**PREDICTIVE ANALYSIS OF ROAD ACCIDENT PREDICTION AND RESPONSE**



Coapps–A Software Company

**REPORT**

**Project partner’s**:

1.Azhagapan T (Team Lead). 2.Sowmya K 3.Kaviya Sri K

**College:**

 Kalaignarkarunanidhi Institution

of Technology, Coimbatore

**Table of Content**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Topic** | **Pages** |
| 1. | Abstract | 3 |
| 2. | Software and model  Required | 5-8 |
| 3. | Architecture | 10 |
| 4. | Steps in building model | 11-22 |
| 5. | Code | 23 |
| 6. | Output | 24-28 |
| 7. | Conclusion | 29 |

**ABSTRACT**

This study introduces a predictive analysis framework employing the Regression Forest classifier algorithm to anticipate road accident severity and propose preventive measures. Key input features include light conditions, junction type, road surface, weather, and collision type. Leveraging historical accident data, the framework aims to construct a robust predictive model for accurate severity prediction. By utilizing Regression Forest, capable of handling complex relationships, the model offers insights into high-risk areas and conditions. This enables stakeholders to implement targeted interventions like improved signage and traffic management. Moreover, the framework supports the development of real-time prediction systems for timely alerts to drivers and authorities, thereby enhancing road safety measures. This research emphasizes the potential of predictive analytics in reducing accident rates and saving lives, underscoring its significance in road safety management.

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table no.** | **Table Name** | **Page No.** |
| 1. | Performance Accuracy | 24 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Figure Name** | **Page No.** |
| 1. | Visualization of Accident severity | 25 |
| 2. | Visualization of cause of accident | 26 |
| 3. | Accident severity | 26 |
| 4. | Vehicle movement | 27 |
| 5. | Weather condition | 27 |
| 6. | Cause of accident vs severity. | 28 |
| 7. | Road surface vs severity | 28 |
| 8. | Type of collision vs severity | 29 |
| 9. | Weather condition vs severity | 29 |

**SOFTWARE REQUIRED**

The software required for model building are mentioned below:

1. Jupyter Notebook
2. Vs code
3. Git-hub
4. Stream-lit

**Jupyter Notebook:**

Jupyter Notebook, an open-source web tool, enables users to generate and distribute documents that merge live code, equations, visualizations, and textual explanations. While supporting various programming languages, it is especially favored among Python users. With its interactive platform, Jupyter Notebook permits code execution within designated cells, promptly displaying outcomes within the document.

**VS code:**

Visual Studio Code (VS Code), a widely-used code editor developed by Microsoft, stands out for its lightweight design and powerful features, making it a top choice for developers working with diverse programming languages and platforms. With a rich ecosystem of extensions, users can easily enhance functionality, including language support and debugging tools. Intelligent code editing features like IntelliSense provide context-aware code completions, while built-in Git commands streamline version control tasks

**Git-Hub:**

GitHub is like a shared space where people work on projects together, especially for writing and managing computer code. It keeps track of all the changes made to the project files over time, so everyone can see what's been done and contribute their own ideas without stepping on each other's toes.

**Stream-lit:**

Stream-lit is a tool that helps you turn your Python code into interactive web apps. You write your code like normal, but with a few extra commands, you can add buttons, sliders, and charts to your app. When you run your code,

Streamlit automatically creates a website that people can use in their web browsers. They can play with the buttons and sliders, and see the results change in real-time. Once you are done, you can easily share your app with others by sending them a link.

**MODEL USED**

Model is like a simplified version of something real. It helps us understand how things work by representing them in a way that is easier to study.

1. Pandas.
2. NumPy.
3. SKlearn.
4. Seaborn.
5. Confusion. Matrix.

**PANDAS:**

Pandas, a Python library, simplifies working with structured data like tables or spreadsheets. Once imported, Pandas provides two primary data structures:

* + 1. One-dimensional array, and Data Frame.
    2. A two-dimensional tabular data structure similar to a spreadsheet.

Data can be loaded into Pandas Data Frames from various file formats such as CSV, Excel, or databases. With Pandas, you can perform numerous data manipulation tasks like selecting, filtering, adding, removing, or renaming columns, handling missing data, grouping, sorting, and merging data

**NUMPY:**

NumPy is a crucial Python library for handling numerical computations efficiently. To utilize NumPy, you begin by importing it into your Python environment using the common convention import numpy as np. Once integrated, you can create arrays with the numpy.array() function, which can be

* + - * 1. One-dimensional,
        2. Two-dimensional
        3. Even multidimensional.

**SKLEARN:**

With scikit-learn, users can access a comprehensive suite of machine learning algorithms encompassing classification, regression, clustering, dimensionality reduction, and model selection. Scikit-learn, often abbreviated as Sklearn, is a Python library widely used for machine learning tasks. It provides an array of efficient tools for data mining and analysis, leveraging underlying scientific computing libraries like NumPy, SciPy, and matplotlib.

**SEABORN:**

Seaborn is a Python library for data visualization that operates on top of matplotlib, offering an elevated interface for crafting visually appealing and informative statistical graphics. Designed to seamlessly integrate with panda’s data structures and the broader Python scientific computing ecosystem, Seaborn streamlines the creation of intricate visualizations. It is used to represent different plot types such as

* + - 1. Statistical
      2. Categorical
      3. Distribution
      4. Regression
      5. Matrix plots.

**CONFUSION MATRIX:**

A confusion matrix is a tabular representation used to evaluate the performance of a classification model. It summarizes the model's predictions by comparing them against the actual labels of the data. In binary classification tasks, it breaks down the predictions into categories such as true positives, true negatives, false positives, and false negatives. These metrics help assess the model's accuracy, precision, recall, and F1-score, providing insights into its effectiveness in correctly classifying instances

**ARCHITRCTURE**

DATA CLEANING

VISUALIZATION

ACCURACY CHECKING

STREAMLIT

MODEL SELECTION

DATA PREPROCESSING

**EXPLANATION:**

Let us see about the process and what are the steps that are followed in-order to make a machine learning model by using suitable algorithm and model step-by-step.

1. **Data cleaning**

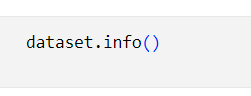
Data cleaning, also known as data preprocessing or data cleansing, is the essential task of identifying and rectifying errors, inconsistencies, and inaccuracies within a dataset to enhance its quality and usability for analysis or modeling purposes.

1. **Eliminating Duplicates:** Identifying and removing duplicate entries from the dataset to ensure that each observation accurately represents the underlying population.
2. **Correcting Errors:** Detecting and rectifying errors or inconsistencies in the data, such as typos, incorrect values, or formatting discrepancies.
3. **Managing Categorical Data:** Converting categorical variables into suitable formats for analysis, such as one-hot encoding or label encoding.

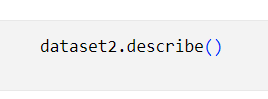
Functions used in Data Cleaning Process:

1. **Info()**

The info() function in pandas gives you a quick summary of your dataset. It tells you how many rows and columns you have, along with the names of the columns and the data types they contain. It also shows how many non-empty values are in each column and gives you an estimate of how much memory your dataset is using.

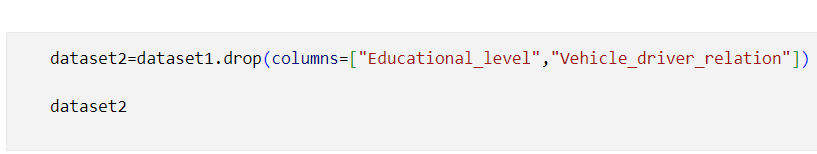
****

1. **describe()**

****  The describe() function in pandas is like a cheat sheet for understanding your dataset. When you apply it to a Data Frame, it gives you a summary of the numerical columns, including statistics like count, mean, standard deviation, minimum, and maximum values. It helps you quickly grasp the distribution and range of values in your data.

1. **Droping unwanted columns:**
   * 1. Droping unwanted columns in a dataset involves removing specific columns that are not necessary or relevant for the analysis or task at hand.
     2. This process helps streamline the dataset, making it more focused and efficient for further processing.
     3. When dropping columns, you identify the columns that are not needed and remove them from the dataset entirely.

* + 1. This can be done using functions or methods provided by data manipulation libraries like pandas in Python.

****

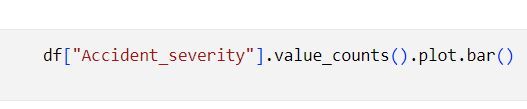
The above code will drop the columns educational level and vehicle driver relation because these two columns are not necessary for model building and prediction.

**B**. **VISUALIZATION:**

Visualization is the process of representing data or information visually through graphical or pictorial means. It involves using charts, graphs, maps, and other visual elements to communicate patterns, trends, and relationships in the data. Visualization helps make complex data more accessible and understandable, enabling individuals to gain insights, identify patterns, and make informed decisions more effectively.

1. **Visualization for Accident severity:**

Accident severity refers to the level of harm caused by an accident, typically measured by factors such as the extent of injuries, damage, and loss resulting from the incident. It involves categorizing accidents based on the severity of their outcomes, from minor incidents with limited consequences to major events causing significant harm or loss**.**



1. **Cause of Accident:**

The causes of accidents can vary widely depending on the context, but they often result from a combination of factors such as human error, environmental conditions, equipment failure, and organizational factors.



|  |  |
| --- | --- |
| 1. Changing lane to the left |  |
| 1. Changing lane to the right |  |
| 1. Driving at high speed |  |
| 1. Driving carelessly |  |
| 1. Driving to the left |  |
| 1. Driving under the influence of drugs |  |
| 1. Drunk driving 2. Overtaking 3. Overspeed   **10.**Overloading |  |

The cause of the accident is:

1. **DATA PREPROCESSING:**

This essential step comprises various operations such as cleaning, transforming, and organizing data to ensure its accuracy, consistency, and relevance for subsequent analytical tasks. By addressing issues like missing values, duplicates, and inconsistent formats, data preprocessing enhances the quality of the dataset and facilitates more accurate and meaningful insights during analysis or modeling endeavors.

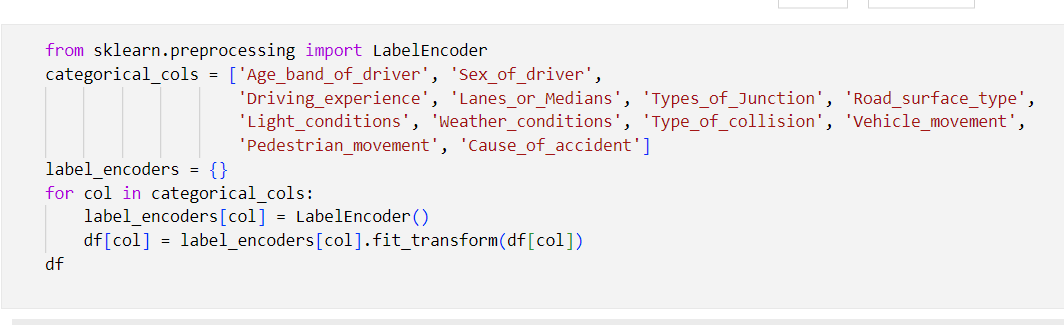
Overall, data preprocessing lays the foundation for effective data analysis by refining raw data into a suitable format that can be readily utilized for deriving insights and making informed decision.

In this project we use LABEL ENCODING method.

**LABEL ENCODING:**

Label encoding is a method used in data preprocessing to convert **categorical variables** into **numerical format**. It assigns a unique numerical value to each category or label in the variable.

While label encoding is effective for categorical variables with ordinal relationships, caution is needed with nominal variables to avoid introducing unintended patterns. Despite its simplicity, label encoding may not always be suitable for all categorical variables, and alternative encoding techniques like one-hot encoding may be more appropriate in certain scenarios.

****

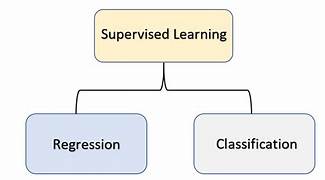
1. **Categorical variable** – text format.
2. **Numerical format** – Integer.
3. **MODEL SELECTION:**

In machine learning, various models serve different purposes and address diverse problems. Here is a breakdown of common types:

**Supervised Learning Models:**

Supervised learning is a branch of machine learning where the model is trained on a labeled dataset. This means that each input data point is paired with a corresponding target or output label. The objective is for the model to learn a relationship between the input features and the output labels so that it can make accurate predictions on new, unseen data.

There are two primary types of tasks in supervised learning



**1. Regression**: This involves predicting a continuous target variable. Examples include forecasting house prices based on features like square footage and location.

**2. Classification:** Here, the goal is to assign data points to discrete classes or categories. For instance, classifying emails as spam or not spam based on their content.

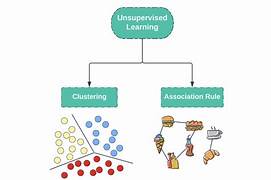
Supervised learning finds applications across various domains, including

* + - 1. Finance,
      2. Healthcare,
      3. Image recognition, and
      4. Recommendation systems.
      5. Machine learning

**Unsupervised Learning Models:**

Unsupervised learning is a branch of machine learning where models are trained on unlabeled datasets, meaning there are no predetermined output labels provided. The objective is to discover inherent patterns, structures, or relationships within the data without explicit guidance. In unsupervised learning, models aim to unveil hidden structures or intrinsic properties in the input data.

Key tasks in unsupervised learning include:



**1.Clustering**: Grouping similar data points into clusters or segments based on their inherent similarities or patterns. Popular clustering algorithms include k-means, hierarchical clustering.

**2.Prediction:** In unsupervised learning, "prediction" refers to making estimations or inferences about the underlying structure or patterns within the data. This includes tasks such as assigning new

data points to existing clusters, detecting anomalies or outliers, and predicting reduced-dimensional representations of data points. Unlike supervised learning, where prediction involves forecasting specific outcomes, in unsupervised learning, it involves understanding the inherent structure of the data without labeled examples.

Application:

* + 1. Customer segmentation,
    2. Anomaly detection,
    3. Data compression,
    4. Exploratory data analysis.

**RANDOM FOREST CLASSIFIER:**

A Random Forest Classifier is a widely employed machine learning algorithm, particularly in classification tasks. It falls under the ensemble learning category, known for its ability to combine multiple decision trees during training.

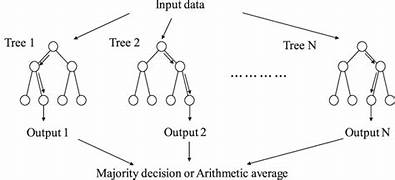
**Here' s how it works:** Instead of relying on a single decision tree, the Random Forest constructs numerous trees using different subsets of the training data and features. This diversity helps enhance the model's robustness and accuracy.

During prediction, each decision tree independently assigns a class to the input data. The final prediction is determined by aggregating the individual tree predictions through a majority voting mechanism.

Random Forest Classifier offers several advantages, including high accuracy, resistance to overfitting, versatility in handling different types of data, and robustness against noisy datasets.

**APPLICATION:**

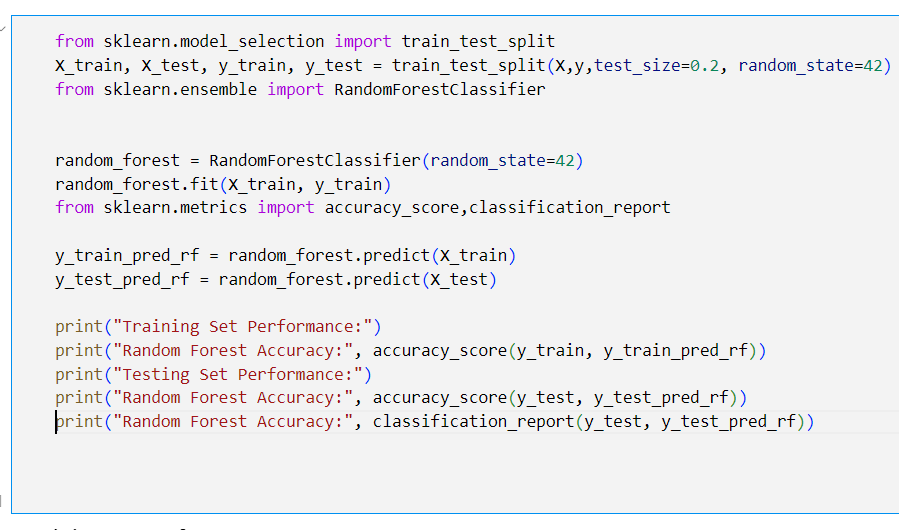
* + 1. Spam detection
    2. Medical diagnosis
    3. Sentiment analysis
    4. Customer churn prediction.

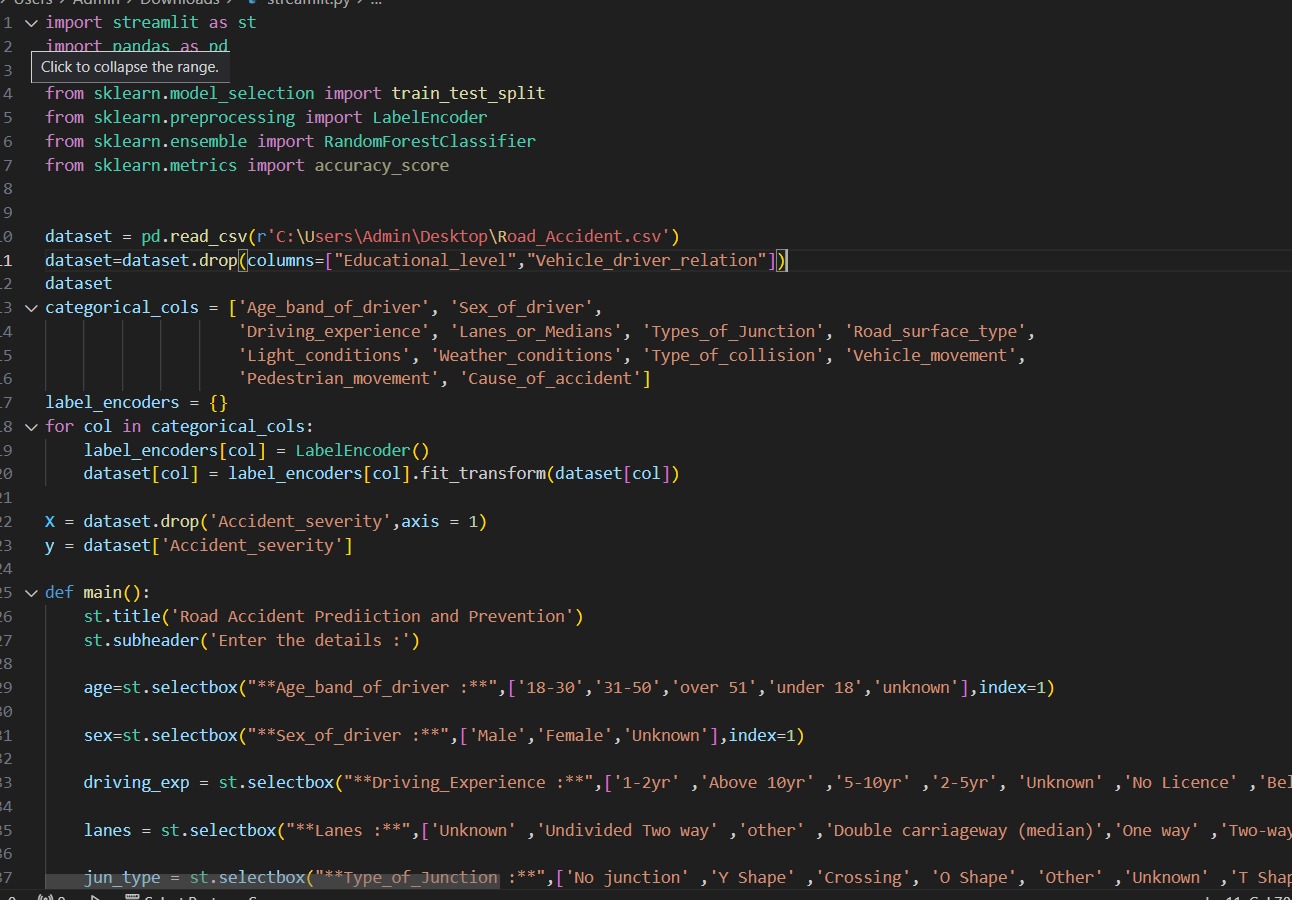


**Representation of Random Forest classifier**

1. **Source code:**

The source code for building model using random forest classifier is given below. And here we use random forest classifier model to predict the output as it gives good accuracy for our model building.



1. **STREAMLIT CODE:**

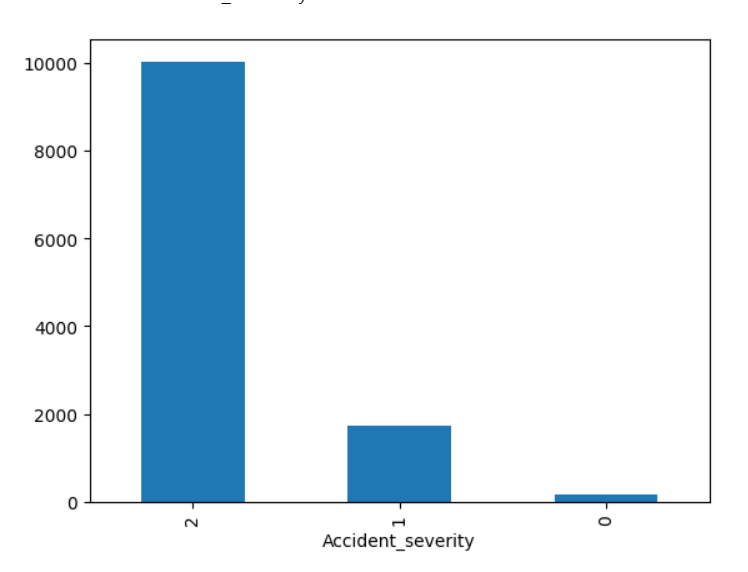
****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **Precision** | **Recall** | **f1-score** | **support** |
| 0 | 0.40 | 0.06 | 0.10 | 37 |
| 1 | 0.16 | 0.03 | 0.04 | 345 |
| 2 | 0.84 | 0.98 | 0.90 | 2007 |
| Accuracy | - | - | 0.83 | 2386 |
| Macro average | 0.47 | 0.35 | 0.35 | 2386 |
| Weighted average | 0.74 | 0.83 | 0.77 | 2386 |

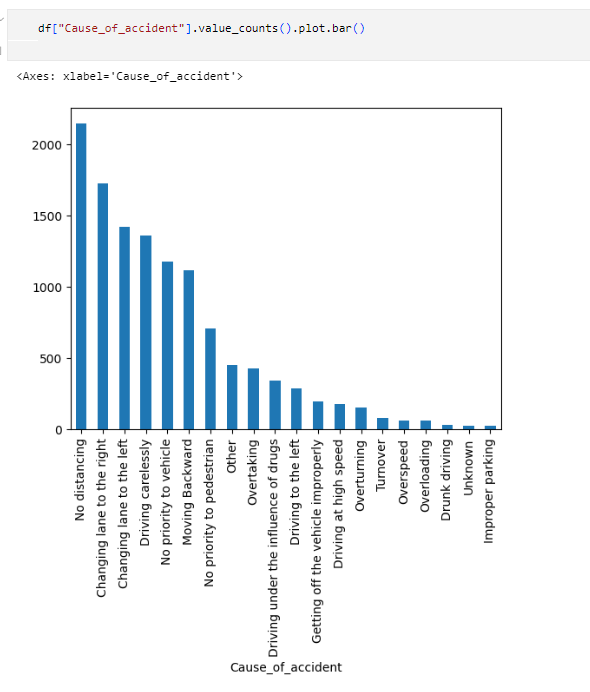
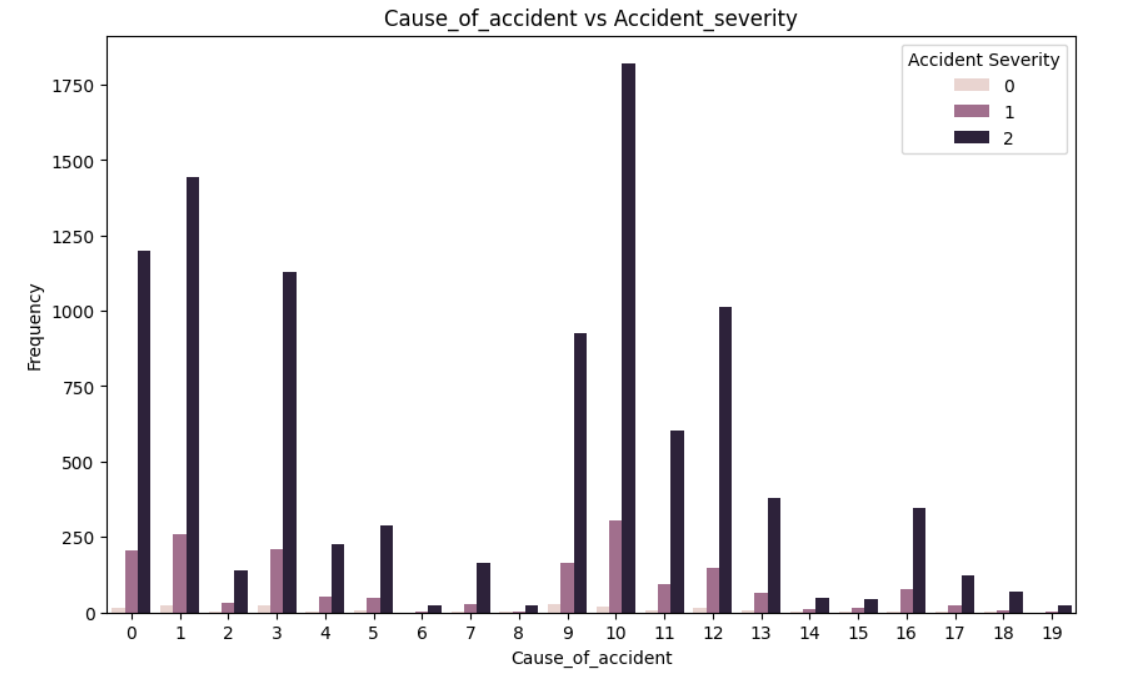
**ANALYSIS OF RANDOM FOREST CLASSIFIER:**

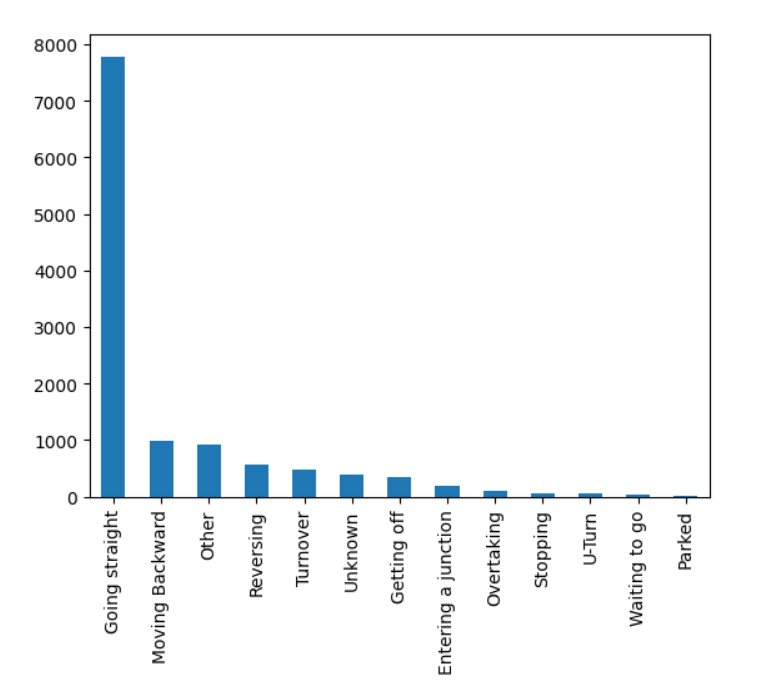
**Training Set Performance:** Random Forest Accuracy- **0.9821802935010482[98%]Testing Set Performance:** Random Forest Accuracy- **0.8252305113160101[82%]**

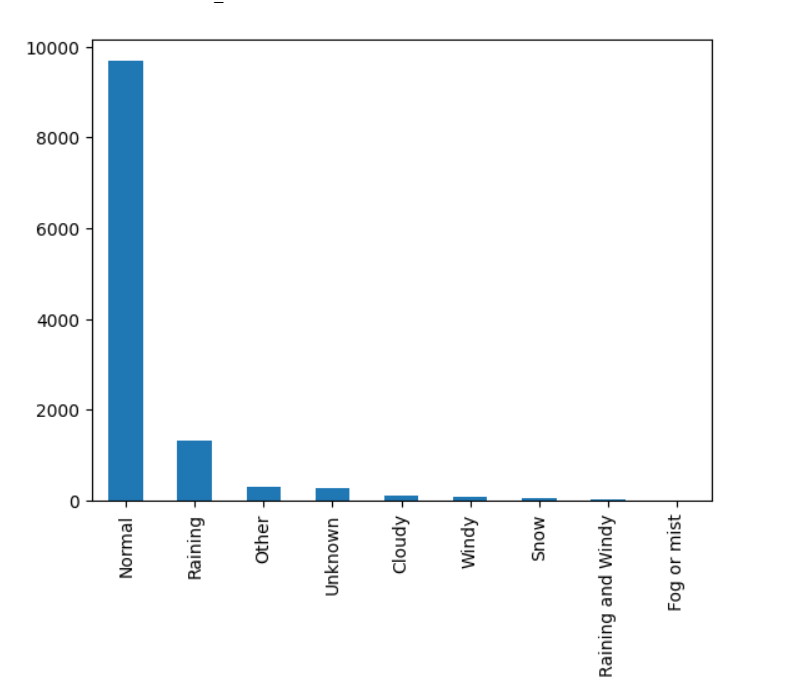
1. **OUTPUT:**

****

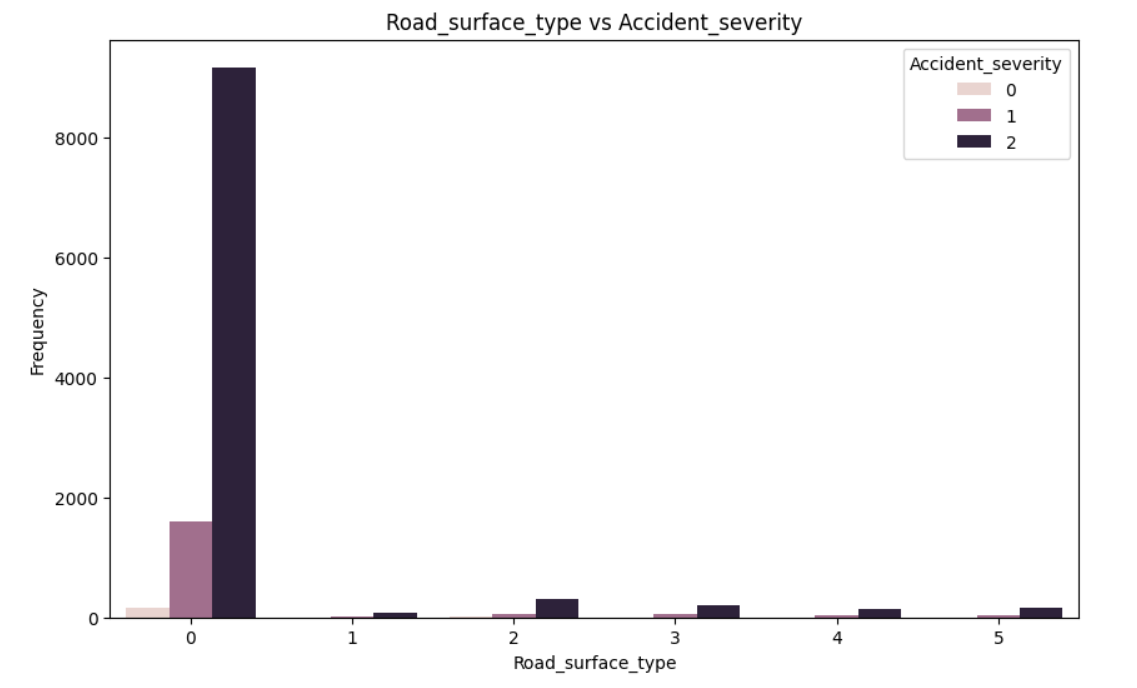
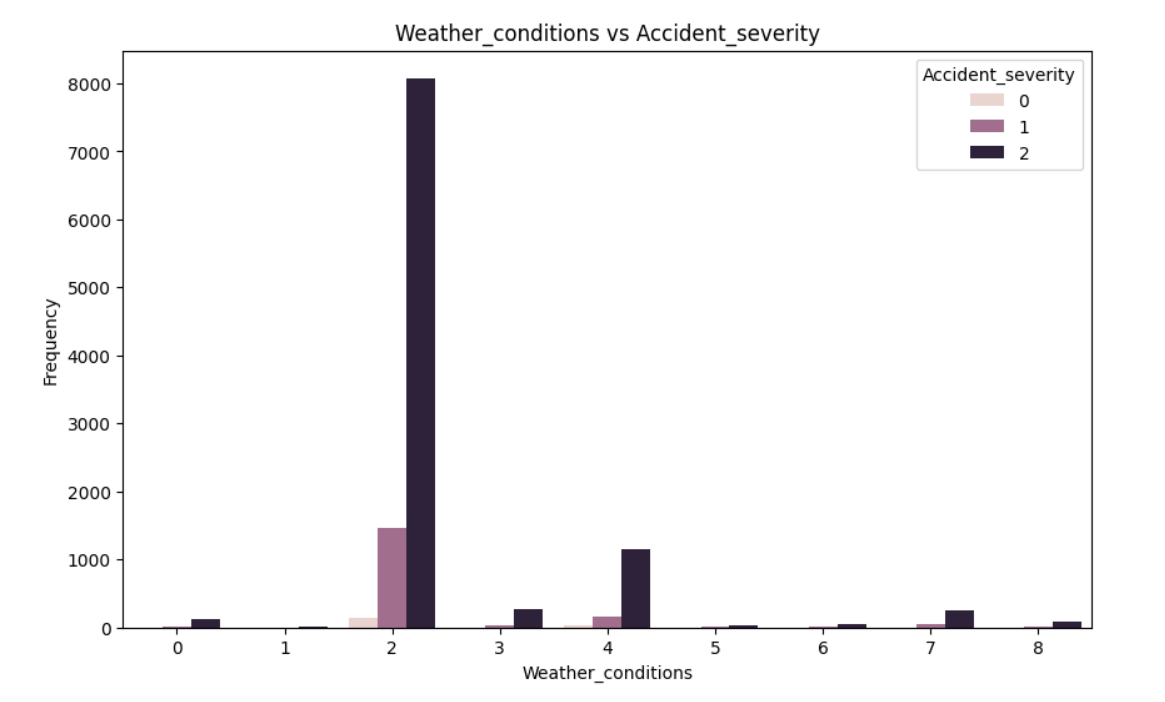
1. Visualization of Accident severity

 b. Visualization of cause of accident  c. accident severity

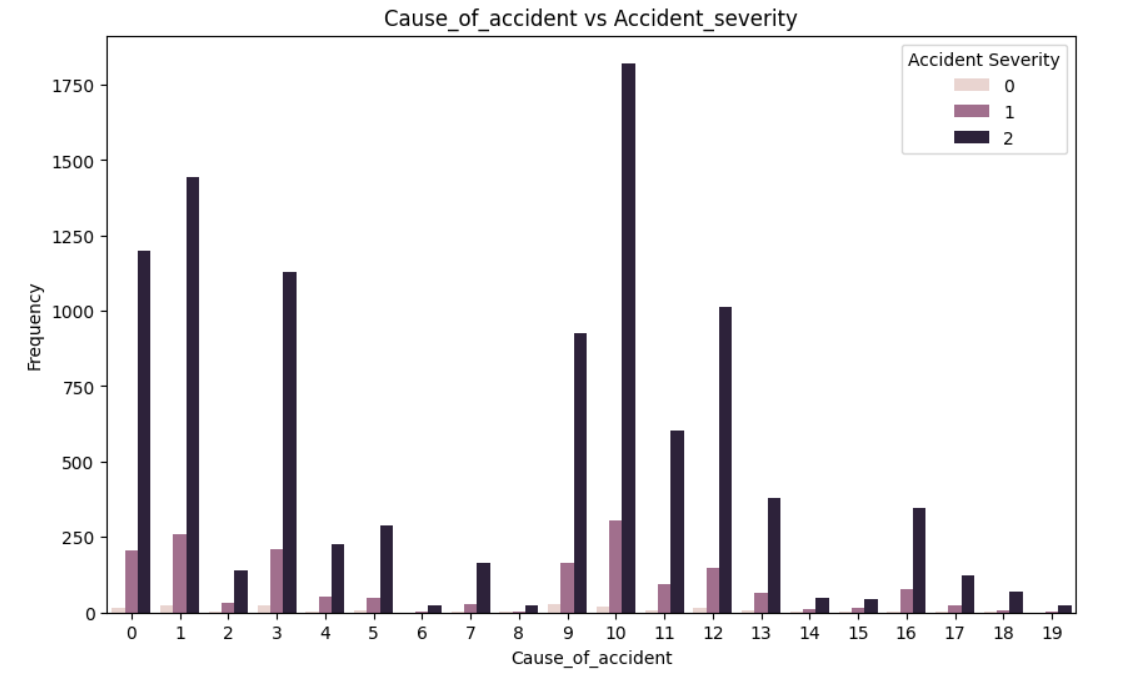
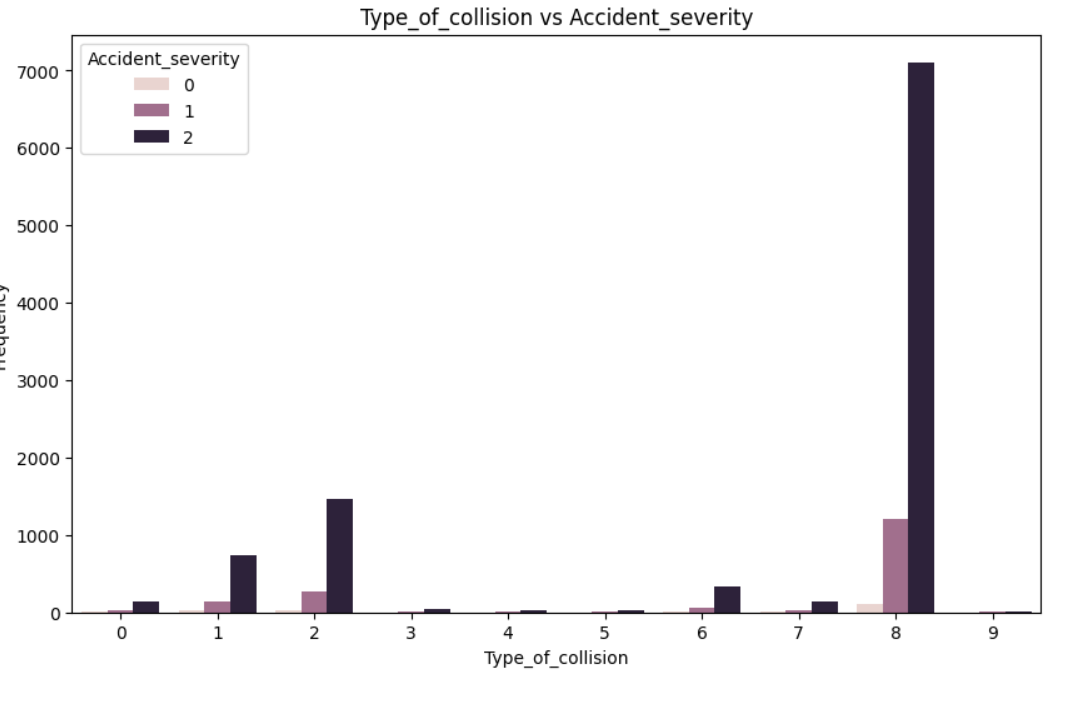
 d. Vehicle movement



e. Weather condition

 f. Road surface vs severity

g. Weather condition vs severity

 h. cause of accident vs severity

i. Type of collision vs severity

**G**. **CONCLUSION:**

The predictive analysis conducted using a random forest classifier model for road accident prevention and response yields several significant conclusions. Firstly, the model demonstrates commendable accuracy in predicting accidents by utilizing diverse factors such as weather conditions, road infrastructure, traffic volume, and historical accident data. Through feature importance analysis, key risk factors contributing to accidents are identified, including adverse weather, poor road conditions, high traffic, and specific accident-prone locations. This understanding forms the basis for an effective early warning system, enabling authorities to proactively allocate resources and respond promptly to high-risk situations. Continuous refinement and validation of the predictive model are emphasized to maintain its accuracy and relevance in supporting decision-making processes related to road safety. Overall, the analysis underscores the importance of data-driven approaches in mitigating road accidents and enhancing emergency response efforts, ultimately contributing to saving lives