**EARTHQUAKE PREDICTION**

**MASTER OF SCIENCE**

**IN**

**BUSINESS ANALYTICS**

**Submitted By**

**GROUP 9**

Harika Manapuram HXM220054

Hema Sri Nelakuditi HXN220021

Sneha Priya Polavarapu SXP220178

Jahnavi Ram Polisetti JXP220020

Jhansi Lakshmi Somarouthu JXS220070

Shravan Yandapalli SXY220048

Under the esteemed guidance of

Prof. Shujing Sun (Business Analytics With R)



**Abstract:**

The "Earthquake Dataset" is a compilation of information about earthquakes from around the globe. The database contains the timing, location, magnitude, and depth of each earthquake, as well as other facts such as the number of persons impacted and the economic effect of the occurrence. The European-Mediterranean Seismological Centre (EMSC) and the United States Geological Survey (USGS) both provided publicly available data that was used to create the dataset. The dataset is designed for use by academics, analysts, and anyone interested in investigating earthquake patterns and trends, as well as developing models and prediction tools for earthquake forecasting and response.

**Data source and Description:**

We have taken the data source from Kaggle (<https://www.kaggle.com/datasets/warcoder/earthquake-dataset>). The dataset covers 782 earthquakes from 1/1/2001 to 1/1/2023. The following table shows the description of the dataset.

|  |  |  |
| --- | --- | --- |
| **No** | **Attribute** | **Description** |
| 1 | title | Title name given to the earthquake |
| 2 | magnitude | The magnitude of the earthquake |
| 3 | date\_time | Date and Time |
| 4 | cdi | The maximum reported intensity for the event range |
| 5 | mmi | The maximum estimated instrumental intensity for the event |
| 6 | alert | The alert level - “green”, “yellow”, “orange”, and “red” |
| 7 | tsunami | "1" for events in oceanic regions and "0" otherwise |
| 8 | sig | A number describing how significant the event is. Larger numbers indicate a more significant event. This value is determined on several factors, including magnitude, maximum MMI, felt reports, and estimated impact |
| 9 | net | The ID of a data contributor. Identifies the network considered to be the preferred source of information for this event. |
| 10 | nst | The total number of seismic stations used to determine earthquake location. |
| 11 | dmin | Horizontal distance from the epicenter to the nearest station |
| 12 | gap | The largest azimuthal gap between azimuthally adjacent stations (in degrees). In general, the smaller this number, the more reliable is the calculated horizontal position of the earthquake. Earthquake locations in which the azimuthal gap exceeds 180 degrees typically have large location and depth uncertainties |
| 13 | magType | The method or algorithm used to calculate the preferred magnitude for the event |
| 14 | depth | The depth where the earthquake begins to rupture |
| 15 | latitude | coordinate system by means of which the position or location of any place on Earth's surface can be determined and described |
| 16 | longitude | coordinate system by means of which the position or location of any place on Earth's surface can be determined and described |
| 17 | location | location within the country |
| 18 | continent | continent of the earthquake hit country |
| 19 | country | affected country |

**Goal of the Project:**

The goal of the project is to predict the areas at higher risk of earthquakes and estimate the likelihood of future seismic events. Ultimately, the aim is to develop a reliable and accurate predictive model that can help governments and emergency services prepare for and respond to earthquakes, potentially saving lives and reducing damage. We will explore the data to answer the following questions.

The project will involve several stages, including data cleaning and preparation, feature engineering, model building and validation.

Data Exploration – we will analyze the data using statistical measures and visual aids to explore the following questions:

1. What are the primary factors of the frequency and intensity of earthquakes?
2. The correlation between the magnitude, tsunami, date/time, depth, latitude/longitude, and other characteristics of the earthquake (column: cdi),

Prediction Model – we will utilize various classification models to anticipate the severity of the earthquake:

1. Data Understanding and Preparation: we will carry out tasks such as data cleaning, merging, normalization, exploration, handling missing values, and feature engineering.
2. Modeling: we will use classification algorithms to forecast the intensity of earthquakes (as indicated by the "cdi" column), using the important attributes identified during the data exploration process.
3. Validation: Validate the model performance on the test data and generate performance metrics.
4. Conclusion: What do we learn from the model in earthquake prediction? How can we use the predictive model to inform disaster response efforts and prioritize resources in areas at higher risk of earthquakes at times, to mitigate potential damage and save lives?

**WORKING:**

INITIAL SETUP – LOADING THE DATA:

Graphical user interface, text, application, email

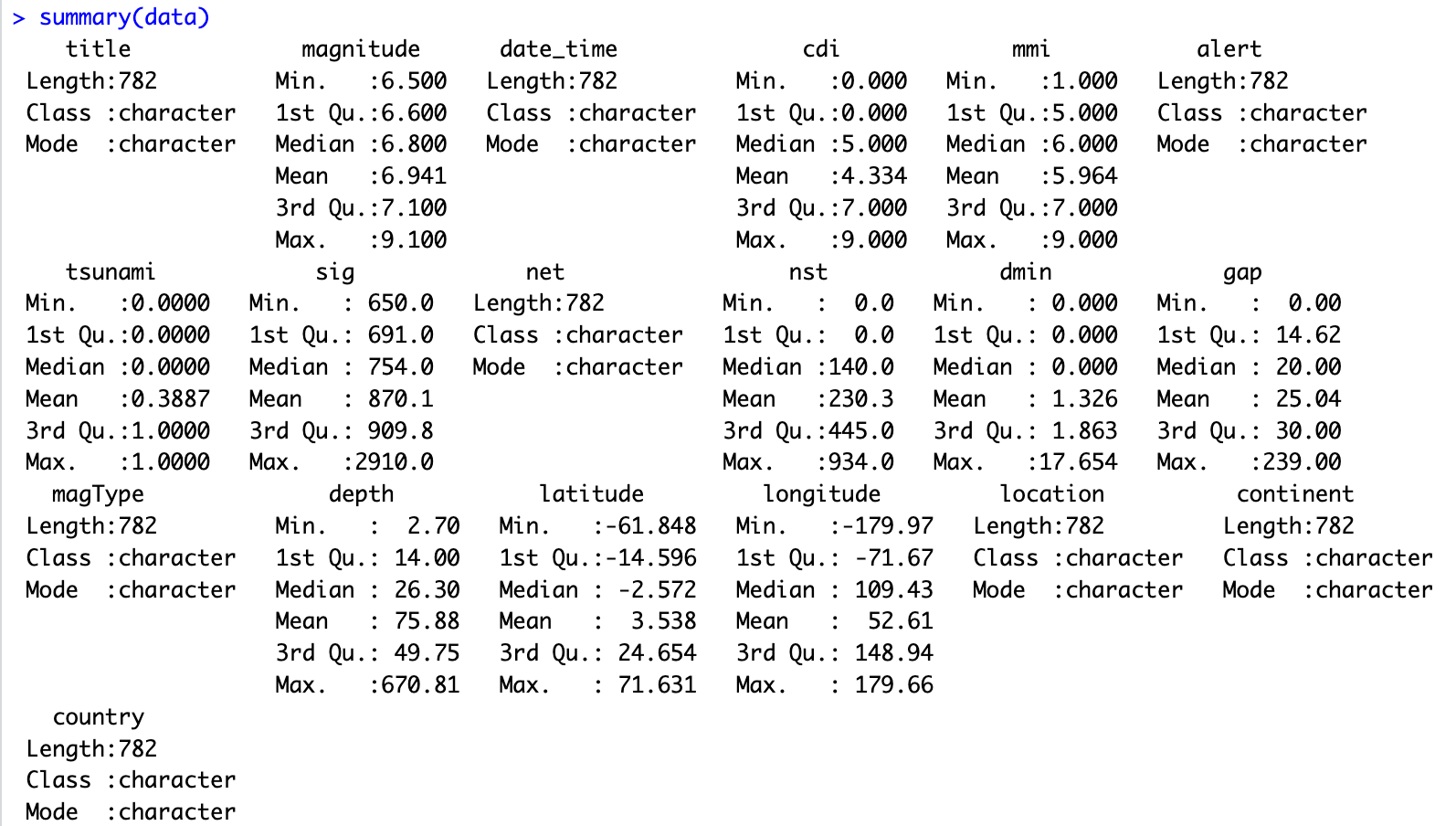
Description automatically generated

rm(list=ls()) is used in R to clear the workspace by removing all the objects that are currently stored.

cat("\014") is an R command that clears the console screen.

Read.csv is used to load the earthquake data from the working directory that is set.

SUMMARY OF DATA:



In R, the summary() function is used to generate summary statistics of the earthquake data frame or matrix. When applied to a data frame, summary() provides a quick way to get an overview of the distribution of values for each variable.

Text

Description automatically generated

The str() function is used to display the internal structure of an earthquake dataset. This will produce a summary of the variables' names and data types, as well as the number of observations in the data frame. The earthquake dataset has 782 observations with 19 variables. The str() function is particularly useful when working with large datasets or complex objects in R, as it allows you to quickly understand the structure of the object and its contents.

**DATA VISUALIZATION:**

Graphical user interface, text, application, chat or text message

Description automatically generated

The command, hist(data$cdi, breaks = 10), creates a histogram of the "cdi" variable from the earthquake data. The breaks = 10 argument specifies that the histogram should be divided into 10 bins.

The command, data$cdi <- cut(data$cdi, breaks = c(-1,3,6,10), labels = c("-1", "0", "1")), changes the "cdi" variable from a numeric variable to a factor variable. The cut() function is used to split the "cdi" variable into three categories based on the specified breaks. The labels for each category are "-1 (negative)", "0 (neutral)", and "1 (positive)".

The next commands, hist(data$magnitude, breaks = 10), hist(data$depth, breaks = 10), and hist(data$sig, breaks = 10), create histograms of the "magnitude", "depth", and "sig" variables, respectively. Each histogram is divided into 10 bins.

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generated

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generated

**CHECKING FOR OUTLIERS:**

Graphical user interface, text, application

Description automatically generated

The boxplot() function can be used to create a box plot of a variable or set of variables.These box plots will show how each variable is distributed. The whiskers extend to the minimum and highest values that are not regarded as outliers, while the box itself shows the distribution's interquartile range (IQR). Outliers are any data points that are outside of the whiskers and are shown as separate points.

Chart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generated

Chart, diagram, schematic

Description automatically generated

**SPLITTING THE DATA:**

Text

Description automatically generated

The earthquake dataset is divided into training and testing sets using the provided code in order to create and assess earthquake prediction models.

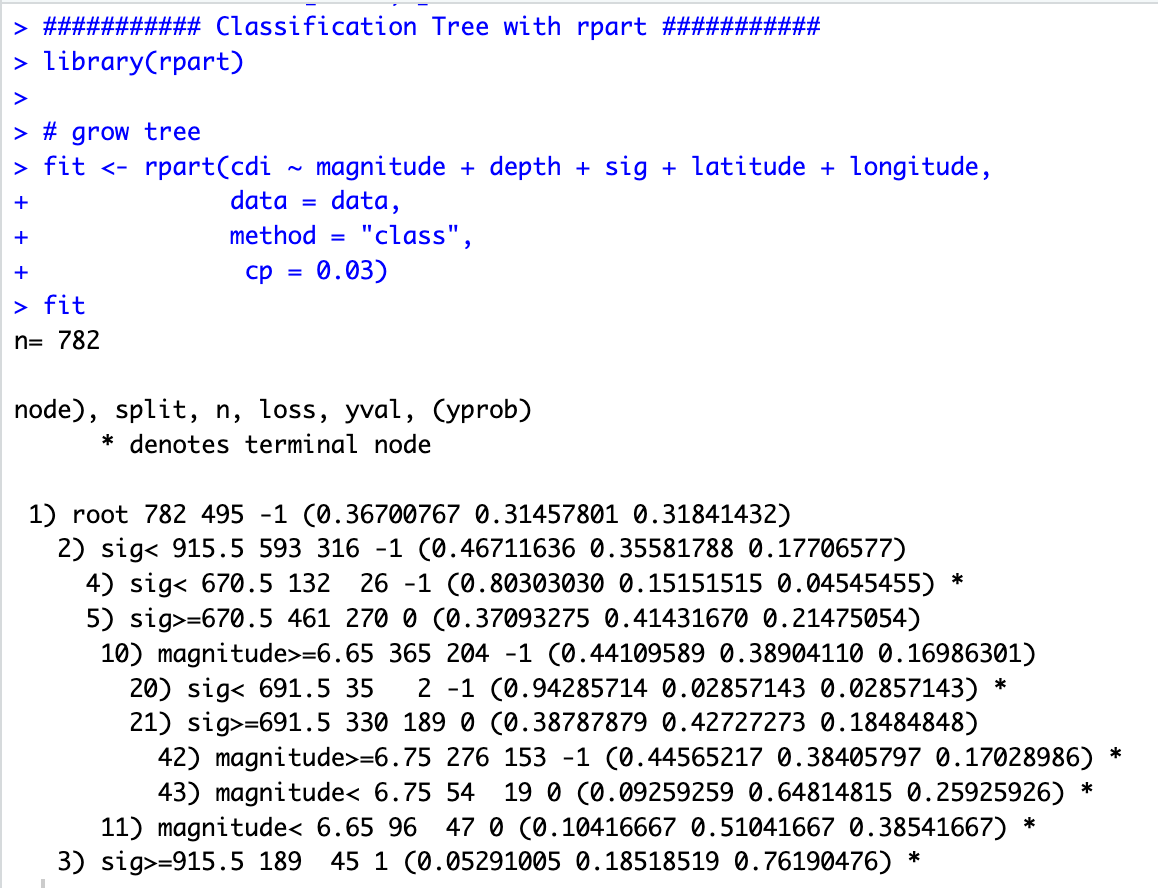
The set.seed(123) command is used to set the seed value for the random number generator in R, ensuring that the randomization is the same every time the code is run. This helps to ensure reproducibility of results.

The train <- sample(1:nrow(data), nrow(data)\*(2/3)) command generates a random sample of row indices from the earthquake dataset. Specifically, it samples 2/3 of the total number of rows in the earthquake dataset. This creates a "train" vector which contains the randomly selected row indices that will be used to subset the earthquake dataset into the training set.

The data.train <- data[train, ] command subsets the earthquake dataset using the "train" vector to create a training set. The resulting dataset "data.train" contains the randomly selected 2/3 of the original data i.e., 521 rows which will be used to build prediction models.

The data.test <- data[-train, ] command creates a testing set by subsetting the earthquake dataset using the complement of the "train" vector. Specifically, it includes all rows that were not included in the "train" vector, resulting in a dataset "data.test" that contains the remaining 1/3 of the original data i.e., 261 rows.This dataset will be used to assess how well the model performs on fresh, untested data.

**DECISION TREE:**



The rpart() function from the "rpart" is to grow a decision tree model for earthquake dataset.

The first line of the code, library(rpart), loads the "rpart" package into the R session, which is required to use the rpart() function.

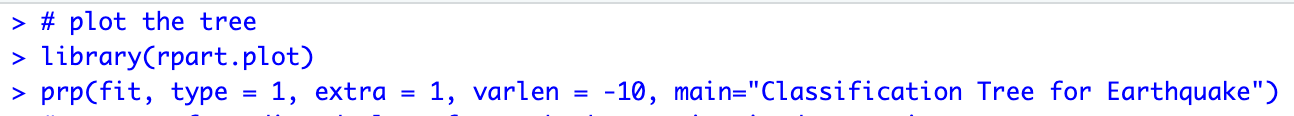
In this case, we are using "cdi" as the response variable, and "magnitude", "depth", "sig", "latitude", and "longitude" as the predictor variables.

The argument, data = data, specifies the data frame from which the variables in the formula are to be taken.

The argument, method = "class", specifies that this is a classification problem, where the response variable is categorical.

The fourth argument, cp = 0.03, specifies the complexity parameter, which controls the level of pruning of the decision tree.

**PLOTTING THE TREE:**



The code provided uses the rpart.plot() function from the "rpart.plot" package in R to plot the decision tree model created in the previous step.

The first line of the code, library(rpart.plot), loads the "rpart.plot" package into the R session, which is required to use the rpart.plot() function.

The rpart.plot() function takes several arguments. The first argument, fit, is the fitted decision tree model object created in the previous step.

The second argument, type = 1, specifies that the plot should be of type 1, which displays the nodes in a rectangular shape.

The third argument, extra = 1, specifies that extra information should be displayed in the plot, such as the number of observations in each node and the percentage of observations that fall into each category.

The fourth argument, varlen = -10, specifies the maximum length of variable names in the plot. In this case, a value of -10 is specified, which allows for variable names of any length.

The fifth argument, main="Classification Tree for Earthquake", specifies the main title for the plot.

Diagram

Description automatically generated

**CONFUSION MATRIX FOR TRAINING DATA OF DECISION TREE:**

Table

Description automatically generated with low confidence

The first line of the code, pred <- predict(fit, data.train, type = "class"), uses the predict() function to generate a vector of predicted class labels for each observation in the training dataset (data.train). The fit object contains the decision tree model, and type = "class" specifies that the predicted values should be class labels rather than probabilities.

The second line of the code, actual <- data.train$cdi, creates a vector of the actual class labels for each observation in the training dataset.

The third line of the code, confusion.matrix <- confusionMatrix(pred, actual, positive = "1"), uses the confusionMatrix() function from the "caret" package to create a confusion matrix that summarizes the performance of the model. The pred and actual vectors are used as input, and positive = "1" specifies that "1" is the positive class label. The resulting confusion matrix is assigned to the confusion.matrix variable.

The confusion matrix is shown in the R console by the code's last line, confusion.matrix. The confusion matrix displays different performance indicators, including accuracy, sensitivity, and specificity, as well as the number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions made by the model.

**CONFUSION MATRIX FOR TESTING DATA OF DECISION TREE:**

Table

Description automatically generated

**LOGISTIC REGRESSION:Text

Description automatically generated**

When the algorithm has trouble fitting the data, resulting in probability that are extremely close to 0 or 1, the logistic regression warning message "glm.fit: fitted probabilities numerically 0 or 1 occurred" appears.

**PLOTTING LOGISTIC REGRESSION:**

**Chart

Description automatically generated Chart, line chart

Description automatically generated**

**Chart

Description automatically generated with medium confidence Chart, scatter chart

Description automatically generated**

**CONFUSION MATRIX FOR TRAINING DATA OF LOGISTIC REGRESSION:**

**Table

Description automatically generated**

**CONFUSION MATRIX FOR TESTING DATA OF LOGISTIC REGRESSION:**

**Table

Description automatically generated**

**K – NEAREST NEIGHBOURS:**

**Graphical user interface, text, application, email

Description automatically generated**

In the code above, the caret package is loaded which is used for machine learning tasks. The trainControl() function is used to set the parameters for 10-fold cross-validation.

The train() function is used to train the K-Nearest Neighbors (KNN) model using the training data. The cdi column is used as the response variable, and the magnitude, depth, sig, latitude, and longitude columns are used as predictors.

The method parameter is set to "knn", indicating that the KNN algorithm will be used to fit the model. The trControl parameter is set to the ctrl object that was created earlier, specifying the 10-fold cross-validation. The preProcess parameter is set to c("center", "scale"), which means that the data will be centered and scaled before the model is fit. The tuneGrid parameter is set to expand.grid(k = 1:10), which specifies that the model will be trained using values of k ranging from 1 to 10.

The train() function returns an object of class train, which contains information about the trained model, including the optimal value of k based on cross-validation. This object is stored in the knnFit variable.

**PLOTTING K – NEAREST NEIGHBOURS:**

**Chart, line chart

Description automatically generated**

**CONFUSION MATRIX FOR TRAINING DATA OF K – NEAREST NEIGHBOURS:**

**Table

Description automatically generated with medium confidence**

**CONFUSION MATRIX FOR TESTING DATA OF K – NEAREST NEIGHBOURS:**

**Table

Description automatically generated**

**NAIVE BAYES CLASSIFIER:**

**Graphical user interface, text, application

Description automatically generated**

**A picture containing text, receipt, screenshot

Description automatically generated**

**CONFUSION MATRIX FOR TRAINING DATA OF NAIVE BAYES CLASSIFIER:**

**Table

Description automatically generated with medium confidence**

**CONFUSION MATRIX FOR TESTING DATA OF NAIVE BAYES CLASSIFIER:**

**Text

Description automatically generated with medium confidence**

**SUPPORT VECTOR MACHINE:**

**Graphical user interface, text, application, email

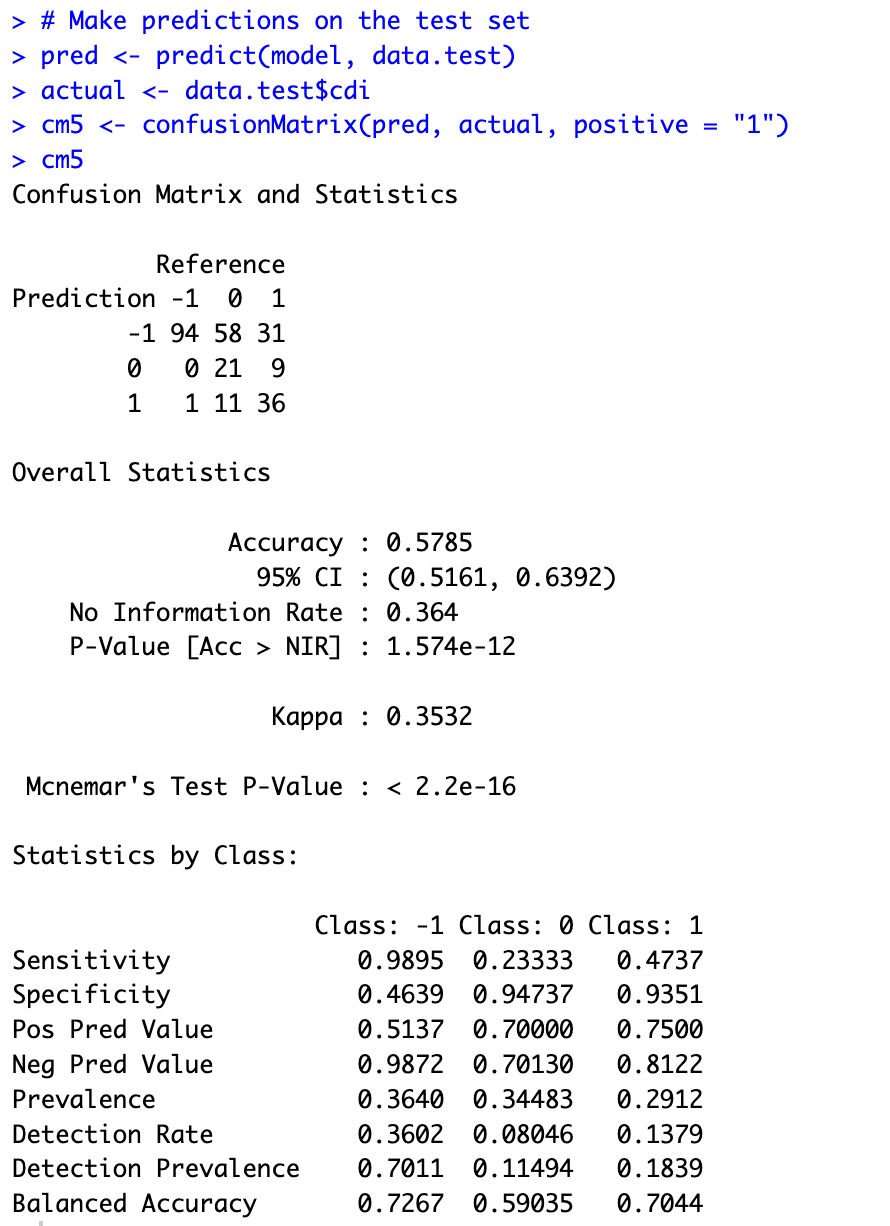
Description automatically generated**

**CONFUSION MATRIX FOR TRAINING DATA OF DECISION SUPPORT VECTOR MACHINE:**

**A screenshot of a computer

Description automatically generated with medium confidence**

**CONFUSION MATRIX FOR TESTING DATA OF SUPPORT VECTOR MACHINE:**

****

**RESULTS:**

**COMPARASION ACROSS DIFFERENT METHODS:**

**CONSIDERING -1 AS THE “NEGATIVE” CLASS:**

**Text

Description automatically generated**

We may deduce from the above result that the Logistic Regression Model has a greater accuracy for the low intensity classification, with an accuracy of 77.91%. The highest Specificity, 0.95, is likewise found in the Logistic Regression Model. Specificity is the percentage of actual negative situations that the model properly identified as being negative, whereas accuracy is the percentage of true predictions provided by the model.

From the above table, it appears that logistic regression performed better than the other models in predicting low intensity (negative class sentiment), as it achieved higher accuracy and specificity scores. This suggests that the logistic regression model is better at distinguishing negative cases from positive cases, which is particularly important in applications where the negative cases are more important or have a greater impact.

**CONSIDERING 0 AS THE “NETURAL” CLASS:**

**Text

Description automatically generated**

From the above result we can infer that the accuracy of the KNN Model is higher for the medium intensity classification with an accuracy of 62.72% and Specificity of SVM Model has the highest Specificity of 0.95.From the above table, it appears that KNN performed better than the other models in predicting medium intensity (neutral class sentiment) in terms of accuracy, while SVM performed better in terms of specificity.

KNN is a non-parametric model that can be effective when the decision boundary is complex. It works by classifying new data points based on their similarity to nearby data points in the training set. The accuracy score indicates the proportion of correctly classified instances, so a higher accuracy for KNN suggests that it can more accurately identify instances of medium intensity.

On the other hand, SVM is a parametric model that seeks to find the optimal hyperplane that separates the classes in feature space. It has a regularization parameter that can help to avoid overfitting, making it potentially useful for high-dimensional datasets. The percentage of negative cases that were correctly classified as negative is shown by the specificity score. A higher specificity for SVM suggests that it can better identify negative cases in the medium intensity class.

**CONSIDERING 0 AS THE “POSITIVE” CLASS:**

Text

Description automatically generated

From the above result we can infer that the accuracy of the KNN Model is higher for the high intensity classification with an accuracy of 75.1% and Specificity of SVM Model has the highest Specificity of 0.93. Based on the statement provided, it appears that logistic regression performed better than the other models in predicting high intensity (positive class sentiment) in terms of accuracy, while SVM performed better in terms of specificity.

**ADVANTAGES:**

* By identifying the models that perform best for each class of intensity, we can create a more nuanced understanding of earthquake risk. This can help governments and emergency services to better prioritize their resources and efforts in areas that are at higher risk of experiencing earthquakes of a particular intensity. For example, if we know that logistic regression is better at predicting high intensity earthquakes, then emergency services can focus their resources on areas that are at higher risk of experiencing such events.
* By identification of models that perform best overall, we can develop a more accurate and reliable predictive model for earthquakes. This can help to save lives and reduce damage by enabling governments and emergency services to better prepare for and respond to earthquakes.
* From the findings we can highlight the importance of considering multiple metrics when evaluating the performance of classification models. We can acquire a more thorough knowledge of how well a model is working and make more educated judgments about which model to utilize for a given application by taking into account both accuracy and specificity (as well as other metrics).

**IMPLICATIONS:**

* The implications of the above conclusions are significant for earthquake prediction and disaster response efforts. By identifying which models perform best for each class of earthquake intensity, we can create more targeted and effective strategies for disaster preparedness and response. For example, if we know that logistic regression is better at predicting high intensity earthquakes, emergency response teams can allocate more resources and prioritize planning efforts in areas that are at higher risk of such events. This can help to reduce the impact of earthquakes, prevent loss of life and property, and ensure a more efficient response to disaster situations.
* Additionally, by identifying which models perform best overall, we can develop more accurate and reliable earthquake prediction models that can be used to inform disaster response efforts on a global scale. This can lead to more effective allocation of resources, better planning and preparedness efforts, and more efficient disaster response and recovery operations.
* More accurate prediction models can also help to improve building codes and other safety regulations, leading to better structural resilience in earthquake-prone areas and reducing the impact of earthquakes on infrastructure and property.
* By identifying which models perform best for each class of earthquake intensity, we can better understand the factors that contribute to earthquakes and potentially develop more effective prevention and mitigation strategies.
* Better earthquake prediction models can also lead to more effective international collaboration on disaster response efforts, with governments and organizations able to share information and coordinate resources to respond to earthquakes more efficiently and effectively.

**APPLICATIONS:**

* The findings of this study can be applied to improve earthquake prediction and response efforts, particularly in regions that are prone to earthquakes. The predictive models can help governments and emergency services to allocate resources and prioritize efforts more effectively, potentially saving lives and reducing the impact of earthquakes on infrastructure and property.
* The results of this study can also be used to inform future research on earthquake prediction and response, particularly regarding the factors that contribute to earthquake intensity and occurrence. This may involve further investigation into the relationship between earthquake intensity and various attributes, such as the magnitude, tsunami, date\_time, depth, and latitude / longitude, and the development of more sophisticated models to predict earthquake intensity based on these factors.
* The predictive models developed in this study can be adapted and applied to other natural disaster prediction and response efforts, such as hurricanes, tsunamis, and tornadoes. This may involve modifying the models to account for different factors and conditions that contribute to these events and applying them to different datasets and regions.
* The methodology used in this study can be adapted and applied to other data science and machine learning projects, particularly those involving classification tasks. This may involve exploring different feature selection and modeling techniques, evaluating model performance, and presenting results in a clear and concise manner.
* Future research can build on the findings of this study by incorporating additional data sources and features, such as satellite imagery and social media data, to improve the accuracy and reliability of earthquake prediction models. This may involve collaborating with experts in other fields, such as geology and seismology, to identify new data sources and refine existing models.

**CONCLUSION AND FUTURE SCOPE:**

The predictive model developed using above techniques can be utilized to inform disaster response efforts and prioritize resources in areas at higher risk of earthquakes, as follows:

* Early Warning System: The model can be integrated with an early warning system to provide alerts and notifications to the residents and authorities in the areas predicted to have higher earthquake risks.
* Evacuation Planning: The model can be used to identify the areas at higher risk of earthquakes and develop evacuation plans accordingly. The model can also help in estimating the number of people likely to be affected by the earthquake and the resources needed for their evacuation.
* Resource Allocation: The model can help in prioritizing resources such as emergency response teams, medical supplies, and rescue equipment to areas predicted to have a higher risk of earthquakes.
* Infrastructure Development: The model can be used to identify the areas with higher earthquake risks and prioritize infrastructure development efforts such as strengthening of buildings, bridges, and other critical infrastructure to reduce the potential damage and save lives.
* Insurance Planning: The model can assist insurance companies in pricing the earthquake insurance policies more accurately by predicting the areas with higher risks of earthquakes.
* Education and Awareness: The model can be used to inform and raise awareness among locals in locations where there is a higher likelihood of an earthquake about the necessary precautions and emergency plans to prevent harm and save lives.