Gladwellchebelyon /
Traffic-Crashes

Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights · Settings

Chicago Traffic/ Car Crashes

GPL-2.0 license

0 stars · 0 forks · 1 watching · 1 Branch · 0 Tags · Activity

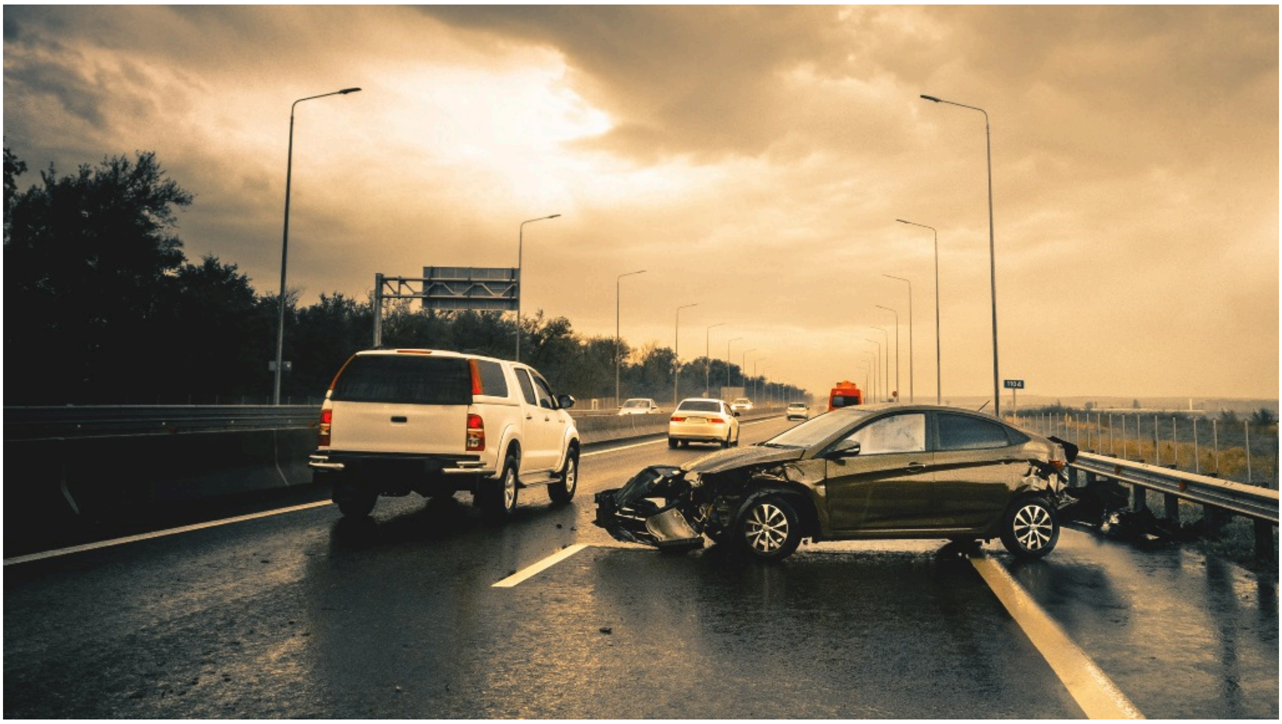
Public repository

1 Branch · 0 Tags · Go to file · t · Go to file · + · Add file · Code · ...

Gladwellchebelyon Update README.md · a0f86e0 · 4 minutes ago

.gitignore	Initial commit	last week
Gladwellchebelyon_Traffic-Cr...	github.pdf	5 hours ago
LICENSE	Initial commit	last week
Predictive Analysis of Traffic ...	pptx presentation	6 hours ago
Presentation.pdf	Presentation.pdf	6 hours ago
README.md	Update README.md	4 minutes ago
Traffic_Crashes.ipynb	Created using Colab	7 hours ago
Traffic_Crashes.pdf	notebook.pdf	5 hours ago

README · GPL-2.0 license



Traffic Crash Analysis: Predicting Primary Contributory Cause

Table of Contents:

Project Overview:

This project aims to analyze traffic crash data from the City of Chicago to predict the primary contributory cause of an accident. The primary goal is to assist organizations such as Vehicle Safety Boards or city administrations in identifying patterns and reducing traffic accidents.

Dataset

Source: The dataset is obtained from the City of Chicago's Traffic Crashes dataset.

Description: The dataset includes information about vehicle crashes, such as crash location, time, weather conditions, and the primary and secondary contributory causes.

Size: Approximately X million records.

Features:

CRASH_DATE

TRAFFIC_CONTROL_DEVICE

WEATHER_CONDITION

LIGHTING_CONDITION

ROADWAY_SURFACE_COND

PRIM_CONTRIBUTORY_CAUSE

MOST_SEVERE_INJURY

Additional engineered features like Is_Weekend, Speed_Weather_Interaction, etc.

Installation

To run this project locally, follow these steps:

Clone the repository:

```
'git clone https://github.com/yourusername/Traffic-Crash-Analysis.git,'
```

Navigate to the project directory:

```
'cd Traffic-Crash-Analysis'
```

Create a virtual environment:

```
 'python -m venv venv'
```

Activate the virtual environment:

```
'venv\Scripts\activate'
```

On macOS/Linux:

```
'source venv/bin/activate'
```

Install the required packages:

```
'pip install -r requirements.txt'
```

Usage

Preprocessing and Feature Engineering:

Run preprocessing.ipynb to clean and prepare the data.

Key steps include handling missing values, feature engineering, and encoding categorical variables. Model Training:

Run model_training.ipynb to train various classification models.

Includes Logistic Regression, Ridge Classifier, Decision Trees, Random Forest, and Gradient Boosting. Model Tuning and Evaluation:

Use model_tuning.ipynb to perform hyperparameter tuning using GridSearchCV and RandomizedSearchCV.

Evaluate models using metrics such as accuracy, precision, recall, and F1-score.

Final Testing and Results:

Run final_testing.ipynb to test the best-performing model on the test set.

Generate performance reports and confusion matrices. Visualization:

Use visualization.ipynb for exploratory data analysis (EDA) and to create plots like heatmaps, pair plots, and bar charts comparing model performances.

Modeling Process

Baseline Model: Logistic Regression was initially used as the baseline.

Feature Selection: Features were selected based on correlation and domain knowledge.

Model Comparison: Multiple models were trained, including Logistic Regression, Ridge Classifier, Random Forest, and Gradient Boosting.

Hyperparameter Tuning: RandomizedSearchCV was used for tuning hyperparameters to improve model performance.

Final Model: Gradient Boosting was selected as the best-performing model based on accuracy and other evaluation metrics.

Results

Best Model: Gradient Boosting

Accuracy: 1.00%

Key Features: The most important features contributing to the model's predictions were Weather_Condition_Mode, Speed_Weather_Interaction, and Is_Weekend.

Insights and Recommendations

Traffic Management: The model suggests that weather conditions and speed limits are significant contributors to crashes.

Implementing stricter speed regulations during adverse weather could reduce accidents.

Policy Changes: Focus on improving road conditions and traffic control devices in areas with high accident rates.

Further Analysis: Recommend analyzing the impact of time of day and specific locations (using latitude and longitude) on crash rates for targeted interventions.

Contributing

Contributions are welcome! Please create a pull request or open an issue for any suggestions or improvements.

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

- Jupyter Notebook 100.0%