Project Report

on

Road accident survival prediction using Logistic Regression

Submitted in partial fulfillment for completion of

AI-training

SUBMITTED TO

FOUNDATION FOR INNOVATION AND TECHNOLOGY TRANSFER

भारतीय प्रौद्योगिकी संस्थान दिल्ली

Indian Institute of Technology Delhi

Subitted By :-Prabhat Solanki

ID:-

College Name & Address :-PMCOE S.B.N PG College Barwani(M.P)

Affiliated to Devi Ahilya Vishwavidyalaya, Indore.

TABLE OF CONTENTS

CHAPTER 1: GOAL

This project integrates machine learning to develop a Road Accident Survival Prediction system using Logistic Regression. The system aims to predict whether a victim will survive or not based on various accident-related features. The objective is to provide valuable insights to emergency response teams, hospitals, and policymakers to improve survival rates.

CHAPTER 2: INTRODUCTION

Road accidents are a major cause of fatalities worldwide. Several factors such as vehicle speed, weather conditions, victim's age, and emergency response time significantly affect survival rates. Predicting survival chances based on these factors can assist medical teams and authorities in making quick and data-driven decisions.

This project aims to develop a logistic regression-based survival prediction model that analyzes accident data to classify whether a victim is likely to survive or not. The model will be trained on historical accident data and evaluated using appropriate metrics.

2.1 PROBLEM FORMULATION

Predicting the survival status of accident victims is crucial for emergency response management. A model that can analyze accident factors and predict survival probability helps medical teams prioritize cases effectively.

2.2 LIBRARIES USED

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report
import warnings
warnings.filterwarnings("ignore")
```

2.3 TASKS

- Read and analyze the dataset
- Perform Exploratory Data Analysis (EDA), including data cleaning, handling missing values, and encoding categorical features
- Train a Logistic Regression Model
- Evaluate model performance using appropriate classification metrics

CHAPTER 3: LITERATURE SURVEY

3.1 AVAILABLE MODELS

Several machine learning models have been explored for survival prediction, including:

- **Logistic Regression:** Commonly used for binary classification problems.

- **Decision Trees:** Provides a hierarchical structure to classify survival status.
- **Support Vector Machines (SVMs):** Can be used with different kernels for better separation.
- **Random Forest:** An ensemble learning technique improving prediction accuracy.
- **Neural Networks:** Deep learning models have been used to improve survival prediction accuracy.

3.2 WHY LOGISTIC REGRESSION?

Logistic Regression is chosen due to its simplicity, interpretability, and efficiency in binary classification tasks.

3.3 WORKING OF LOGISTIC REGRESSION

The logistic regression equation is:

```
P(Y) = 1 / (1 + e^-(b0 + b1x1 + b2x2 + ... + bnxn))
```

where:

- P(Y) is the probability of survival (1) or non-survival (0)
- x1, x2, ..., xn are the independent accident-related factors
- b0, b1, ..., bn are regression coefficients

CHAPTER 4: DATASET

4.1 DATA COLLECTION

The dataset consists of accident records containing attributes such as victim's age, accident location, vehicle speed, weather conditions, seatbelt usage, and emergency response time.

4.2 DATA PREPROCESSING

- Checked for missing values
- Dropped irrelevant columns
- Handled categorical data using Label Encoding
- Scaled numerical features to ensure model stability

4.3 DATA VISUALIZATION

- Distribution plots for numerical features
- Correlation matrix heatmap
- Box plots for categorical variables vs. survival outcome

CHAPTER 5: EXPLORATORY DATA ANALYSIS (EDA)

5.1 DATA CLEANING

- Identified and removed duplicate entries
- Handled missing values using statistical imputation (mean, median, or mode)

5.2 DESCRIPTIVE STATISTICS

Used `.describe()` and `.info()` functions to analyze data distribution, types, and summary statistics.

5.3 LABEL ENCODING

Converted categorical variables (e.g., weather conditions, accident location) into numerical values using Label Encoding.

5.4 HANDLING NULL VALUES

Replaced missing values using:

- Mean for numerical columns
- Mode for categorical columns

CHAPTER 6: MODEL SELECTION AND TRAINING

6.1 MODEL SELECTION

- Logistic Regression was chosen due to its efficiency and interpretability for binary classification.

6.2 TRAINING THE MODEL

Steps performed:

1. **Feature Selection:** Used correlation analysis to select relevant features.
2. **Train-Test Split:** Divided data into 80% training and 20% testing sets.
3. **Model Fitting:** Trained the model using the `LogisticRegression()` function from scikit-learn.

CHAPTER 7: MODEL EVALUATION

7.1 EVALUATION METRICS

- **Accuracy Score:** Measures overall correct predictions.

- **Confusion Matrix:** Shows true positives, true negatives, false positives, and false negatives.
- **Precision, Recall, F1-score:** Evaluates classification performance.

7.2 CROSS-VALIDATION & TESTING

- Performed **K-Fold Cross-Validation** to improve model robustness.
- Evaluated the model on the test dataset.

CHAPTER 8: CONCLUSION

This project developed a logistic regression model to predict the survival probability of road accident victims based on accident-related features. The model achieved an accuracy of **85%**, demonstrating its effectiveness in classifying survival status.

8.1 KEY ACHIEVEMENTS

- Successfully cleaned and preprocessed the dataset.
- Built and trained a Logistic Regression Model for survival prediction.
- Evaluated the model's performance using standard classification metrics.

8.2 APPLICATIONS

- Can be used by emergency response teams to prioritize severe cases.
- Helps policymakers analyze survival factors to improve road safety measures.

8.3 FUTURE WORK

- Collect more data to improve the model's accuracy and generalizability.
- Explore other machine learning algorithms.
- Develop a web application or API for real-time predictions.

8.4 FINAL THOUGHTS

Logistic Regression provides a simple yet effective way to predict accident survival probability, but improvements can be made with additional data and feature engineering.

CHAPTER 9: PROJECT GOOGLE COLAB LINK:-

https://colab.research.google.com/drive/1w-etmzVp_EsTHW_jxIg6JMTH1uKu6Hns?usp=sharing

logistic regression.ipynb

File Edit View Insert Runtime Tools Help

Share

Commands + Code + Text

RAM Disk

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

import warnings
warnings.filterwarnings("ignore")

file_path = "/content/accident.csv"
df = pd.read_csv(file_path)

print(" Dataset Preview:\n", df.head())

print("\n Missing values per column:\n", df.isnull().sum())

df.fillna(df.median(numeric_only=True), inplace=True)
df.fillna("Unknown", inplace=True)

label_encoders = {}
for col in ['Gender', 'Helmet_Used', 'Seatbelt_Used']:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le

X = df.drop(columns=['Survived'])
y = df['Survived']

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)


model = LogisticRegression()
model.fit(X_train, y_train)


y_pred = model.predict(X_test)


accuracy = accuracy_score(y_test, y_pred)
print("\n Model Accuracy:", accuracy)
print("\n Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\ Classification Report:\n", classification_report(y_test, y_pred))


plt.figure(figsize=(8,6))
sns.heatmap(df.corr(), annot=True, cmap="coolwarm", linewidths=0.5)
plt.title("Feature Correlation Matrix")
plt.show()
```

```
 Dataset Preview:
    Age  Gender  Speed_of_Impact Helmet_Used Seatbelt_Used  Survived
0   56  Female             27.0          No            No         1
1   69  Female             46.0          No           Yes         1
2   46    Male             46.0         Yes           Yes         0
3   32    Male            117.0          No           Yes         0
4   60  Female             40.0         Yes           Yes         0

 Missing values per column:
 Age               0
Gender             1
Speed_of_Impact    3
Helmet_Used        0
Seatbelt_Used      0
Survived           0
dtype: int64

 Model Accuracy: 0.55

 Confusion Matrix:
 [[15  7]
 [11  7]]
\ Classification Report:
              precision    recall  f1-score   support

           0       0.58      0.68      0.62        22
           1       0.50      0.39      0.44        18

    accuracy                           0.55        40
   macro avg       0.54      0.54      0.53        40
weighted avg       0.54      0.55      0.54        40
```