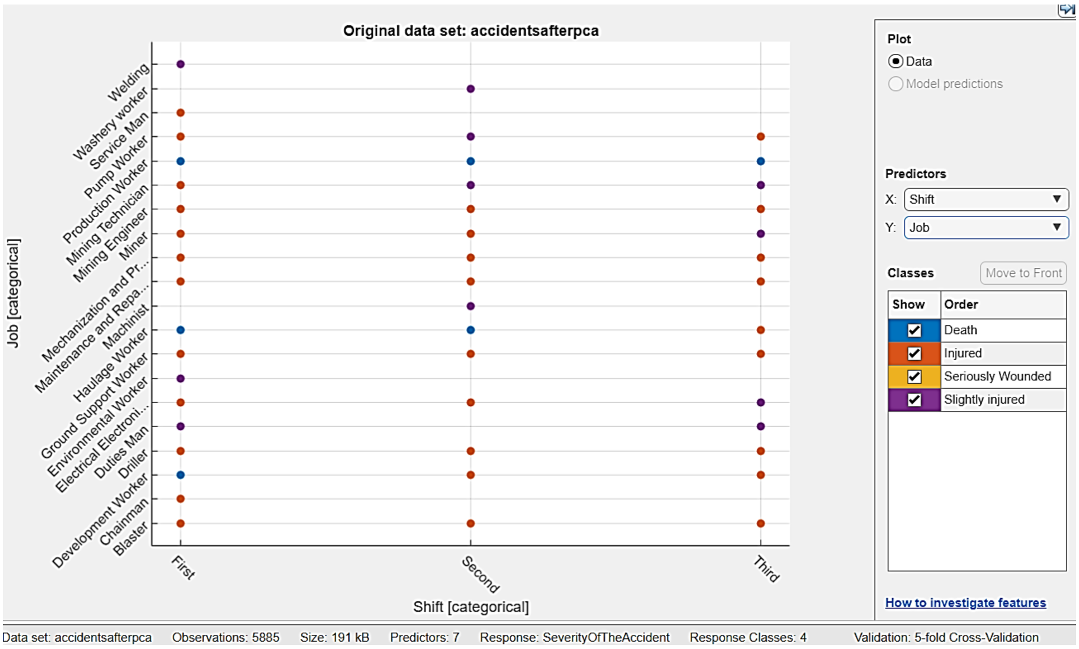
Zhang, Y., Tang, S., & Shi, K. (2022). Risk assessment of coal mine water inrush based on PCA-DBN. *Scientific Reports, 12*(1). https://doi.org/10.1038/s41598-022-05473-8

APPENDICES

A. Used Codes for Principal Component Analysis

```
library(readxl)

accidents<-read_excel("C:/Users/umpg0473/Desktop/accidents.xlsx")

str(accidents)

a<- factor(accidents$Shift)

 shift <- as.numeric(a)

 b<- factor(accidents$Day)

 day<- as.numeric(b)

 c<- factor(accidents$Job)

 job<- as.numeric(c)

 d<- factor(accidents$Education)

 education<- as.numeric(d)

 e<- factor(accidents$`Type of Accident`)

 typeofaccident<- as.numeric(e)

 f<- factor(accidents$`Reason of The Accident`)

 reasonofaccident<- as.numeric(f)

 g<- factor(accidents$`Location of The Accident`)

 location<- as.numeric(g)

 h<- factor(accidents$`Severity of The Accident`)

 severityofaccident<- as.numeric(h)
 i<- factor(accidents$Age)

 age<- as.numeric(as.character(i))

 j<- factor(accidents$`Seniorty (year)`)

 seniorty<- as.numeric(as.character(j))

 k<- factor(accidents$`Body Part`)

 bodypart<- as.numeric(k)

mydata<- cbind(shift,day,job,education,typeofaccident,reasonofaccident,locationofaccident,severityofaccident,age,seniorty,bodypart)

data_omit <- na.omit(mydata)

prin_comp <- prcomp(data_omit, scale = TRUE)

names(prin_comp)

CentData <-prin_comp$center

prin_comp$scale

prin_comp$rotation

dim(prin_comp$x)

std_dev <- prin_comp$sdev

std_dev

pr_var <- std_dev^2

pr_var

plot(prin_comp)

fviz_eig(prin_comp, addlabels = TRUE, ylim = c(0, 50))

fviz_pca_var(prin_comp, col.var = "black")
fviz_pca_var(prin_comp, col.var="cos2", gradient.cols = c("#00AFBB",
"#E7B800", "#FC4E07"), repel = TRUE # Avoid text overlapping)

library(ggfortify)

autoplot(prin_comp, data = accidents, colour = 'severityofaccident', loadings =
TRUE)

autoplot(prin_comp, data = accidents, colour = 'severityofaccident',loadings =
TRUE, loadings.colour = 'blue',loadings.label = TRUE, loadings.label.size = 3)
```

B. Scatter Plots with Respect to Predictors



Plot

● Data

○ Model predictions

Predictors

X: Shift

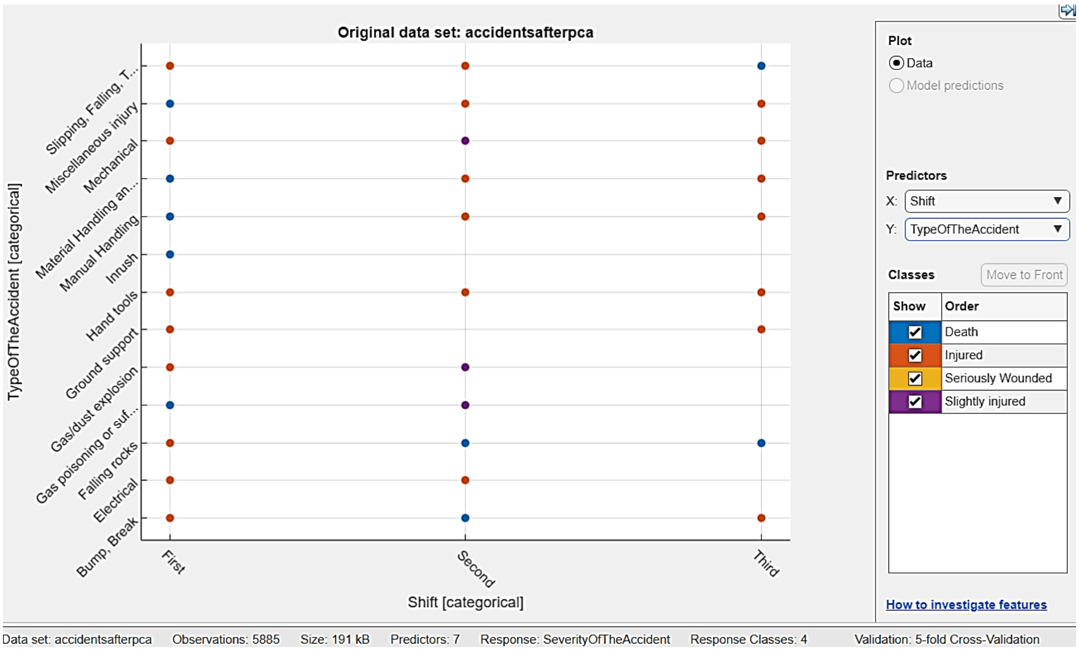
Y: Job

Classes Move to Front

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 1 Scatter Plot (job versus shift)



Plot

● Data

○ Model predictions

Predictors

X: Shift

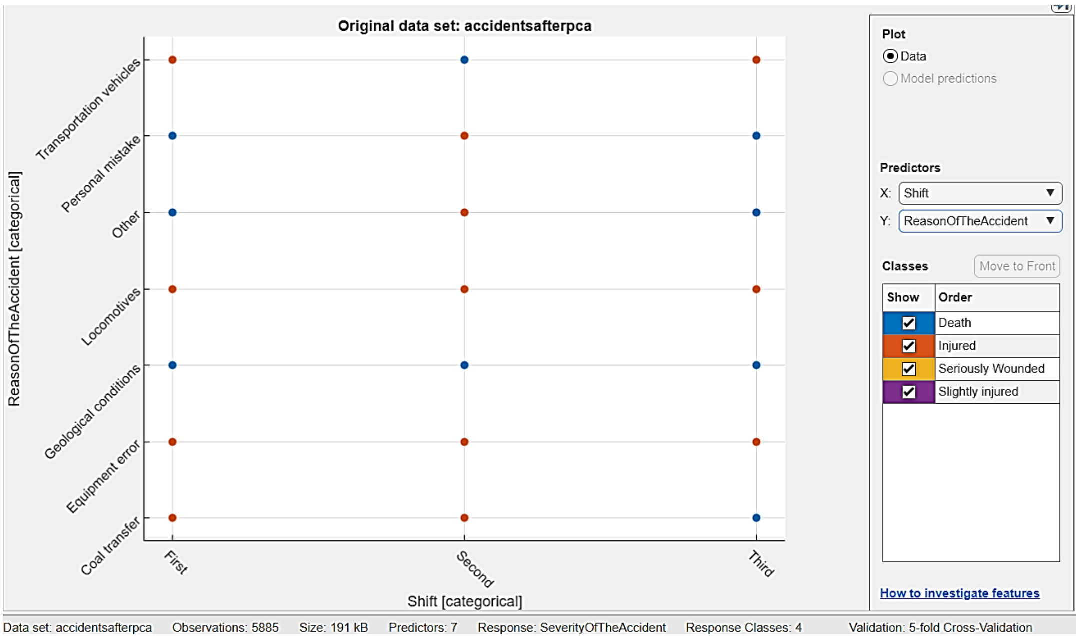
Y: TypeOfTheAccident

Classes Move to Front

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 2 Scatter Plot (type of accidents versus shift)



Plot
● Data
○ Model predictions

Predictors
X: Shift
Y: ReasonOfTheAccident

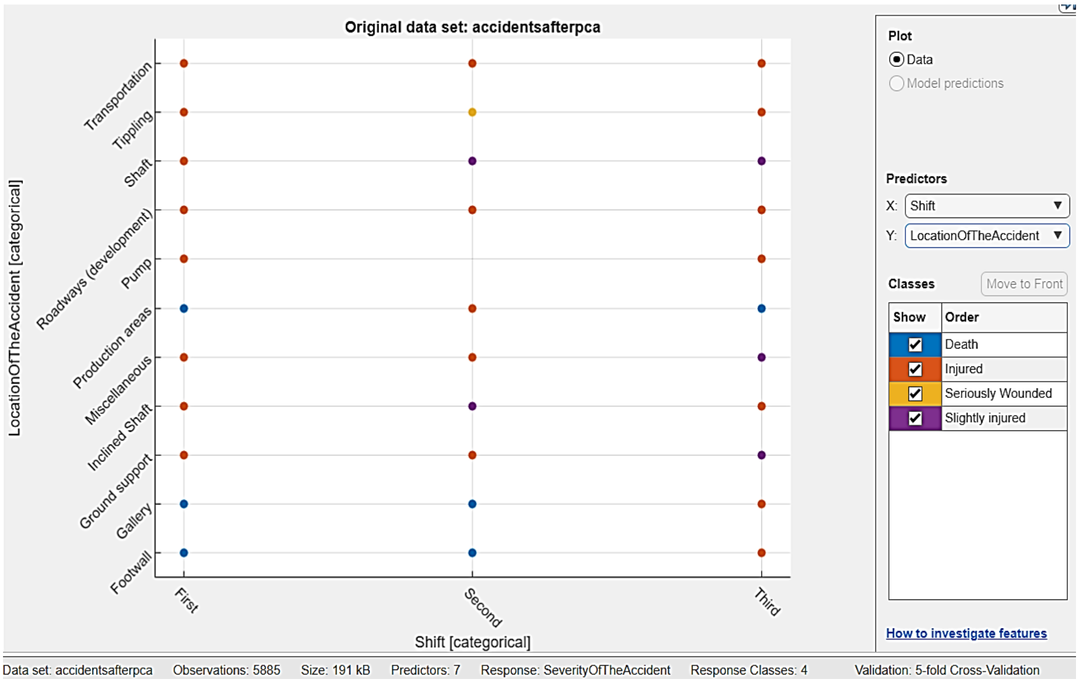
Classes Move to Front

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

How to investigate features

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 3 Scatter Plot (reason of accidents versus shift)



Plot
● Data
○ Model predictions

Predictors
X: Shift
Y: LocationOfTheAccident

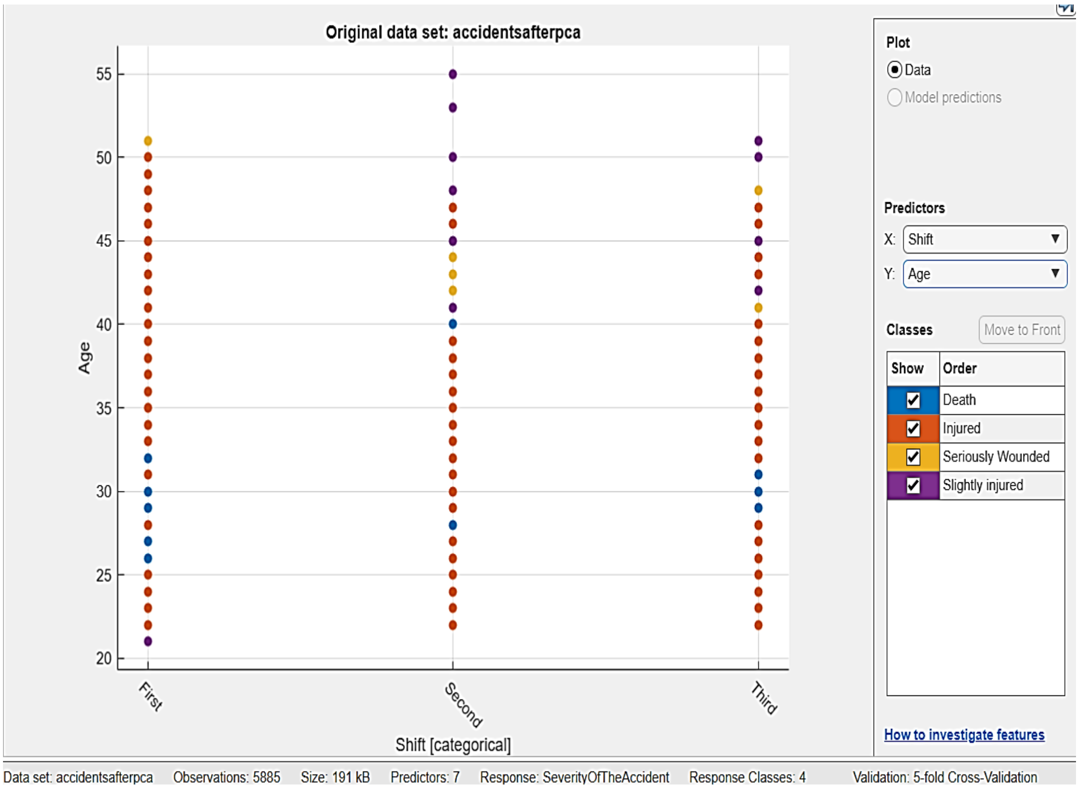
Classes Move to Front

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

How to investigate features

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 4 Scatter Plot (location of accidents versus shift)



Plot

● Data

○ Model predictions

Predictors

X: Shift

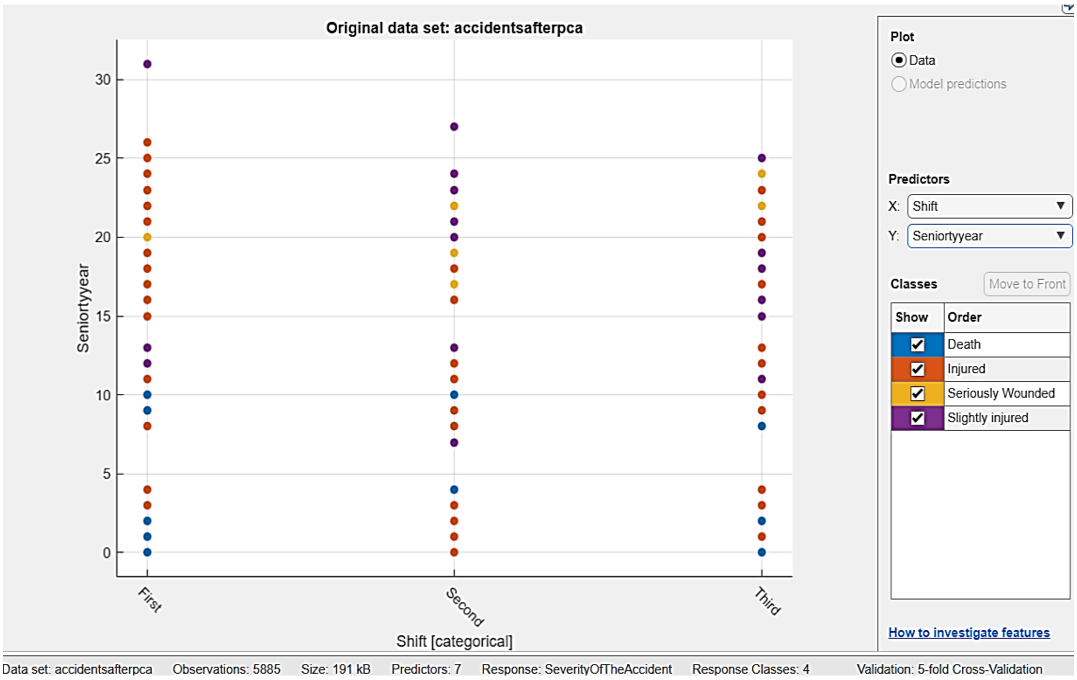
Y: Age

Classes

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 5 Scatter Plot (age versus shift)



Plot

● Data

○ Model predictions

Predictors

X: Shift

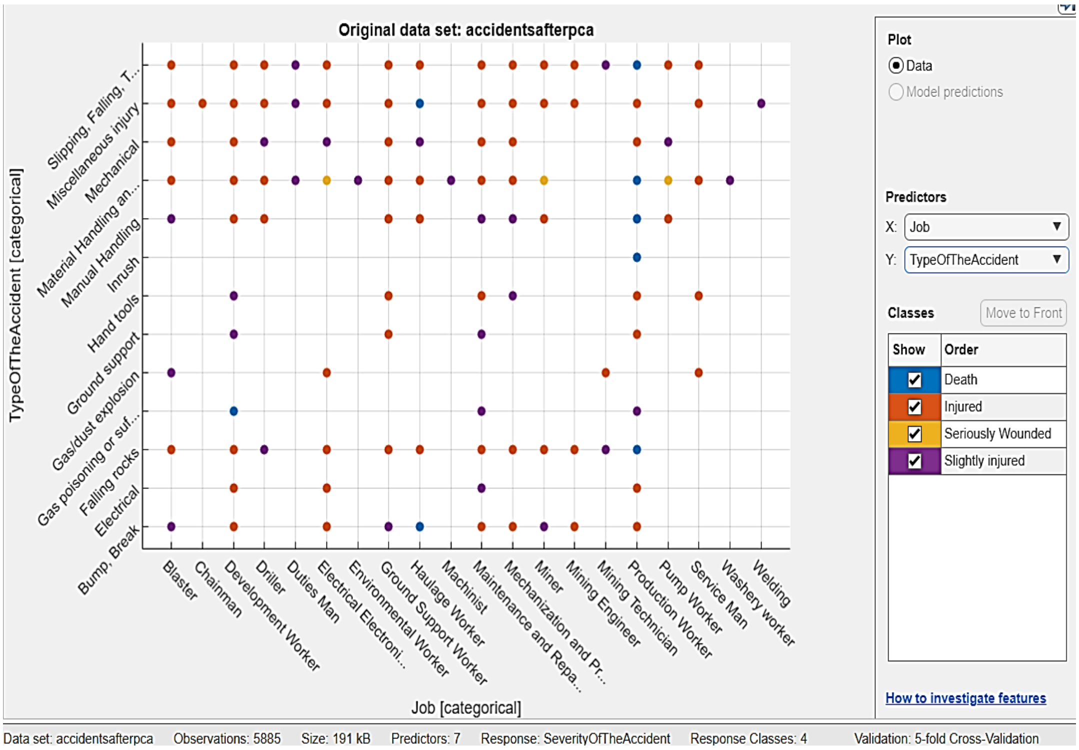
Y: Seniortyyear

Classes

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 6 Scatter Plot (seniority versus shift)



Plot
● Data
○ Model predictions

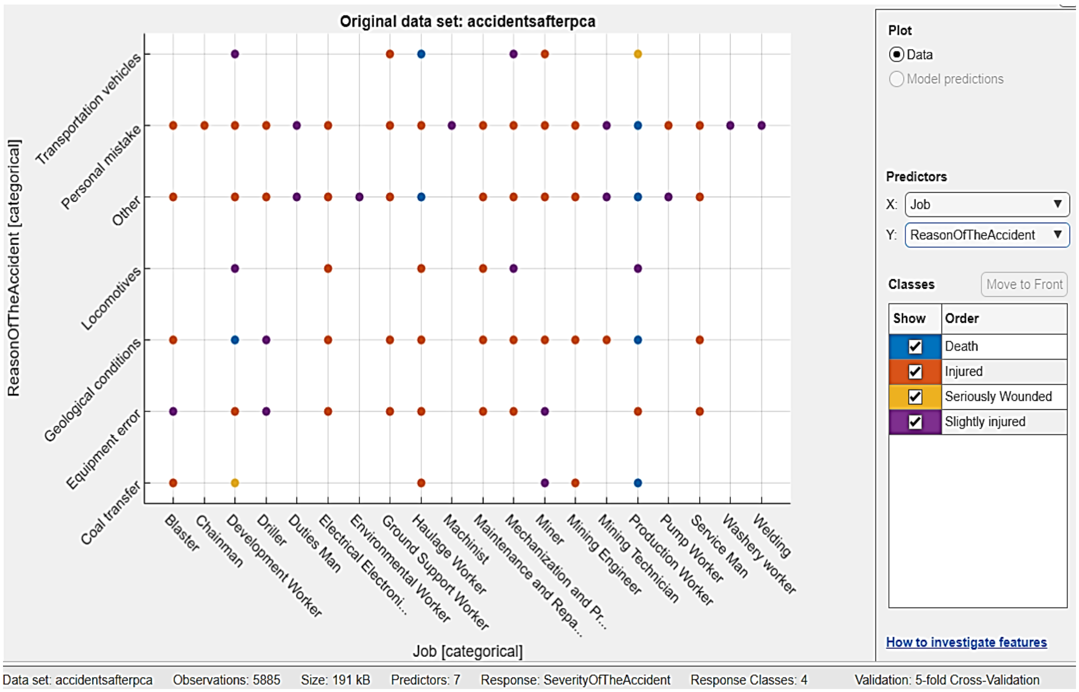
Predictors
X: Job
Y: TypeOfTheAccident

Classes

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 7 Scatter Plot (type of accidents versus job)



Plot
● Data
○ Model predictions

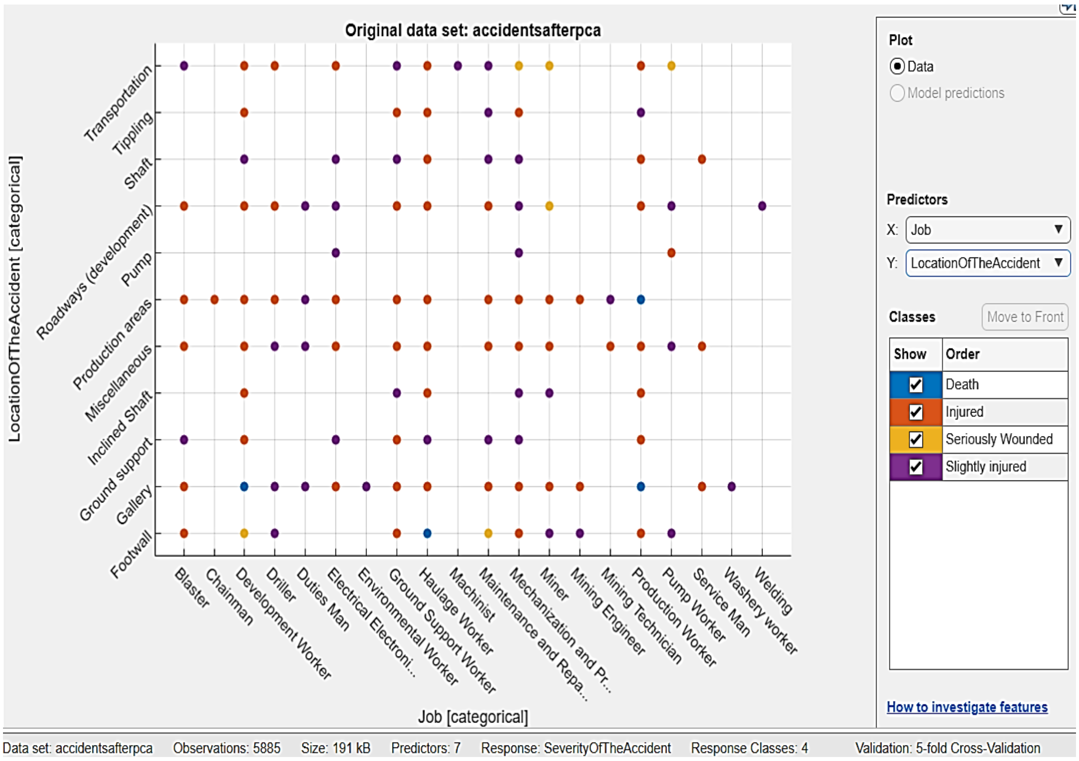
Predictors
X: Job
Y: ReasonOfTheAccident

Classes

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 8 Scatter Plot (reason of accidents versus job)



Plot

● Data
○ Model predictions

Predictors

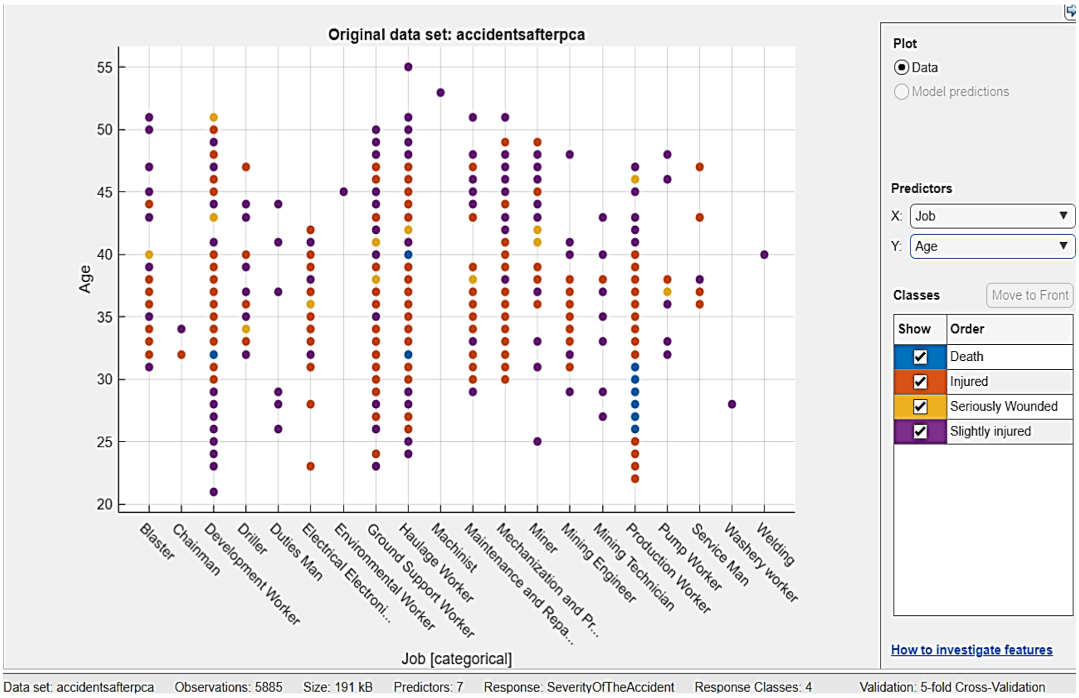
X: Job ▼
Y: LocationOfTheAccident ▼

Classes

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 9 Scatter Plot (location of accidents versus job)



Plot

● Data
○ Model predictions

Predictors

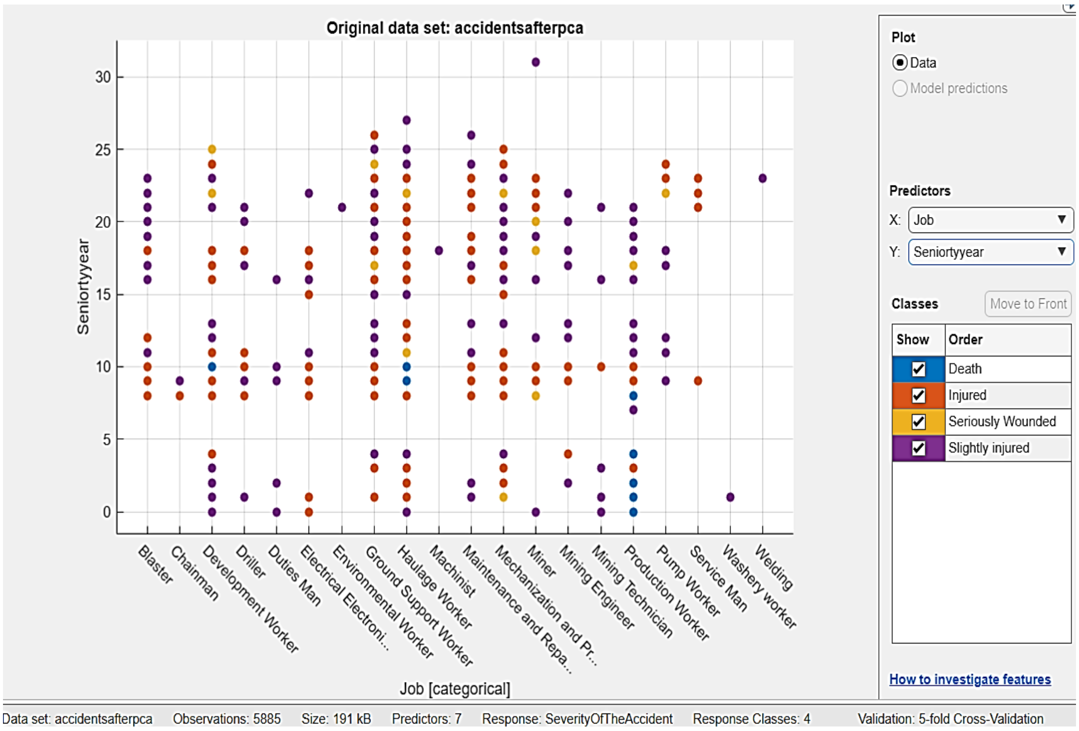
X: Job ▼
Y: Age ▼

Classes

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 10 Scatter Plot (age versus job)



Plot

● Data

○ Model predictions

Predictors

X: Job

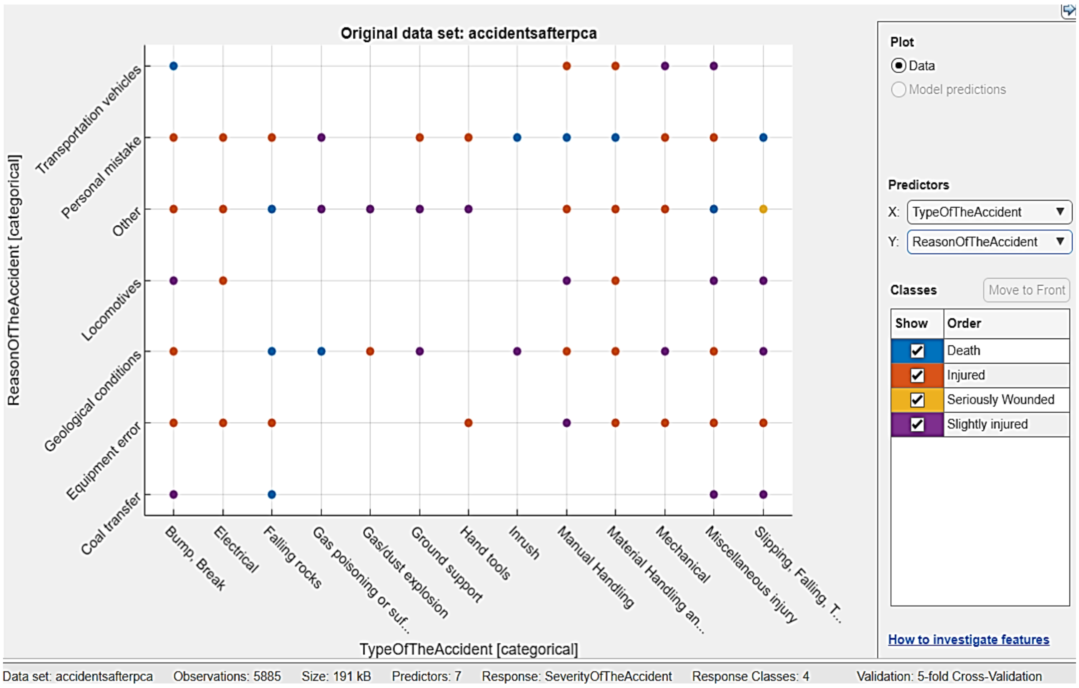
Y: Seniortyyear

Classes

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 11 Scatter Plot (seniority versus job)



Plot

● Data

○ Model predictions

Predictors

X: TypeOfTheAccident

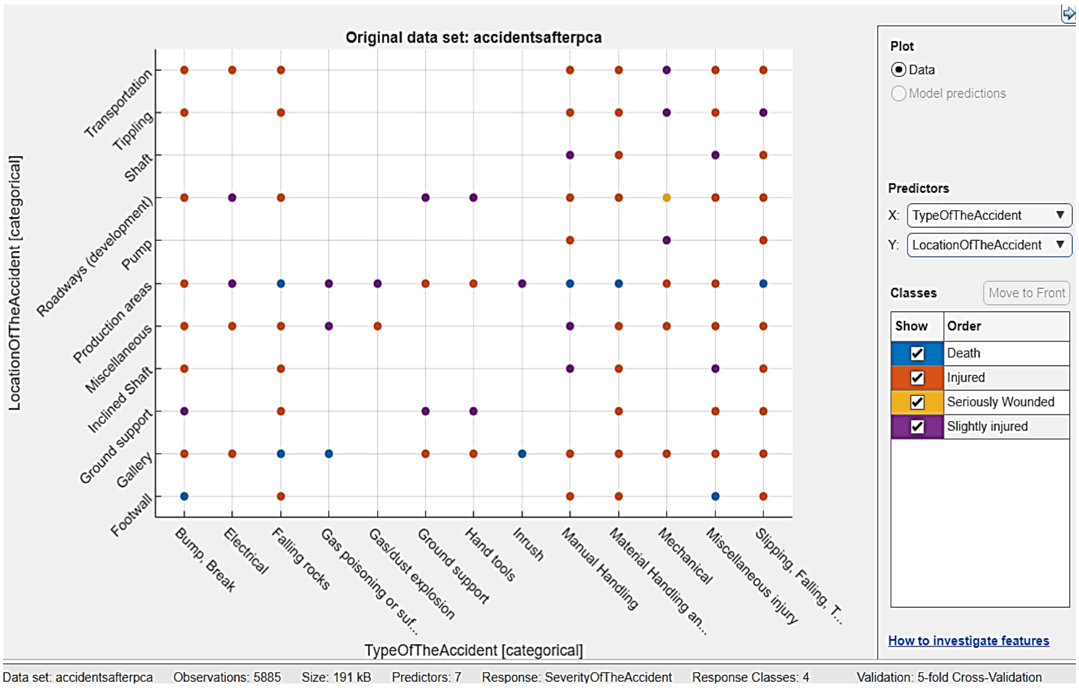
Y: ReasonOfTheAccident

Classes

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 12 Scatter Plot (reason of accidents versus type of accidents)



Plot

● Data
○ Model predictions

Predictors

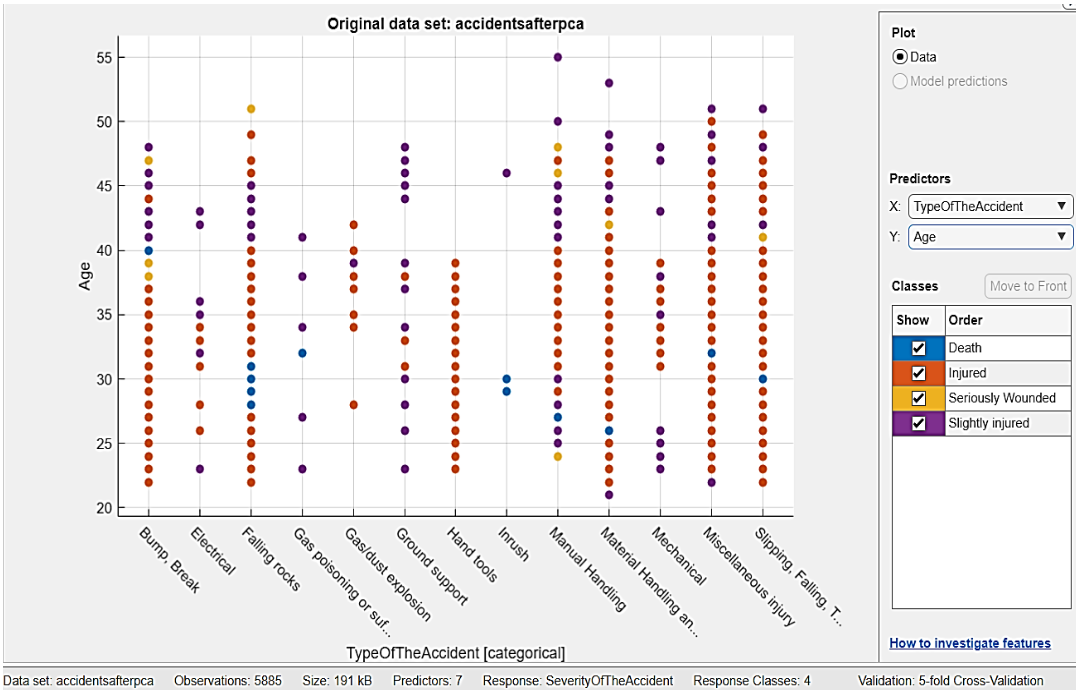
X: TypeOfTheAccident ▼
Y: LocationOfTheAccident ▼

Classes

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 13 Scatter Plot (location of accidents versus type of accidents)



Plot

● Data
○ Model predictions

Predictors

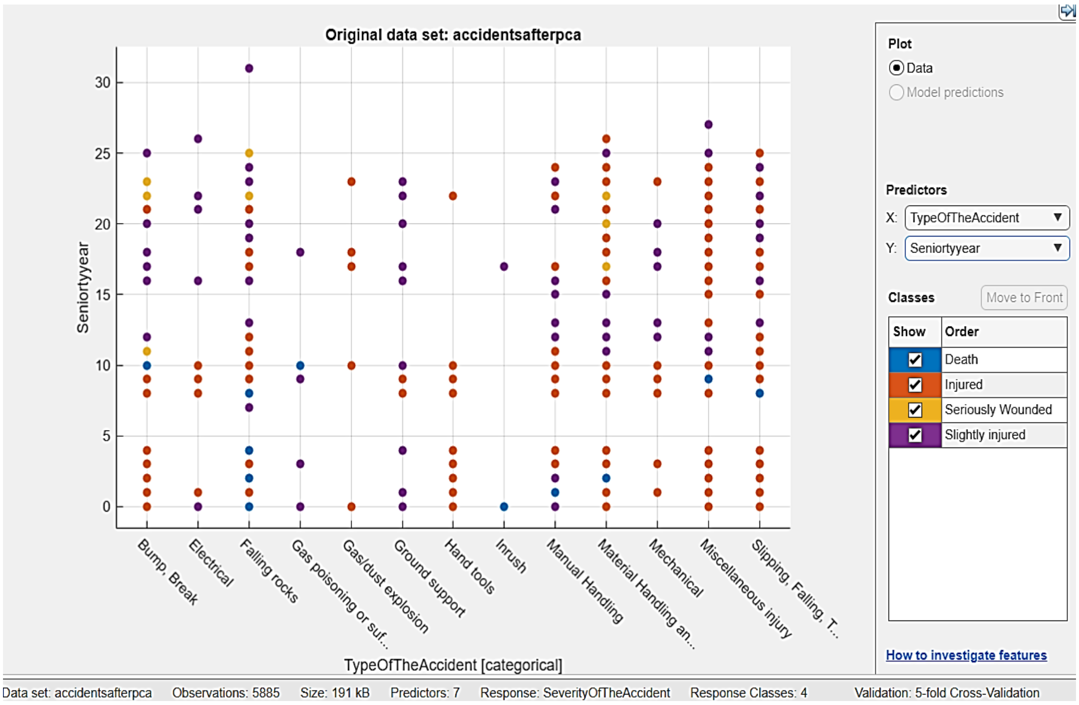
X: TypeOfTheAccident ▼
Y: Age ▼

Classes

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 14 Scatter Plot (age versus type of accidents)



Plot

● Data

○ Model predictions

Predictors

X: TypeOfTheAccident ▼

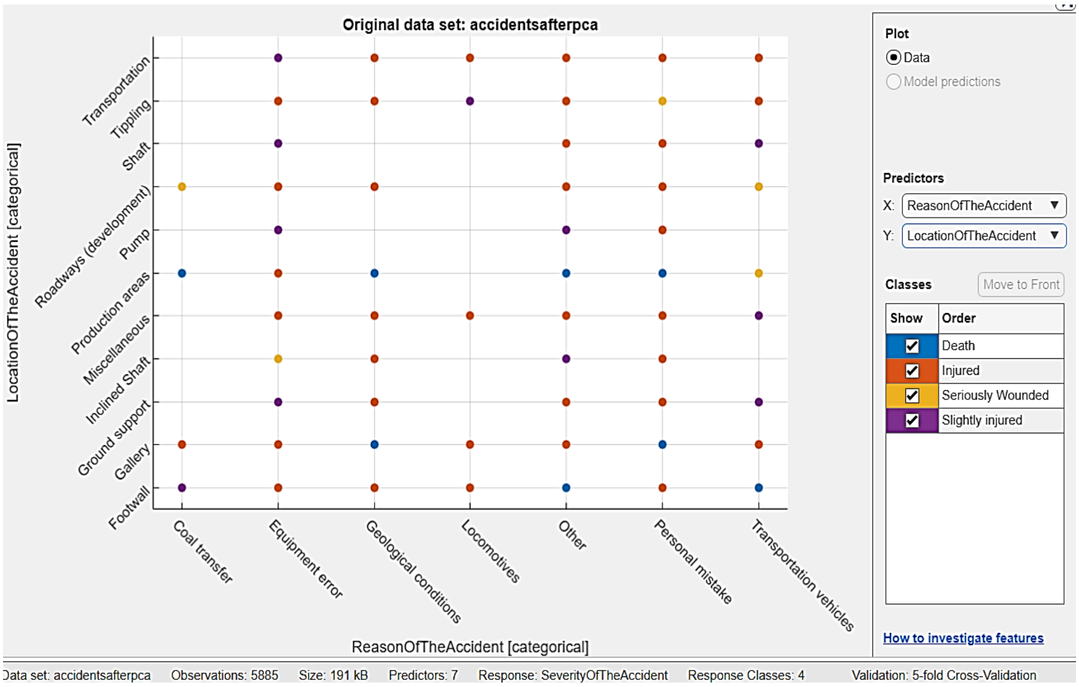
Y: Seniortyyear ▼

Classes

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 15 Scatter Plot (seniority versus type of accidents)



Plot

● Data

○ Model predictions

Predictors

X: ReasonOfTheAccident ▼

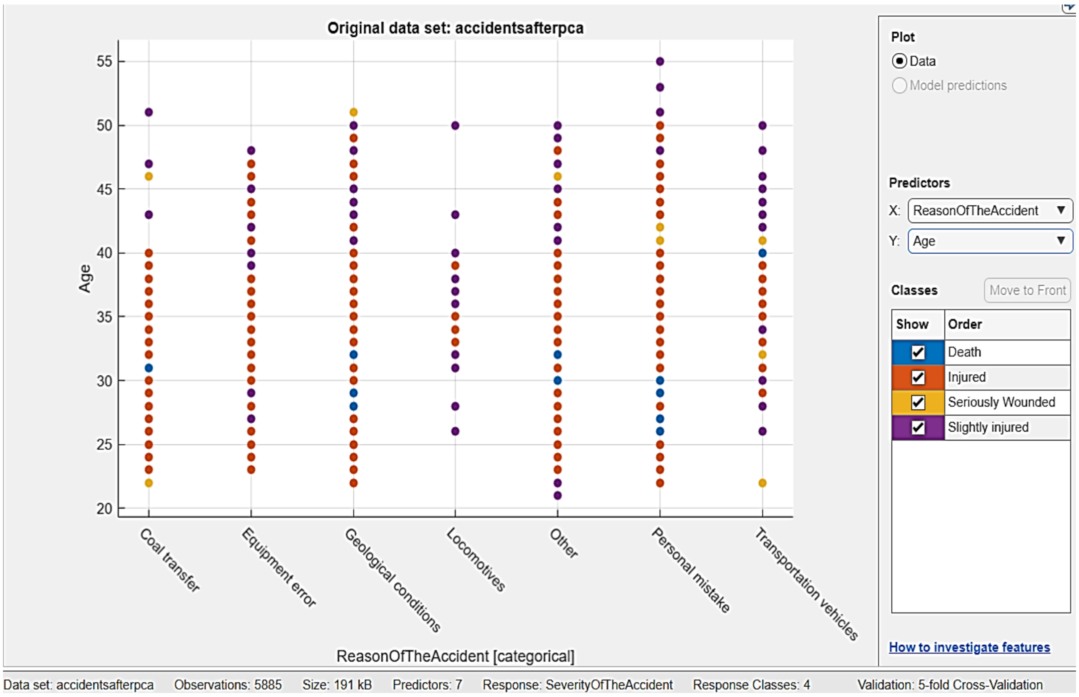
Y: LocationOfTheAccident ▼

Classes

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 16 Scatter Plot (location of accidents versus reason of accidents)



Plot

● Data

○ Model predictions

Predictors

X: ReasonOfTheAccident ▼

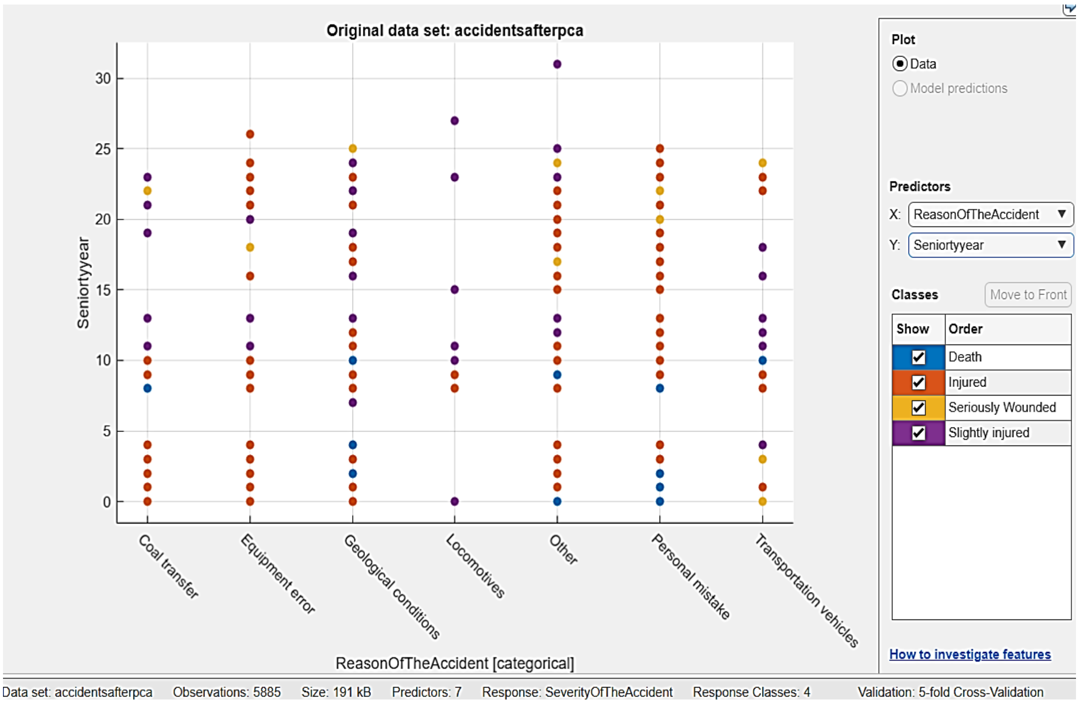
Y: Age ▼

Classes Move to Front

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 17 Scatter Plot (age versus reason of accidents)



Plot

● Data

○ Model predictions

Predictors

X: ReasonOfTheAccident ▼

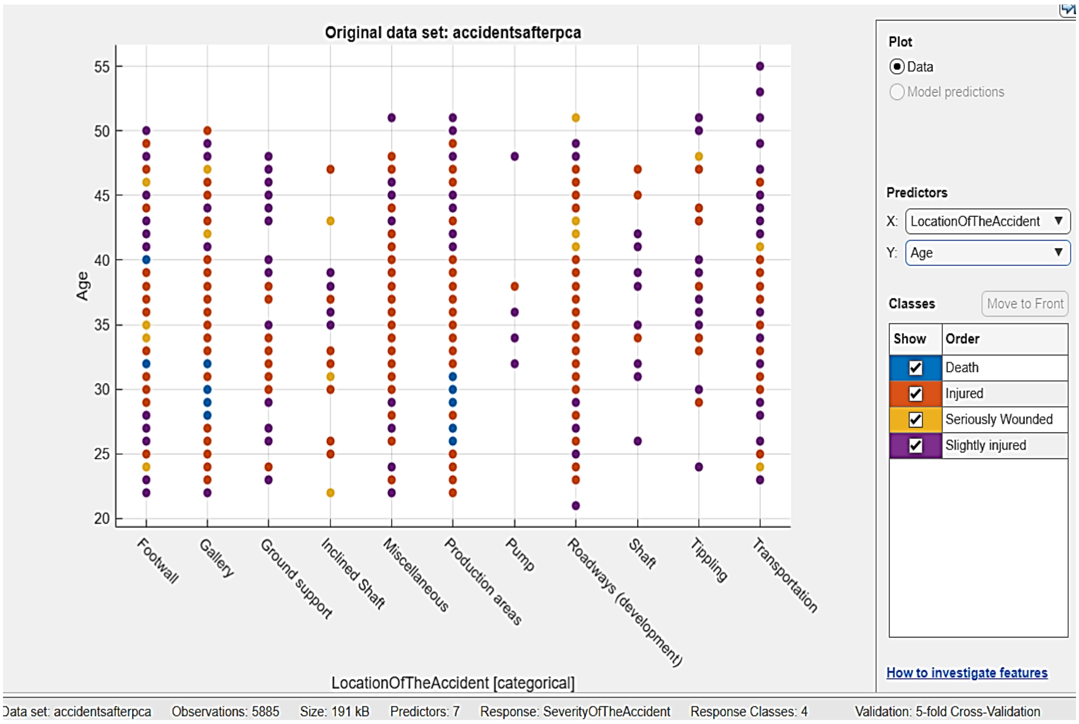
Y: Seniortyyear ▼

Classes Move to Front

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 18 Scatter Plot (seniority versus reason of accidents)



Plot

● Data

○ Model predictions

Predictors

X: LocationOfTheAccident ▼

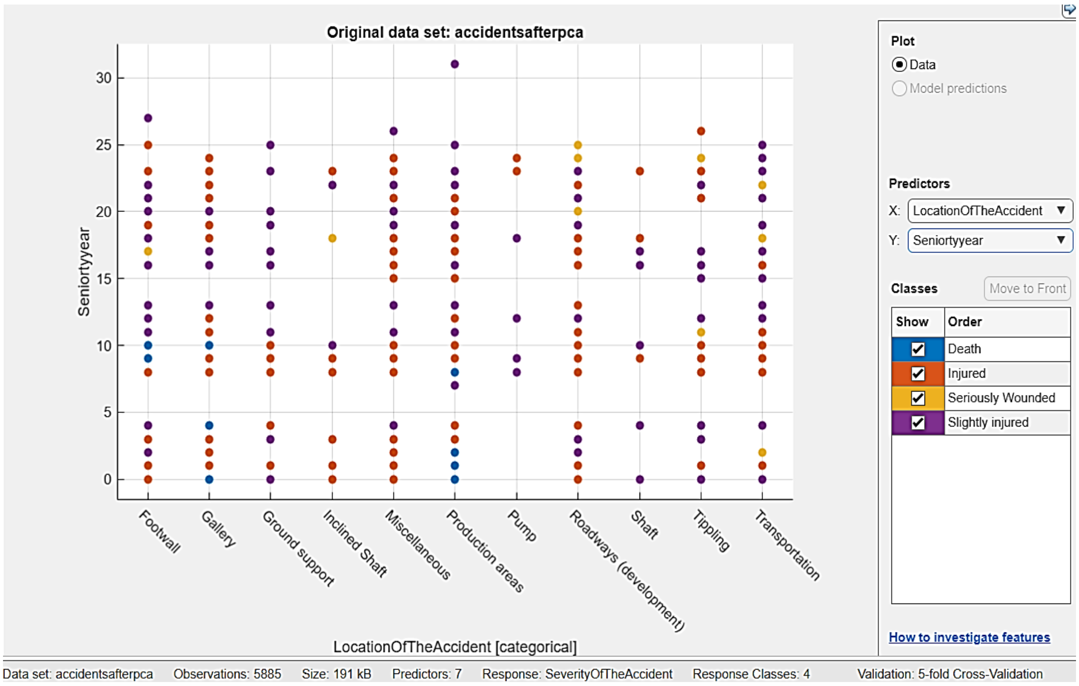
Y: Age ▼

Classes

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 19 Scatter Plot (age versus location of accidents)



Plot

● Data

○ Model predictions

Predictors

X: LocationOfTheAccident ▼

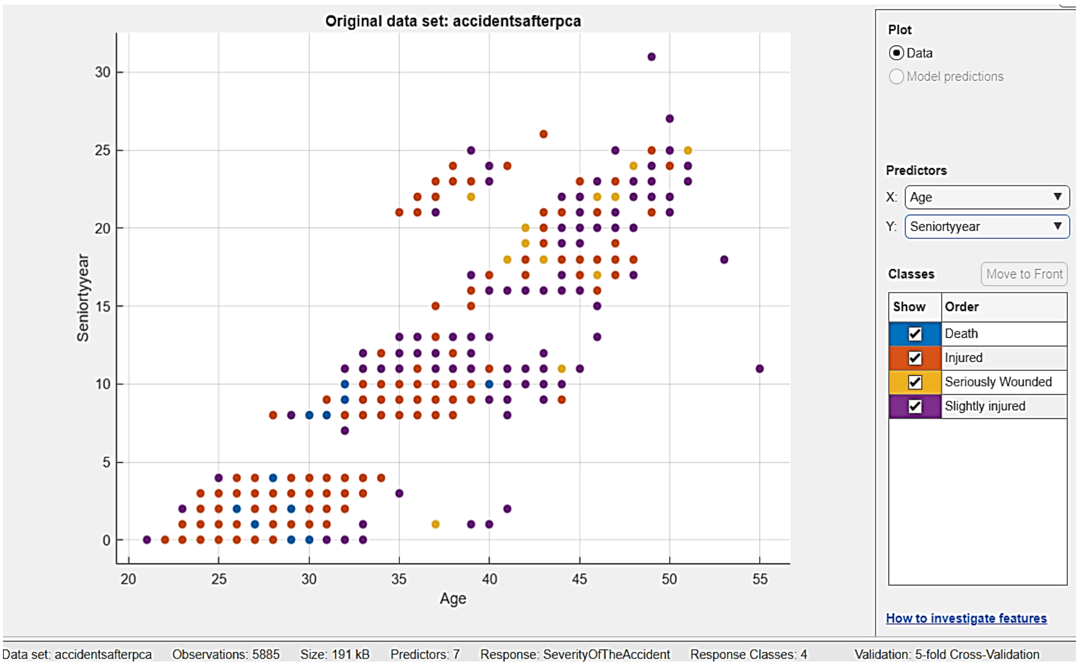
Y: Seniortyyear ▼

Classes

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 20 Scatter Plot (seniority versus location of accidents)


Original data set: accidentsafterpca
Seniortyyear
0
5
10
15
20
25
30
Age
20
25
30
35
40
45
50
55
Move to Front
How to investigate features

Plot

● Data

○ Model predictions

Predictors

X: Age

Y: Seniortyyear

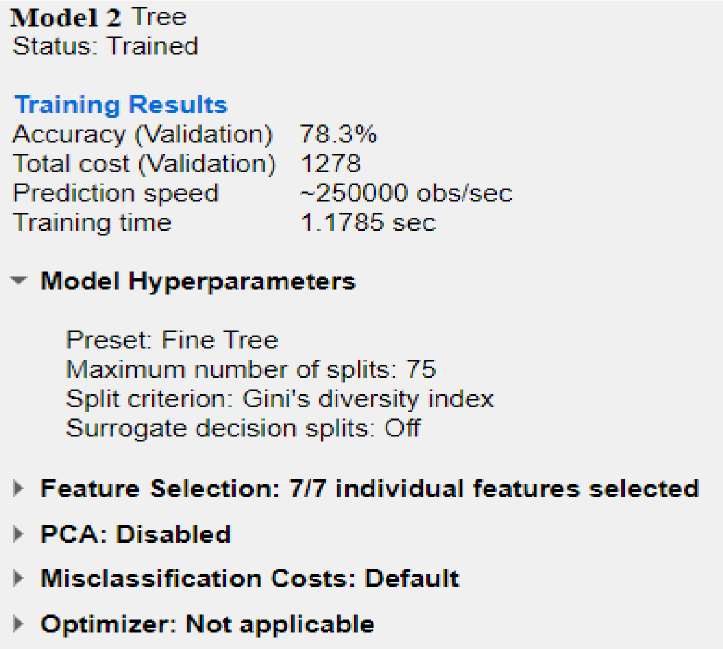
Classes

Show	Order
☑	Death
☑	Injured
☑	Seriously Wounded
☑	Slightly injured

Data set: accidentsafterpca Observations: 5885 Size: 191 kB Predictors: 7 Response: SeverityOfTheAccident Response Classes: 4 Validation: 5-fold Cross-Validation

Figure B 21 Scatter Plot (seniorty versus age)

C. Summary of the Trained Models



Model 2 Tree
Status: Trained

Training Results

Accuracy (Validation)	78.3%
Total cost (Validation)	1278
Prediction speed	~250000 obs/sec
Training time	1.1785 sec

▾ **Model Hyperparameters**

Preset: Fine Tree
Maximum number of splits: 75
Split criterion: Gini's diversity index
Surrogate decision splits: Off

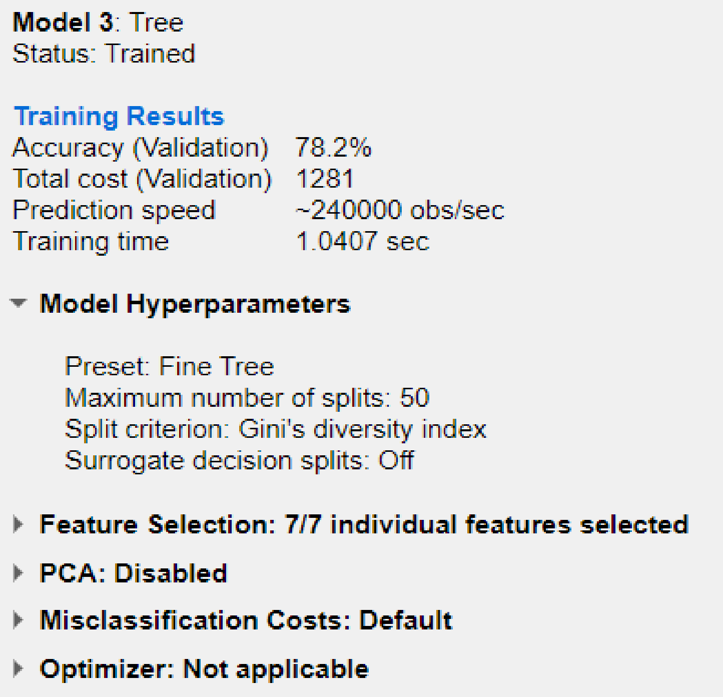
▸ **Feature Selection: 7/7 individual features selected**

▸ **PCA: Disabled**

▸ **Misclassification Costs: Default**

▸ **Optimizer: Not applicable**

Figure C 1 Summary of the DT Model 2



Model 3: Tree
Status: Trained

Training Results

Accuracy (Validation)	78.2%
Total cost (Validation)	1281
Prediction speed	~240000 obs/sec
Training time	1.0407 sec

▾ **Model Hyperparameters**

Preset: Fine Tree
Maximum number of splits: 50
Split criterion: Gini's diversity index
Surrogate decision splits: Off

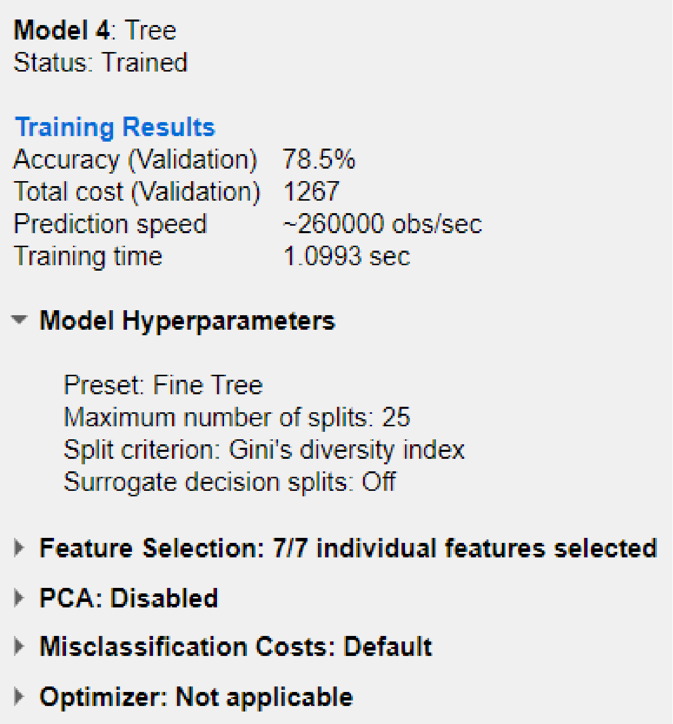
▸ **Feature Selection: 7/7 individual features selected**

▸ **PCA: Disabled**

▸ **Misclassification Costs: Default**

▸ **Optimizer: Not applicable**

Figure C 2 Summary of the DT Model 3



Model 4: Tree
Status: Trained

Training Results

```
Accuracy (Validation)   78.5%
Total cost (Validation) 1267
Prediction speed        ~260000 obs/sec
Training time           1.0993 sec
```

Model Hyperparameters

Preset: Fine Tree
Maximum number of splits: 25
Split criterion: Gini's diversity index
Surrogate decision splits: Off

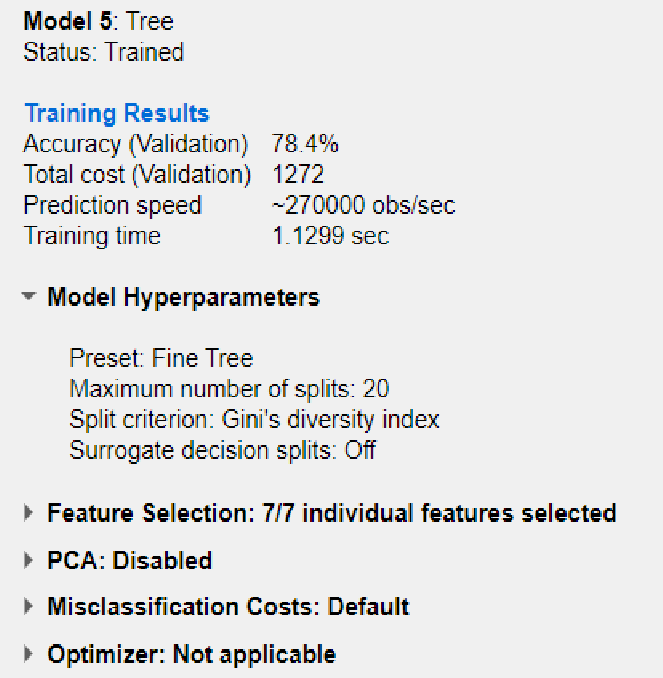
Feature Selection: 7/7 individual features selected

PCA: Disabled

Misclassification Costs: Default

Optimizer: Not applicable

Figure C 3 Summary of the DT Model 4



Model 5: Tree
Status: Trained

Training Results

```
Accuracy (Validation)   78.4%
Total cost (Validation) 1272
Prediction speed        ~270000 obs/sec
Training time           1.1299 sec
```

Model Hyperparameters

Preset: Fine Tree
Maximum number of splits: 20
Split criterion: Gini's diversity index
Surrogate decision splits: Off

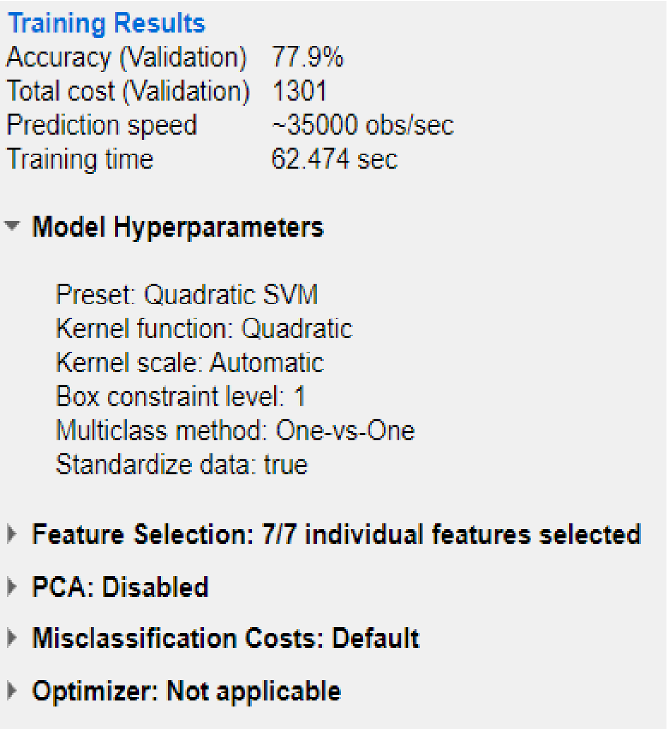
Feature Selection: 7/7 individual features selected

PCA: Disabled

Misclassification Costs: Default

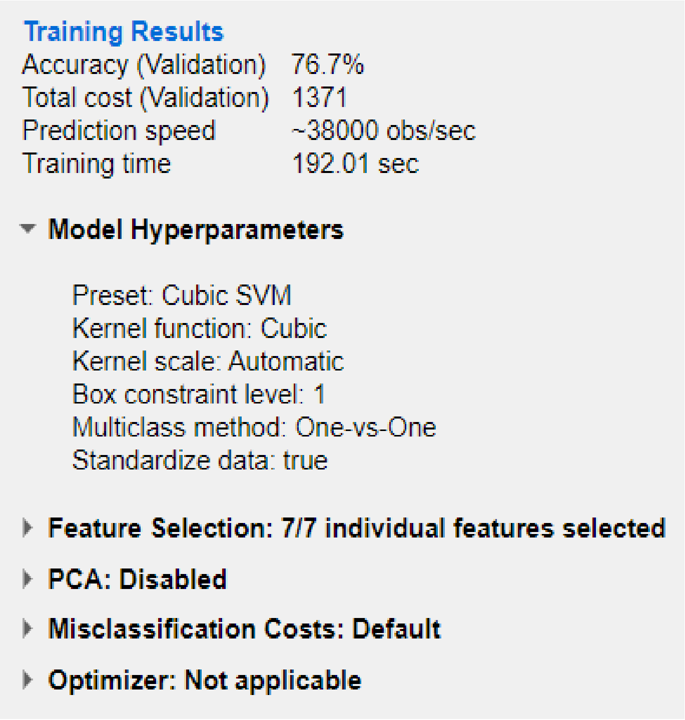
Optimizer: Not applicable

Figure C 4 Summary of the DT Model 5



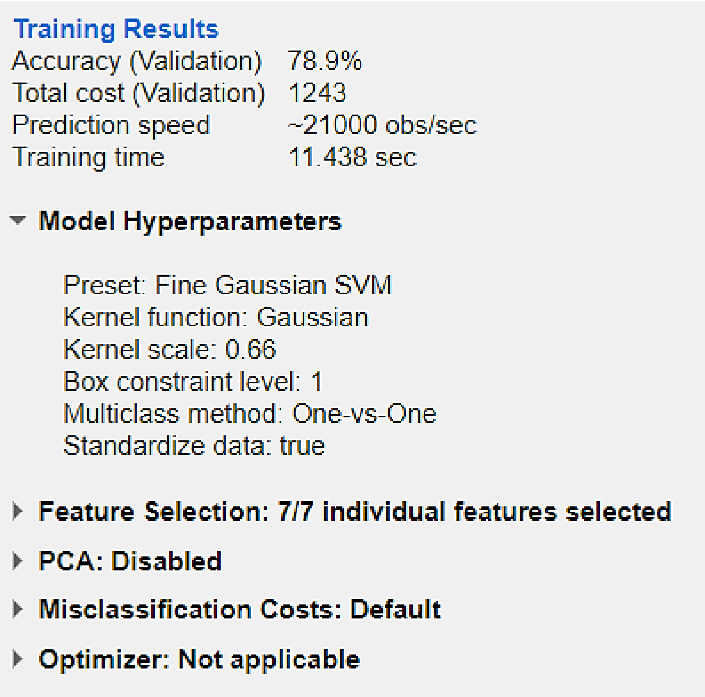
```
Training Results
Accuracy (Validation)   77.9%
Total cost (Validation) 1301
Prediction speed        ~35000 obs/sec
Training time           62.474 sec

▾ Model Hyperparameters

    Preset: Quadratic SVM
    Kernel function: Quadratic
    Kernel scale: Automatic
    Box constraint level: 1
    Multiclass method: One-vs-One
    Standardize data: true

▸ Feature Selection: 7/7 individual features selected

▸ PCA: Disabled

▸ Misclassification Costs: Default

▸ Optimizer: Not applicable
```

Figure C 5 Summary of the quadratic model



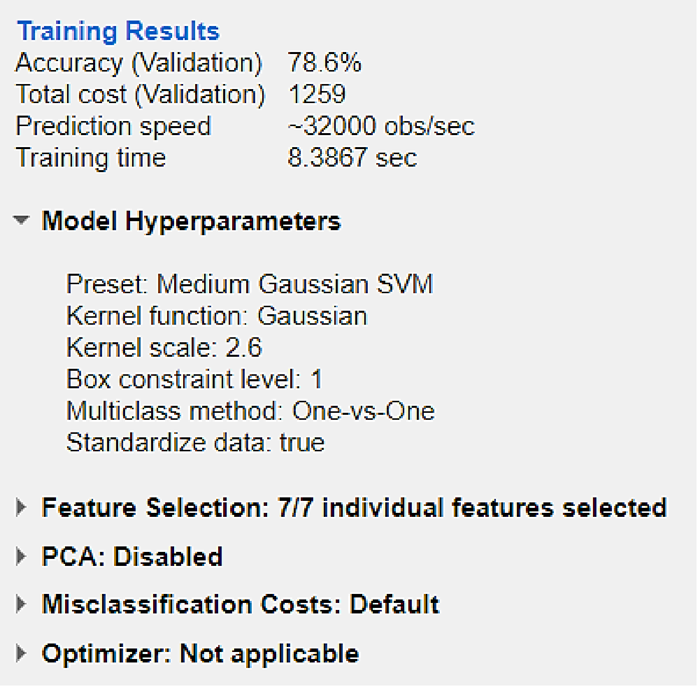
```
Training Results
Accuracy (Validation)   76.7%
Total cost (Validation) 1371
Prediction speed        ~38000 obs/sec
Training time           192.01 sec

▾ Model Hyperparameters

    Preset: Cubic SVM
    Kernel function: Cubic
    Kernel scale: Automatic
    Box constraint level: 1
    Multiclass method: One-vs-One
    Standardize data: true

▸ Feature Selection: 7/7 individual features selected

▸ PCA: Disabled

▸ Misclassification Costs: Default

▸ Optimizer: Not applicable
```

Figure C 6 Summary of the cubic model



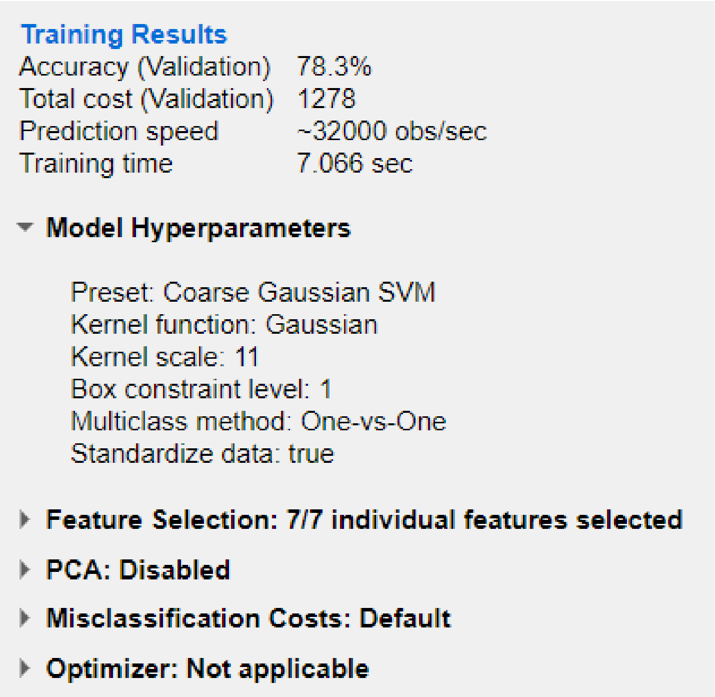
```
Training Results
Accuracy (Validation)   78.9%
Total cost (Validation) 1243
Prediction speed        ~21000 obs/sec
Training time           11.438 sec

▾ Model Hyperparameters

    Preset: Fine Gaussian SVM
    Kernel function: Gaussian
    Kernel scale: 0.66
    Box constraint level: 1
    Multiclass method: One-vs-One
    Standardize data: true

▸ Feature Selection: 7/7 individual features selected
▸ PCA: Disabled
▸ Misclassification Costs: Default
▸ Optimizer: Not applicable
```

Figure C 7 Summary of the fine gaussian model



```
Training Results
Accuracy (Validation)   78.6%
Total cost (Validation) 1259
Prediction speed        ~32000 obs/sec
Training time           8.3867 sec

▾ Model Hyperparameters

    Preset: Medium Gaussian SVM
    Kernel function: Gaussian
    Kernel scale: 2.6
    Box constraint level: 1
    Multiclass method: One-vs-One
    Standardize data: true

▸ Feature Selection: 7/7 individual features selected
▸ PCA: Disabled
▸ Misclassification Costs: Default
▸ Optimizer: Not applicable
```

Figure C 8 Summary of the medium gaussian model



Training Results

Accuracy (Validation) 78.3%
Total cost (Validation) 1278
Prediction speed ~32000 obs/sec
Training time 7.066 sec

▼ **Model Hyperparameters**

Preset: Coarse Gaussian SVM
Kernel function: Gaussian
Kernel scale: 11
Box constraint level: 1
Multiclass method: One-vs-One
Standardize data: true

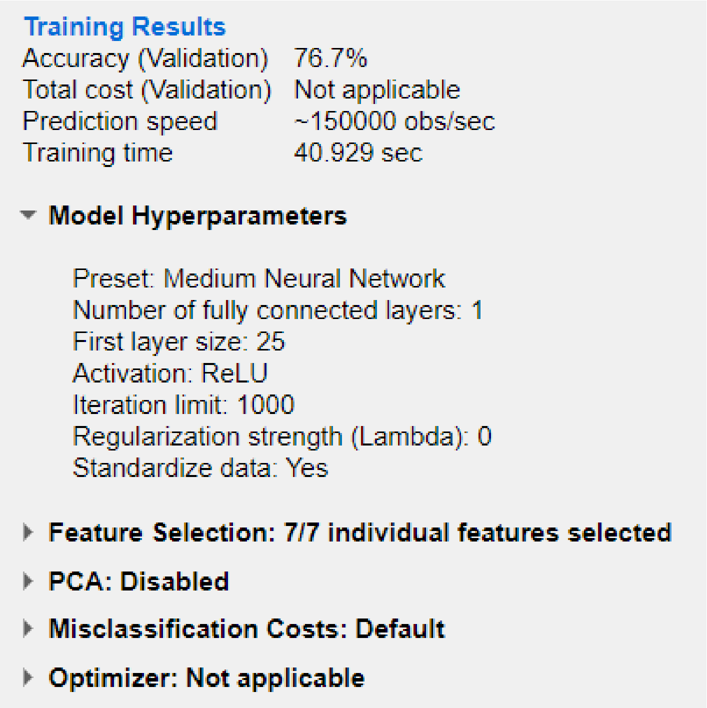
▶ **Feature Selection: 7/7 individual features selected**

▶ **PCA: Disabled**

▶ **Misclassification Costs: Default**

▶ **Optimizer: Not applicable**

Figure C 9 Summary of the coarse gaussian model



Training Results

Accuracy (Validation) 76.7%
Total cost (Validation) Not applicable
Prediction speed ~150000 obs/sec
Training time 40.929 sec

▼ **Model Hyperparameters**

Preset: Medium Neural Network
Number of fully connected layers: 1
First layer size: 25
Activation: ReLU
Iteration limit: 1000
Regularization strength (Lambda): 0
Standardize data: Yes

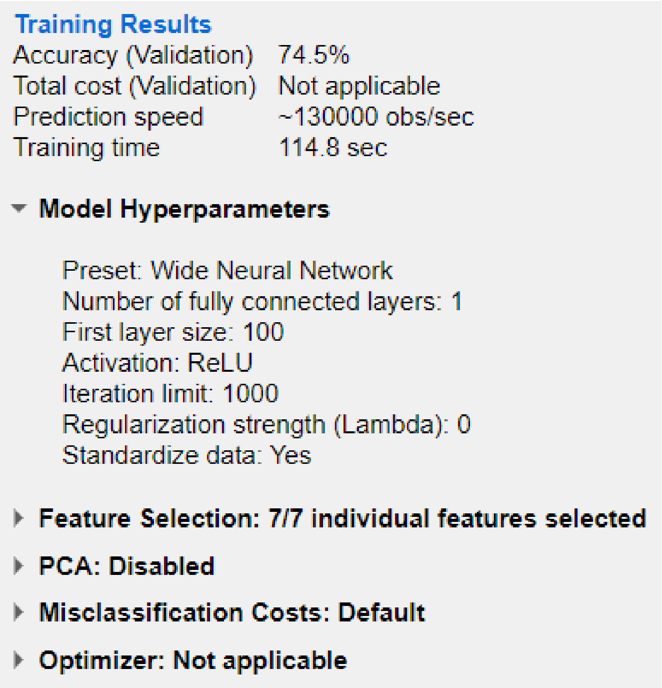
▶ **Feature Selection: 7/7 individual features selected**

▶ **PCA: Disabled**

▶ **Misclassification Costs: Default**

▶ **Optimizer: Not applicable**

Figure C 10 Summary of medium neural network model



Training Results

Accuracy (Validation)   74.5%
Total cost (Validation)   Not applicable
Prediction speed   ~130000 obs/sec
Training time   114.8 sec

▾ **Model Hyperparameters**

Preset: Wide Neural Network
Number of fully connected layers: 1
First layer size: 100
Activation: ReLU
Iteration limit: 1000
Regularization strength (Lambda): 0
Standardize data: Yes

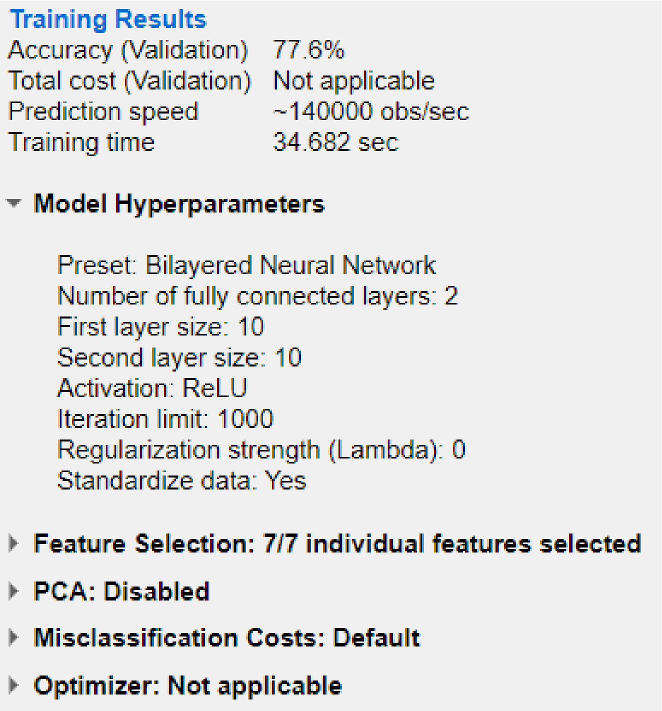
▸ **Feature Selection: 7/7 individual features selected**

▸ **PCA: Disabled**

▸ **Misclassification Costs: Default**

▸ **Optimizer: Not applicable**

Figure C 11 Summary of wide neural network model



Training Results

Accuracy (Validation)   77.6%
Total cost (Validation)   Not applicable
Prediction speed   ~140000 obs/sec
Training time   34.682 sec

▾ **Model Hyperparameters**

Preset: Bilayered Neural Network
Number of fully connected layers: 2
First layer size: 10
Second layer size: 10
Activation: ReLU
Iteration limit: 1000
Regularization strength (Lambda): 0
Standardize data: Yes

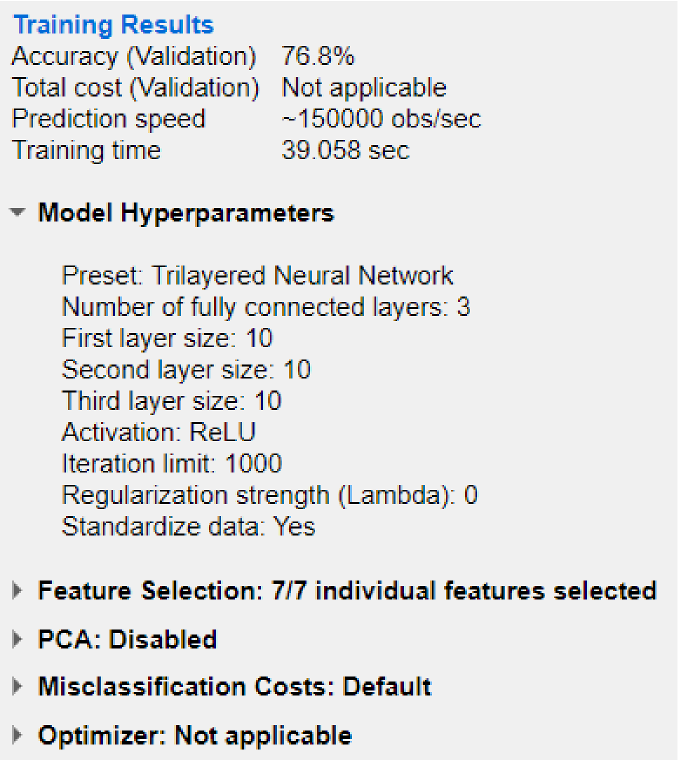
▸ **Feature Selection: 7/7 individual features selected**

▸ **PCA: Disabled**

▸ **Misclassification Costs: Default**

▸ **Optimizer: Not applicable**

Figure C 12 Summary of bilayered neural network model



Training Results

Accuracy (Validation)	76.8%
Total cost (Validation)	Not applicable
Prediction speed	~150000 obs/sec
Training time	39.058 sec

▾ Model Hyperparameters

Preset: Trilayered Neural Network
Number of fully connected layers: 3
First layer size: 10
Second layer size: 10
Third layer size: 10
Activation: ReLU
Iteration limit: 1000
Regularization strength (Lambda): 0
Standardize data: Yes

▸ Feature Selection: 7/7 individual features selected

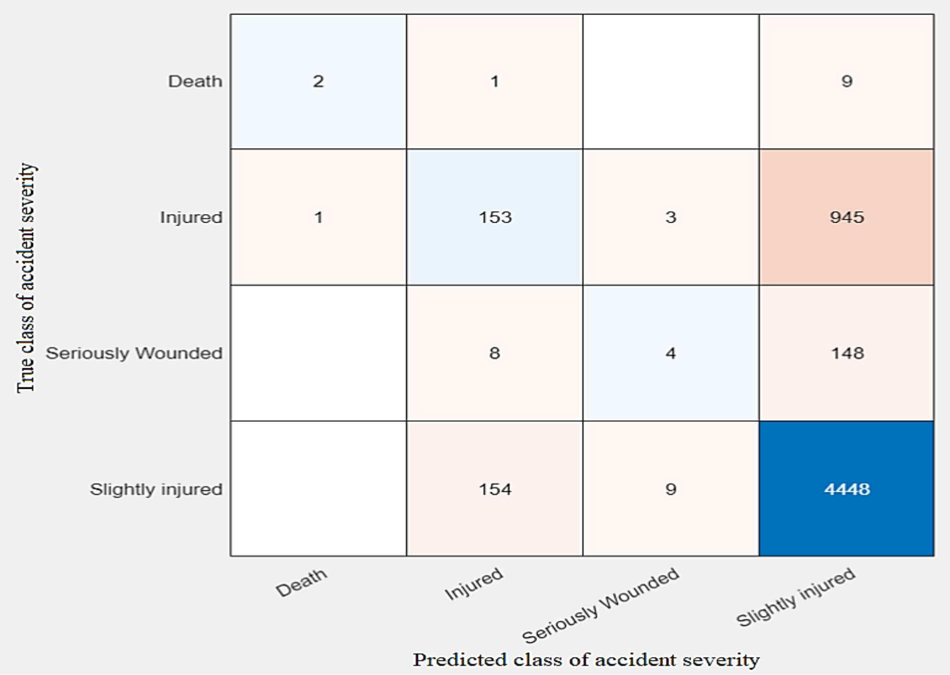
▸ PCA: Disabled

▸ Misclassification Costs: Default

▸ Optimizer: Not applicable

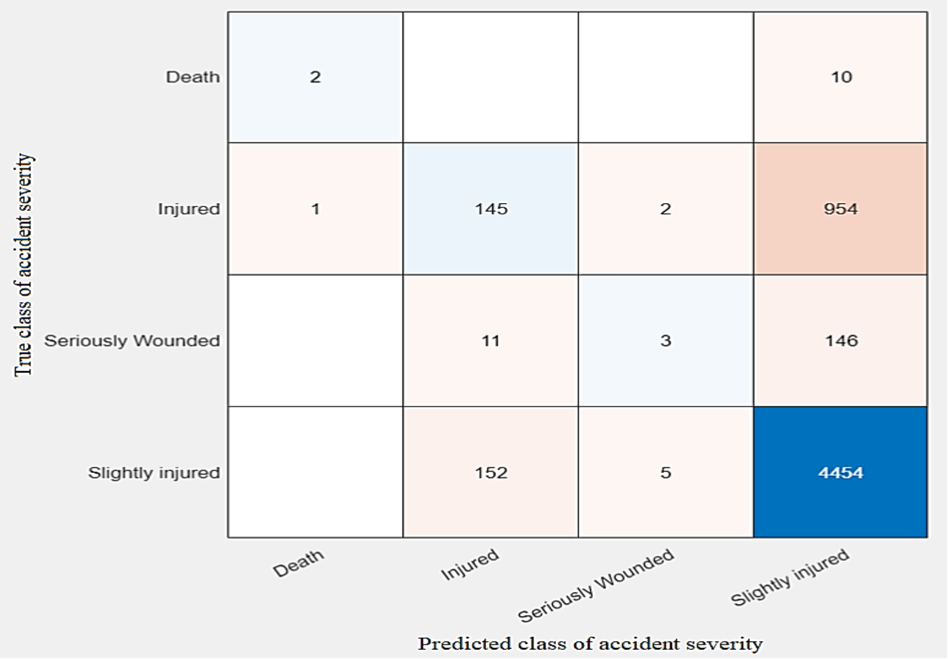
Figure C 13 Summary of trilayered neural network model

D. The Validation Confusion Matrix of the Trained Models



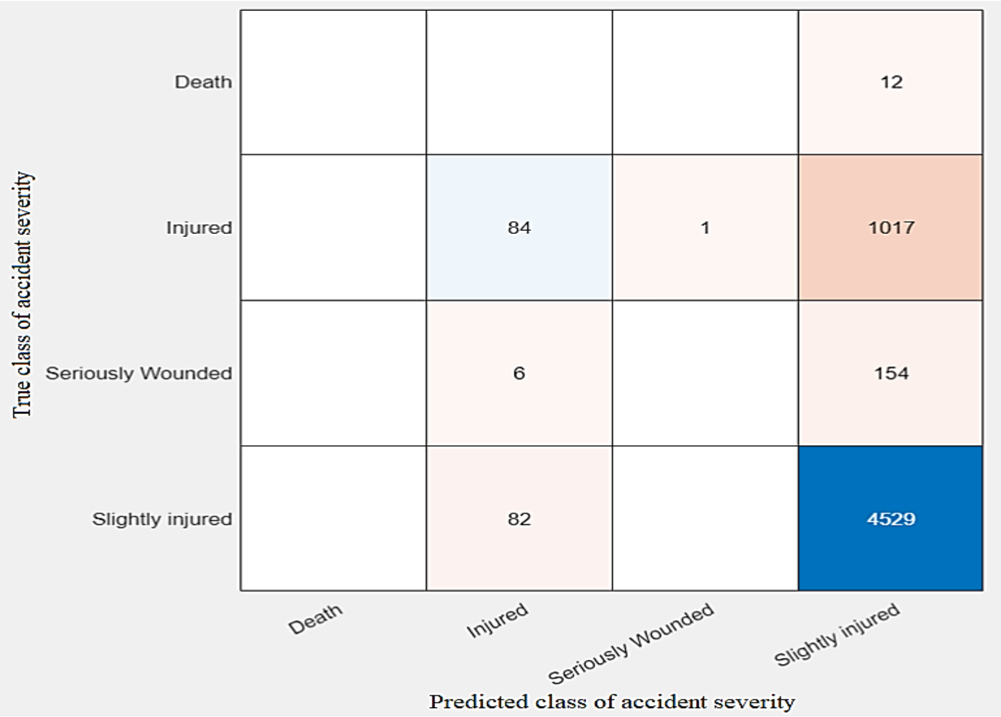
	Death	Injured	Seriously Wounded	Slightly injured
Death	2	1		9
Injured	1	153	3	945
Seriously Wounded		8	4	148
Slightly injured		154	9	4448

Figure D 1 The validation confusion matrix of DT model 2 (number of splits:75)



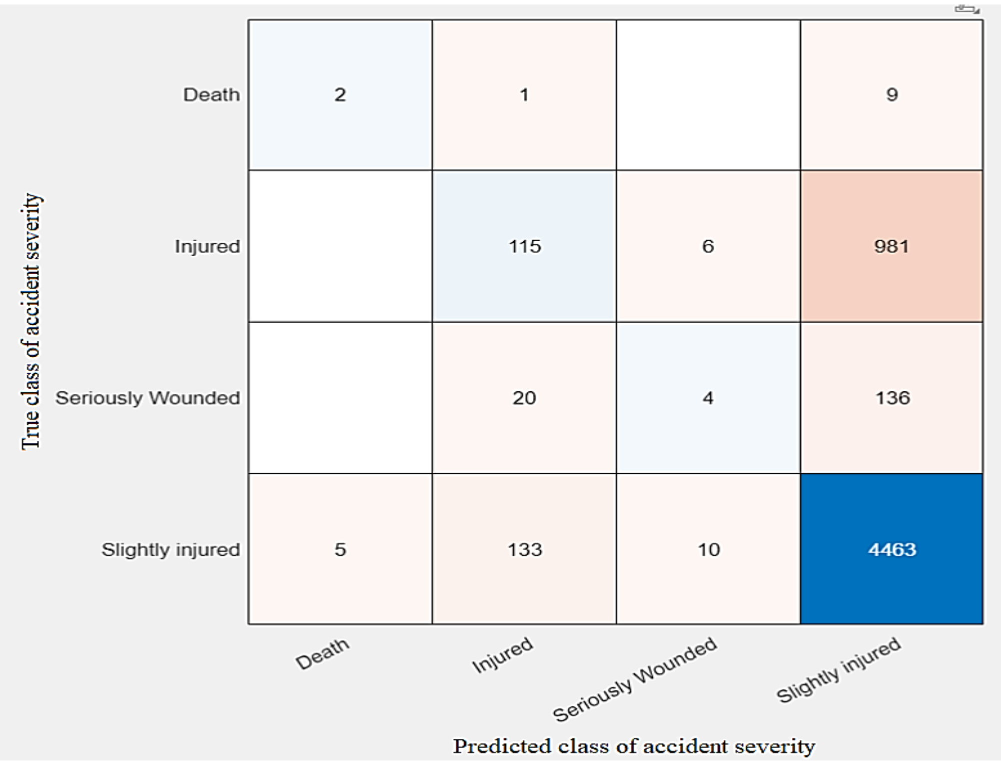
	Death	Injured	Seriously Wounded	Slightly injured
Death	2			10
Injured	1	145	2	954
Seriously Wounded		11	3	146
Slightly injured		152	5	4454

Figure D 2 The validation confusion matrix of DT model 3 (number of splits:50)



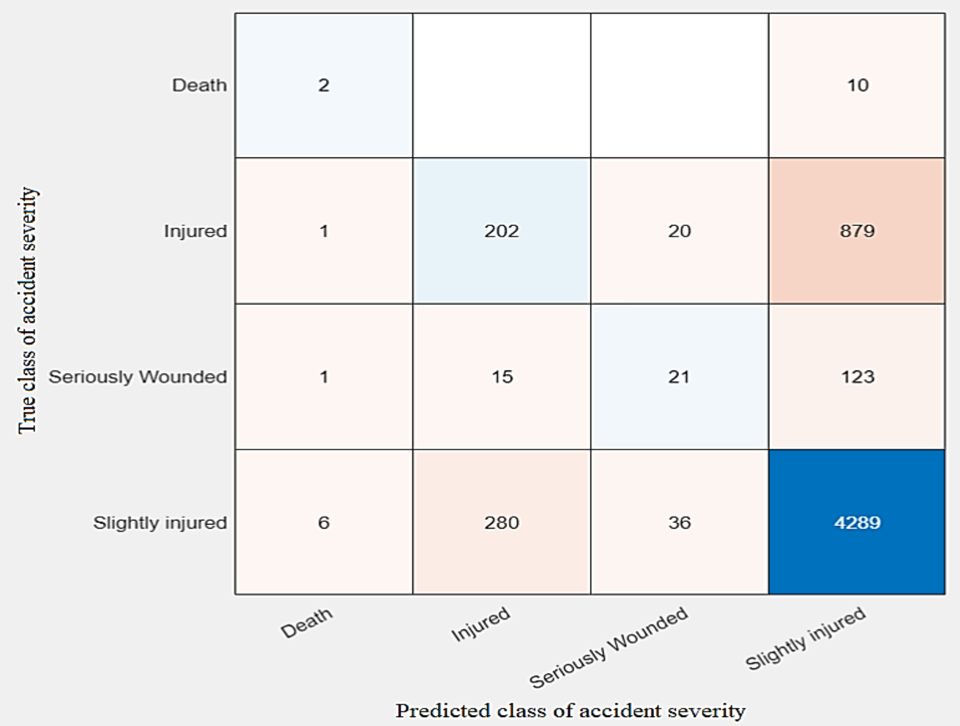
True class of accident severity \ Predicted class of accident severity	Death	Injured	Seriously Wounded	Slightly injured
Death				12
Injured		84	1	1017
Seriously Wounded		6		154
Slightly injured		82		4529

Figure D 3 The validation confusion matrix of DT model 5 (number of splits:20)



True class of accident severity \ Predicted class of accident severity	Death	Injured	Seriously Wounded	Slightly injured
Death	2	1		9
Injured		115	6	981
Seriously Wounded		20	4	136
Slightly injured	5	133	10	4463

Figure D 4 Validation confusion matrix of the quadratic model SVM

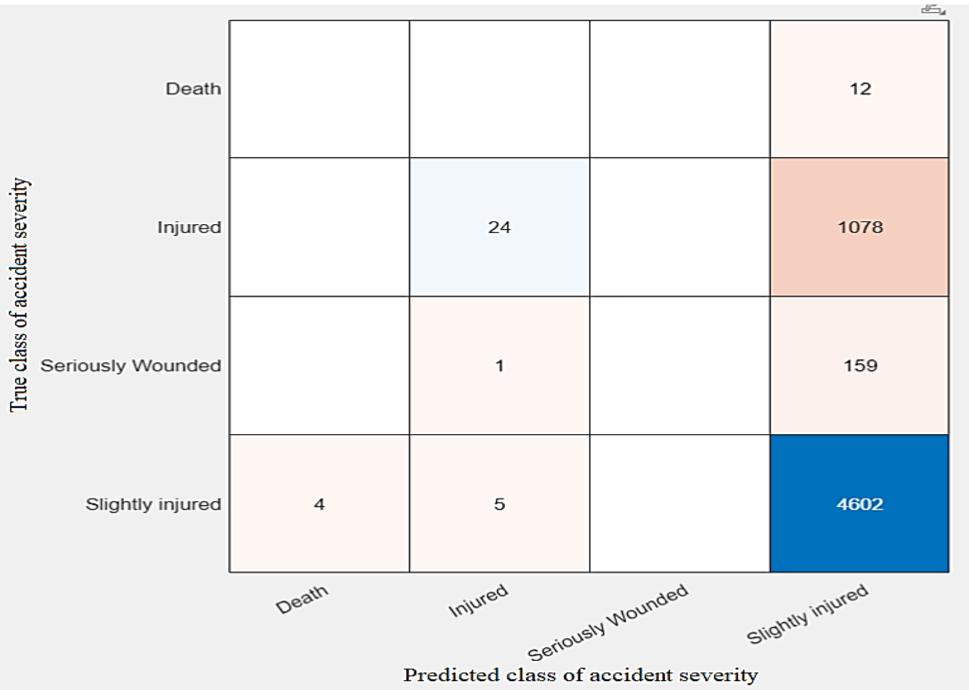


True class of accident severity

Predicted class of accident severity

	Death	Injured	Seriously Wounded	Slightly injured
Death	2			10
Injured	1	202	20	879
Seriously Wounded	1	15	21	123
Slightly injured	6	280	36	4289

Figure D 5 Validation confusion matrix of cubic model SVM



True class of accident severity

Predicted class of accident severity

	Death	Injured	Seriously Wounded	Slightly injured
Death				12
Injured		24		1078
Seriously Wounded		1		159
Slightly injured	4	5		4602

Figure D 6 Validation confusion matrix of medium gaussian model SVM

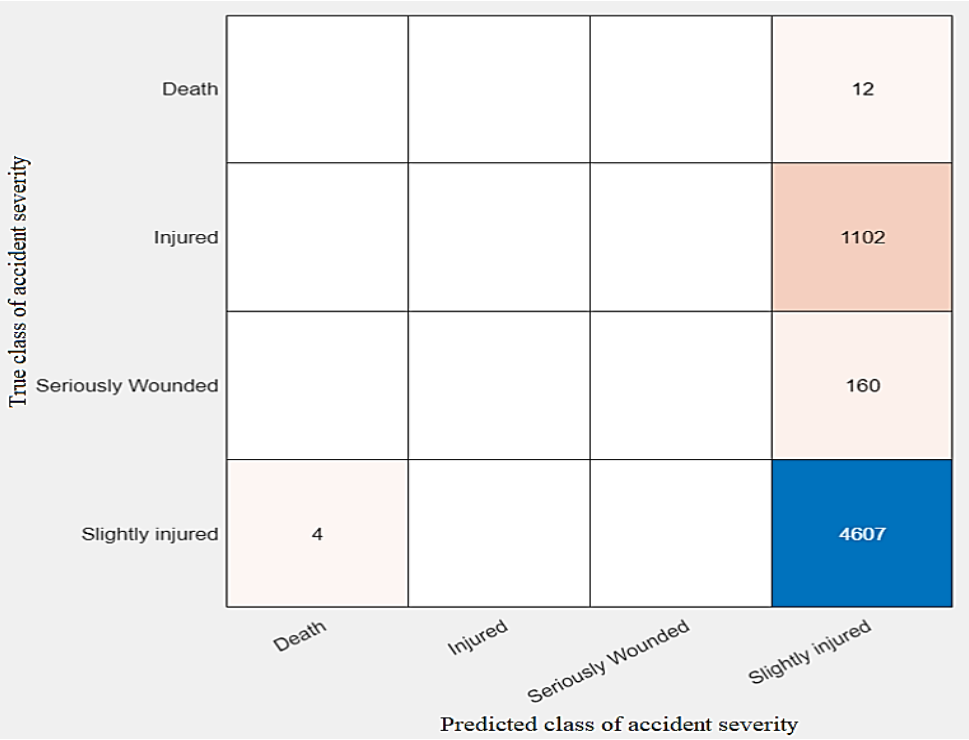


Figure D 7 Validation confusion matrix of coarse gaussian model SVM

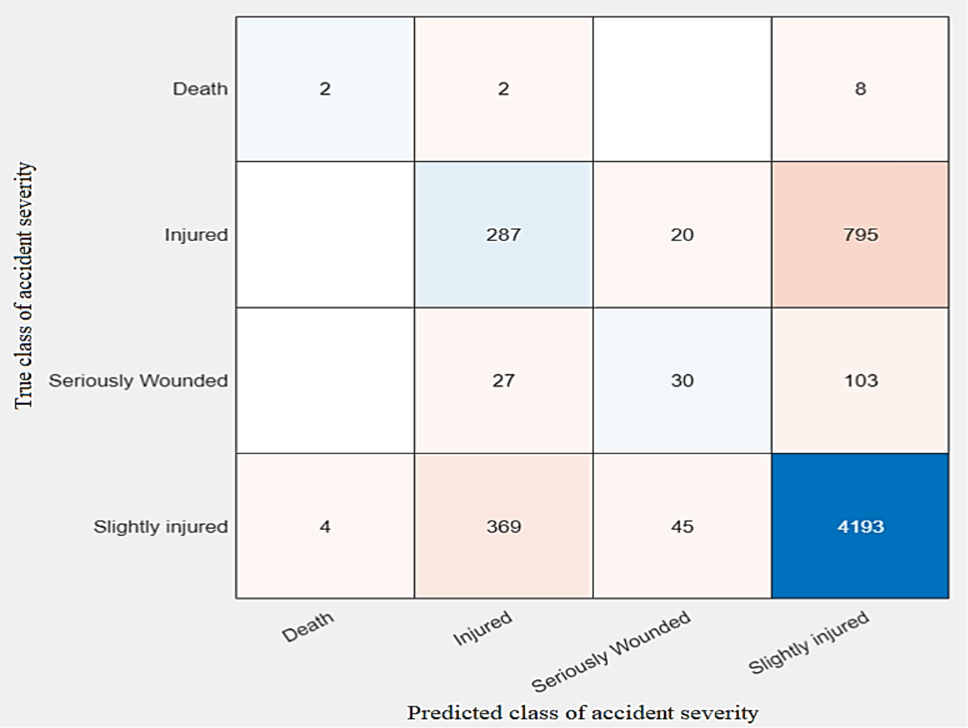
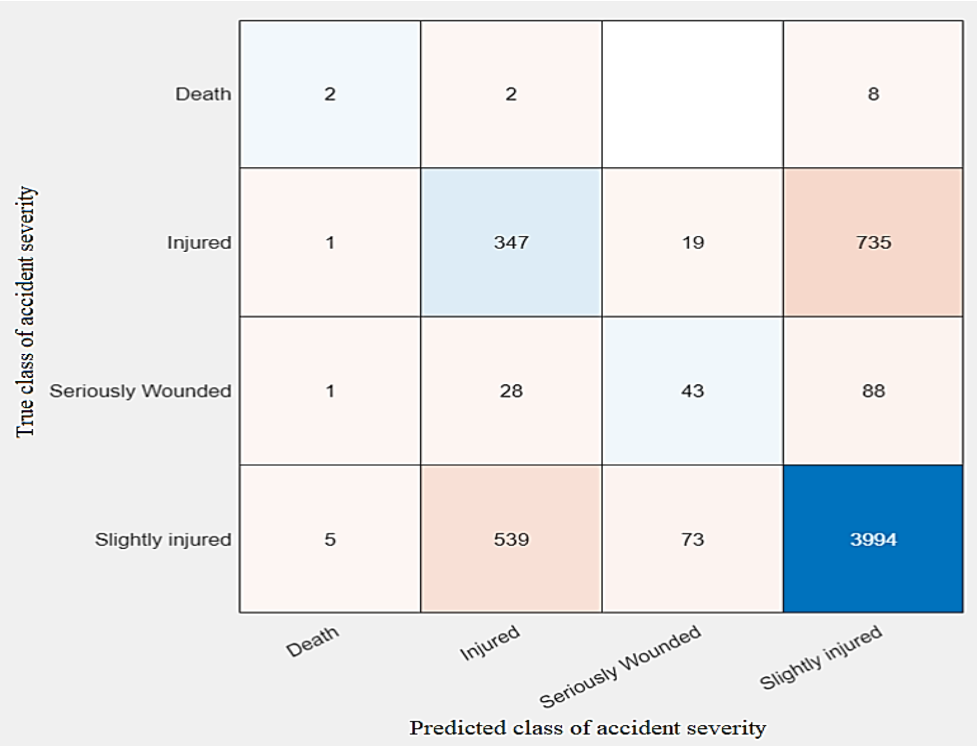
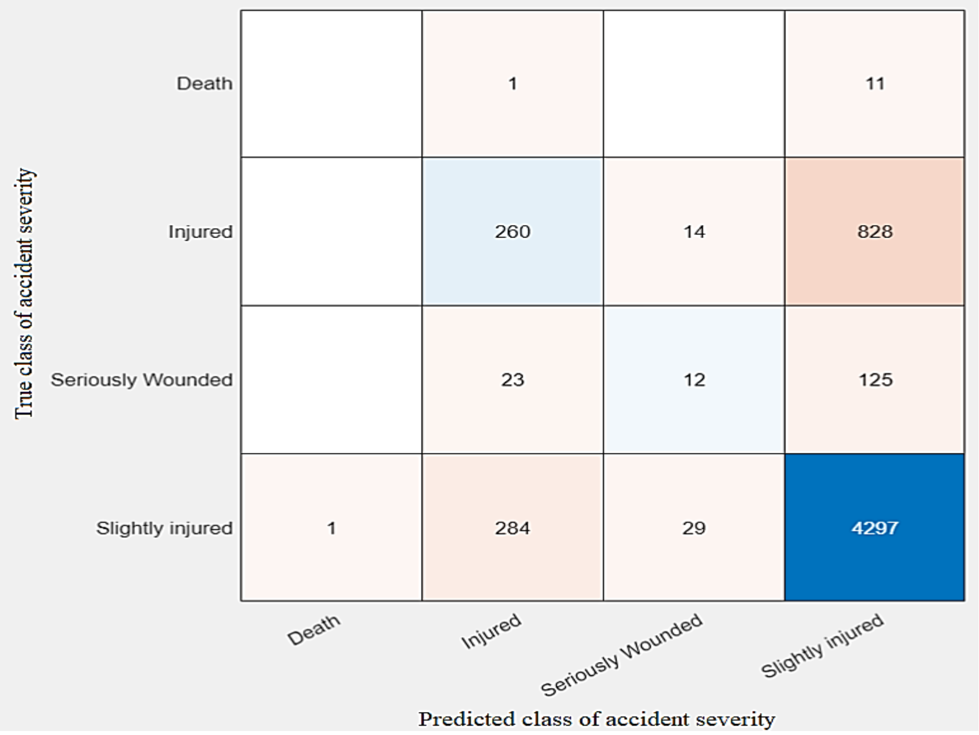


Figure D 8 Validation confusion matrix of medium neural network model



	Death	Injured	Seriously Wounded	Slightly injured
Death	2	2		8
Injured	1	347	19	735
Seriously Wounded	1	28	43	88
Slightly injured	5	539	73	3994

Figure D 9 Validation confusion matrix of wide neural network model



	Death	Injured	Seriously Wounded	Slightly injured
Death		1		11
Injured		260	14	828
Seriously Wounded		23	12	125
Slightly injured	1	284	29	4297

Figure D 10 Validation confusion matrix of bilayered neural network model

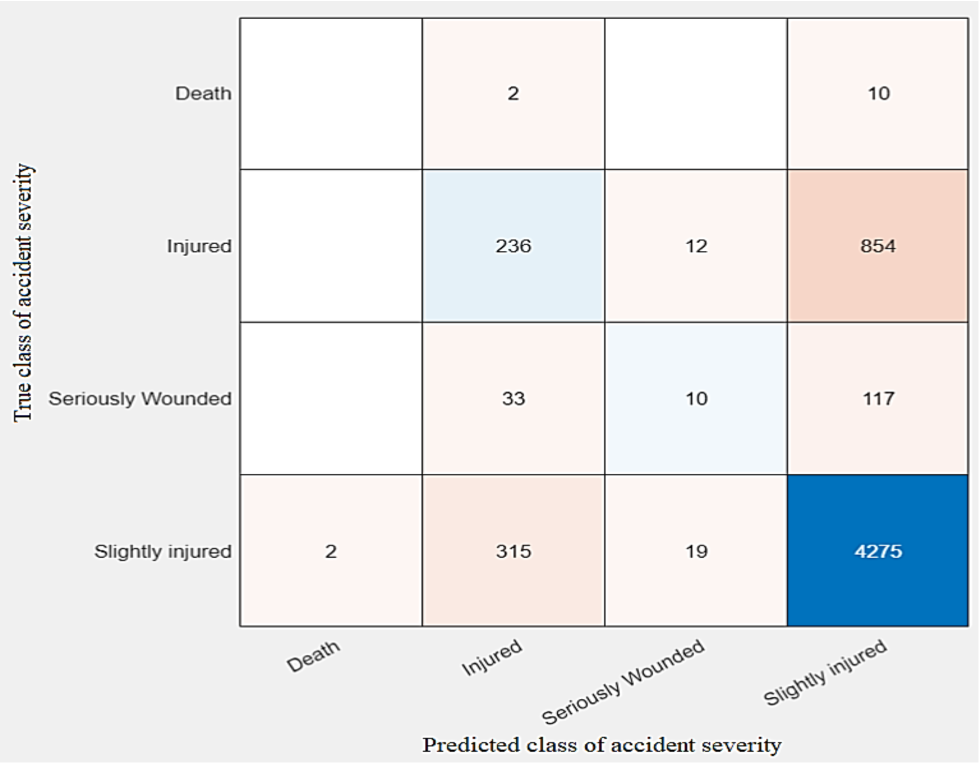


Figure D 11 Validation confusion matrix of trilayered neural network model