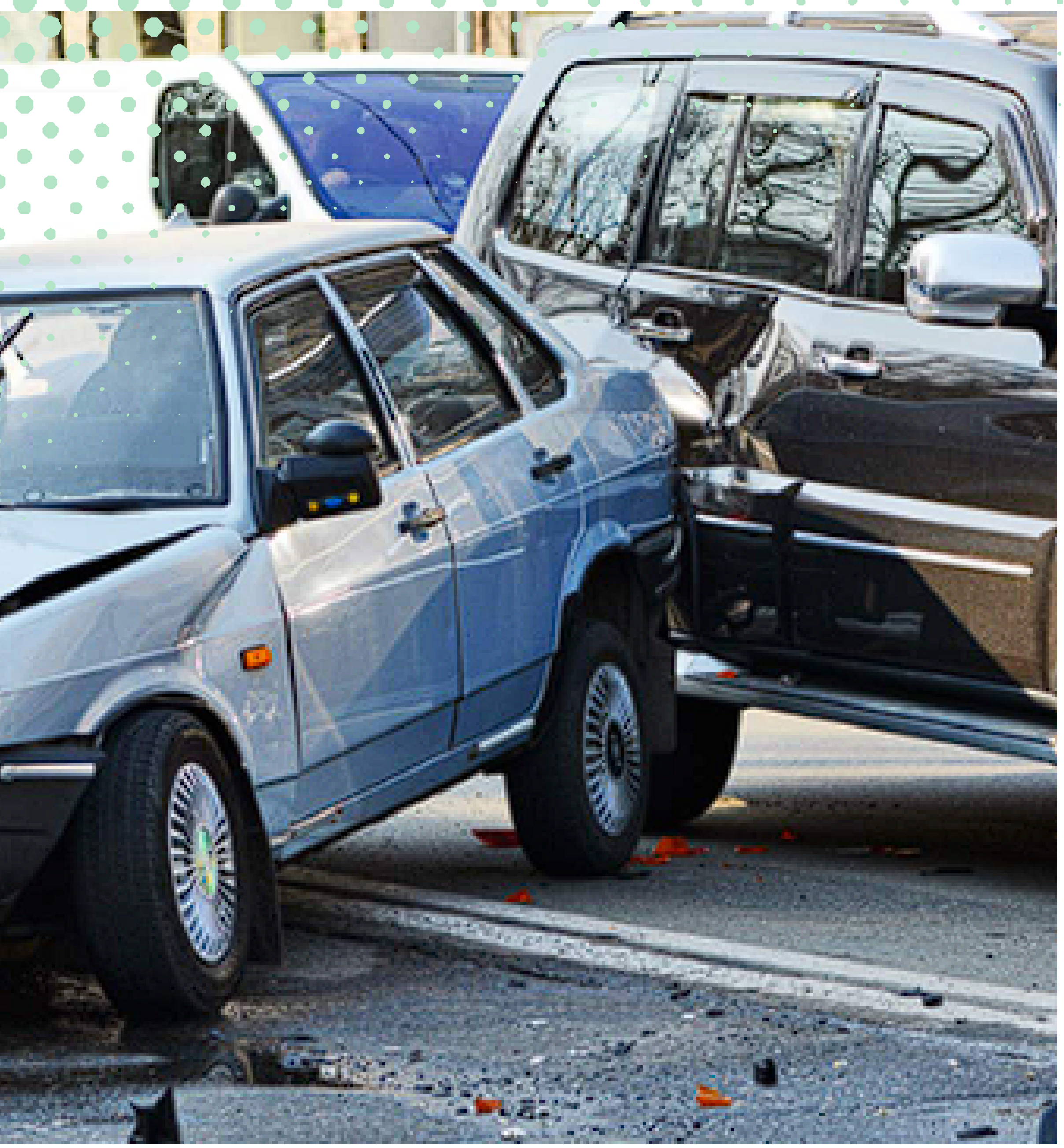
Machine Learning: Car Accident Predictions

Presented by:
Ikran Askar
Molly Campbell
Matthew Hernandez

Agenda

- Project overview
- Description of data and purpose
- Data preprocessing
- Machine learning models
- Visualizations
- Lessons learned



Back to Agenda

Project Overview

In this project we tried to classify the severity of road accidents based off of different factors (sex, weather, time of day,etc). The data was collected from vehicle collisions in Ethiopia ranging from 2017-2020.

Road accidents are one of the major causes of unnatural deaths around the world. We used Machine Learning with the objecting of measuring how influential certain factors can be in accidents taking place in the hopes of reducing the amount and severity in the future.

Back to Agenda

Discription of Data

Dataset:

- 32 columns and 12316 rows.
- Accident Severity broken into 3 categories: slight injury, serious injury fatal injury.
- Other factors considered:
  Road surface, Time of Day,
  Weather, Sex of Diver, Age of Driver .

Back to Agenda

Target variable

Accident severity

- Minor
- Severe
- Fatal

Back to Agenda

Data Preprocessing steps

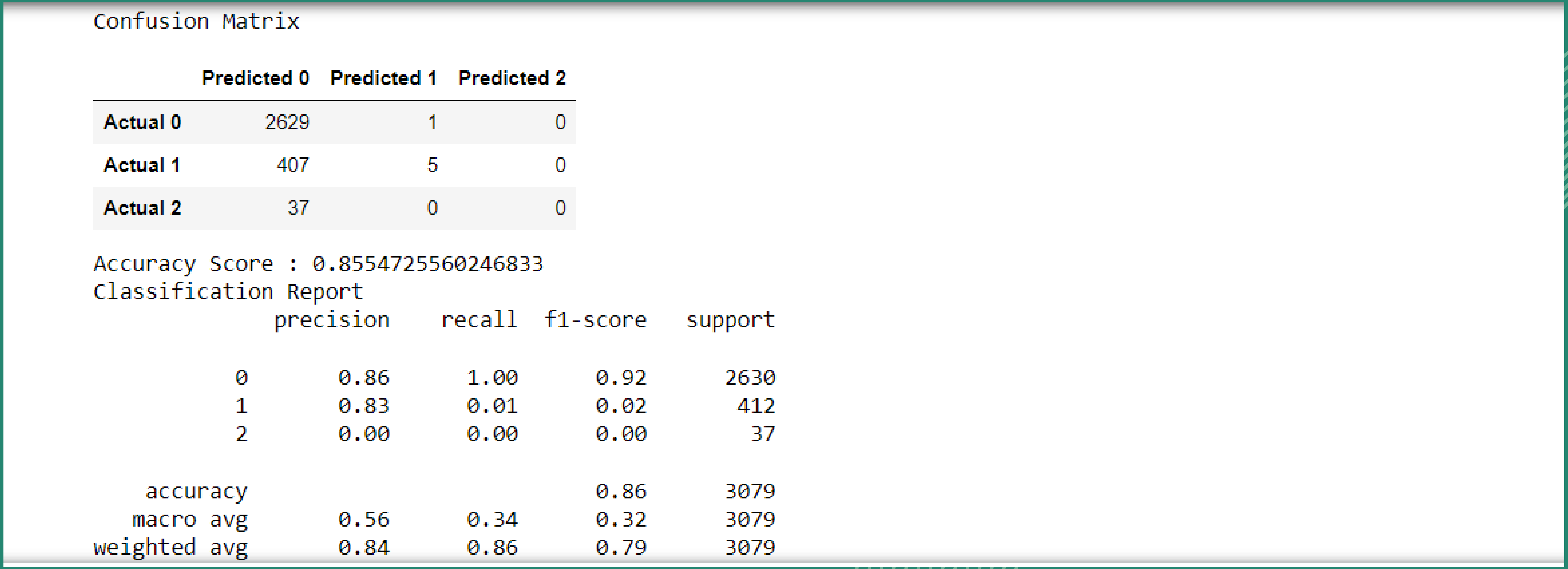
- Dataset from Kaggle
- Import libraries
- Identifying missing values
- Convert Time to datetime and extract hour
- Encoding Categorical data
- Concat 2 Data Frames
- Split dataset
- Standard scaling
- Train models
- Prediction

Random Forrest Model

- Used to handle large datasets with many features
- Identify important features to help identify which variables are most predictive of outcome
- Ability to handle both numerical and categorical data types
- Supervised Model

Random Forrest Model

85.5% Accuracy

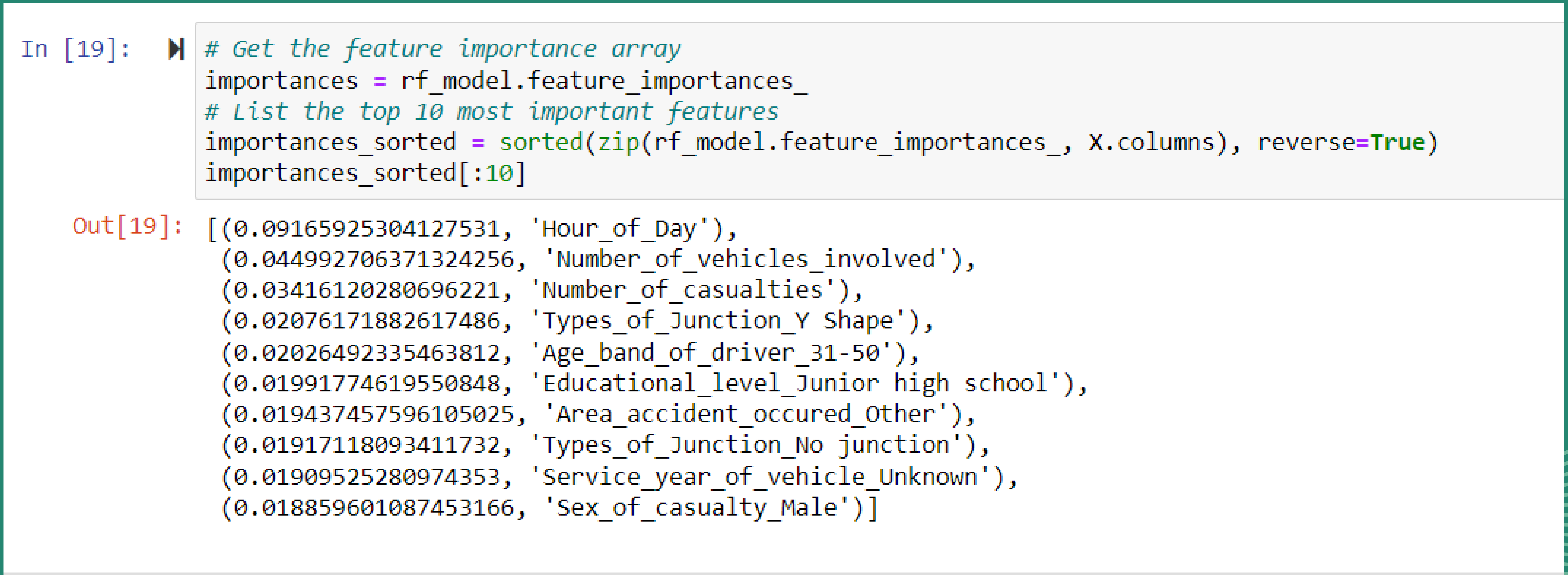


Confusion Matrix

	Predicted 0	Predicted 1	Predicted 2
Actual 0	2629	1	0
Actual 1	407	5	0
Actual 2	37	0	0

```
Accuracy Score : 0.8554725560246833
Classification Report
              precision    recall  f1-score   support

           0       0.86      1.00      0.92      2630
           1       0.83      0.01      0.02       412
           2       0.00      0.00      0.00        37

    accuracy                           0.86      3079
   macro avg       0.56      0.34      0.32      3079
weighted avg       0.84      0.86      0.79      3079
```

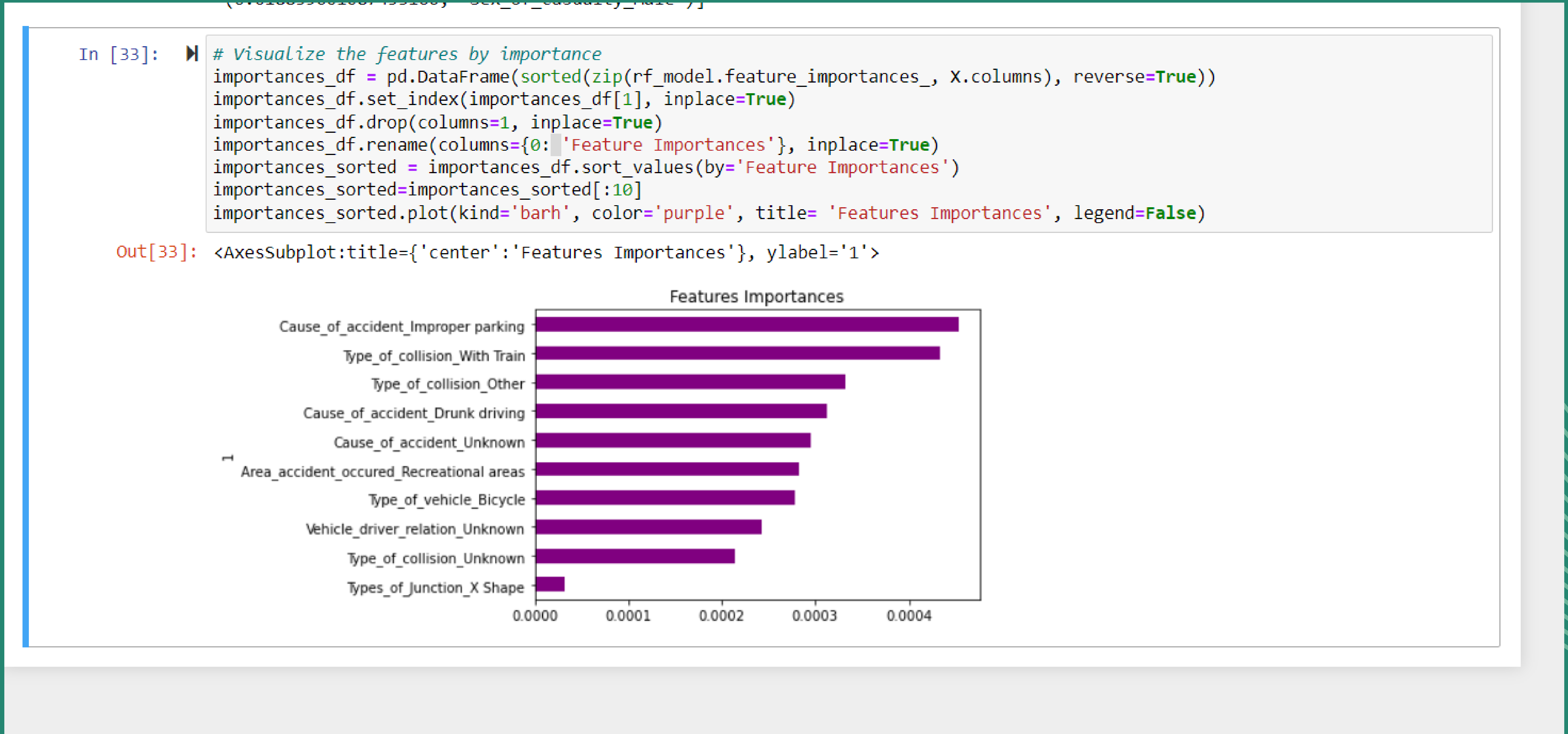
Random Forrest Model



```python
# Get the feature importance array
importances = rf_model.feature_importances_
# List the top 10 most important features
importances_sorted = sorted(zip(rf_model.feature_importances_, X.columns), reverse=True)
importances_sorted[:10]
```

```
[(0.09165925304127531, 'Hour_of_Day'),
 (0.04499270637132425б, 'Number_of_vehicles_involved'),
 (0.03416120280696221, 'Number_of_casualties'),
 (0.02076171882617486, 'Types_of_Junction_Y Shape'),
 (0.02026492335463812, 'Age_band_of_driver_31-50'),
 (0.01991774619550848, 'Educational_level_Junior high school'),
 (0.019437457596105025, 'Area_accident_occured_Other'),
 (0.01917118093411732, 'Types_of_Junction_No junction'),
 (0.01909525280974353, 'Service_year_of_vehicle_Unknown'),
 (0.018859601087453166, 'Sex_of_casualty_Male')]
```

Random Forrest Model



In [33]:

```
# Visualize the features by importance
importances_df = pd.DataFrame(sorted(zip(rf_model.feature_importances_, X.columns), reverse=True))
importances_df.set_index(importances_df[1], inplace=True)
importances_df.drop(columns=1, inplace=True)
importances_df.rename(columns={0: 'Feature Importances'}, inplace=True)
importances_sorted = importances_df.sort_values(by='Feature Importances')
importances_sorted=importances_sorted[:10]
importances_sorted.plot(kind='barh', color='purple', title= 'Features Importances', legend=False)
```

Out[33]:

```
<AxesSubplot:title={'center':'Features Importances'}, ylabel='1'>
```

Features Importances
Cause_of_accident_Improper parking
Type_of_collision_With Train
Type_of_collision_Other
Cause_of_accident_Drunk driving
Cause_of_accident_Unknown
Area_accident_occured_Recreational areas
Type_of_vehicle_Bicycle
Vehicle_driver_relation_Unknown
Type_of_collision_Unknown
Types_of_Junction_X Shape
1
0.0000
0.0001
0.0002
0.0003
0.0004

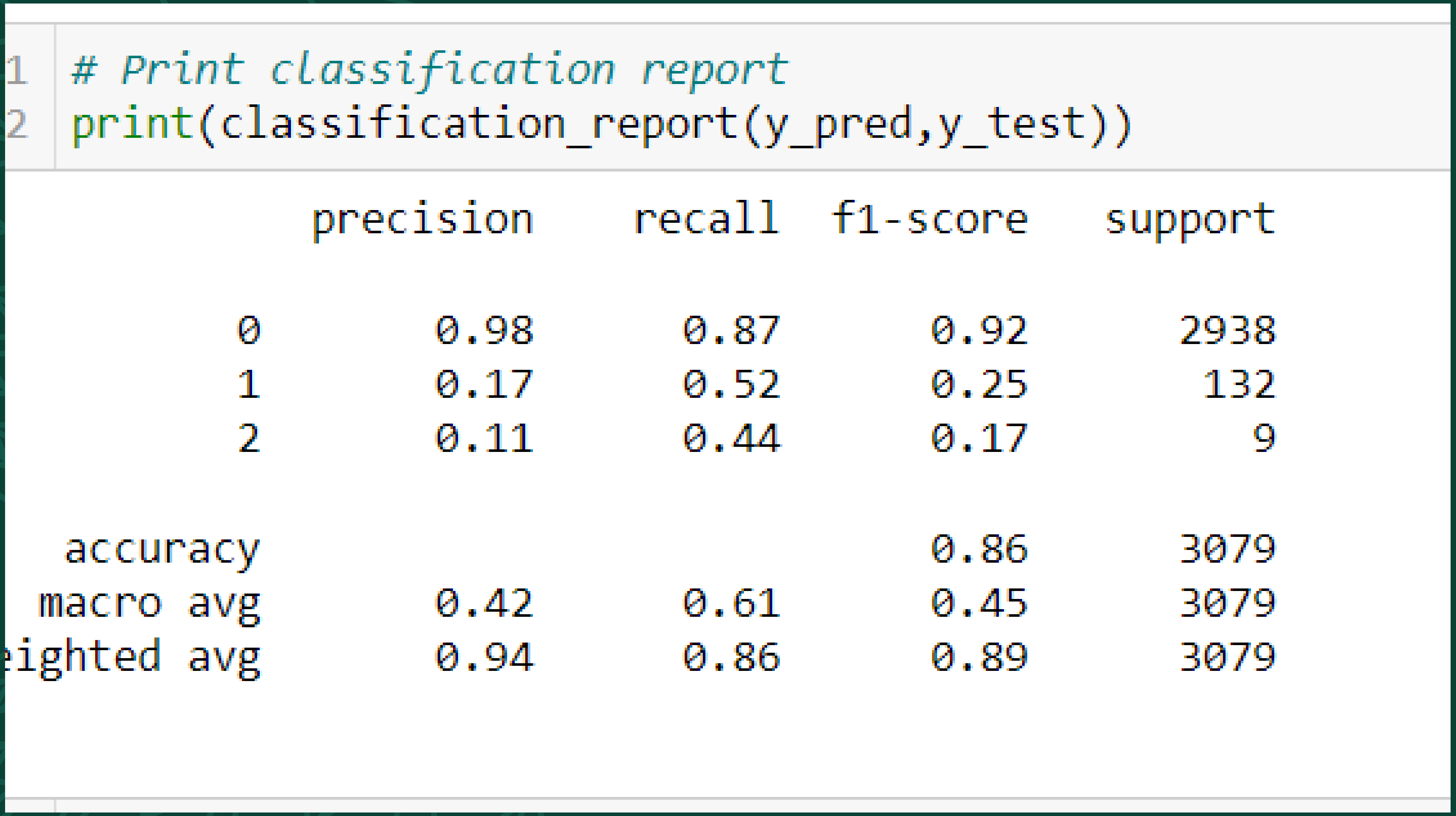
K-Nearest Neighbors Model

- Supervised learning
- It is a simple and easy-to-understand
- Used in both regression and classification predictive problems

K-Nearest Neighbors Model

Back to Agenda

- Trained to predict the severity of the accident based on certain features such as weather conditions, road type, time and etch.
- K=2 neighbors accuracy 86%

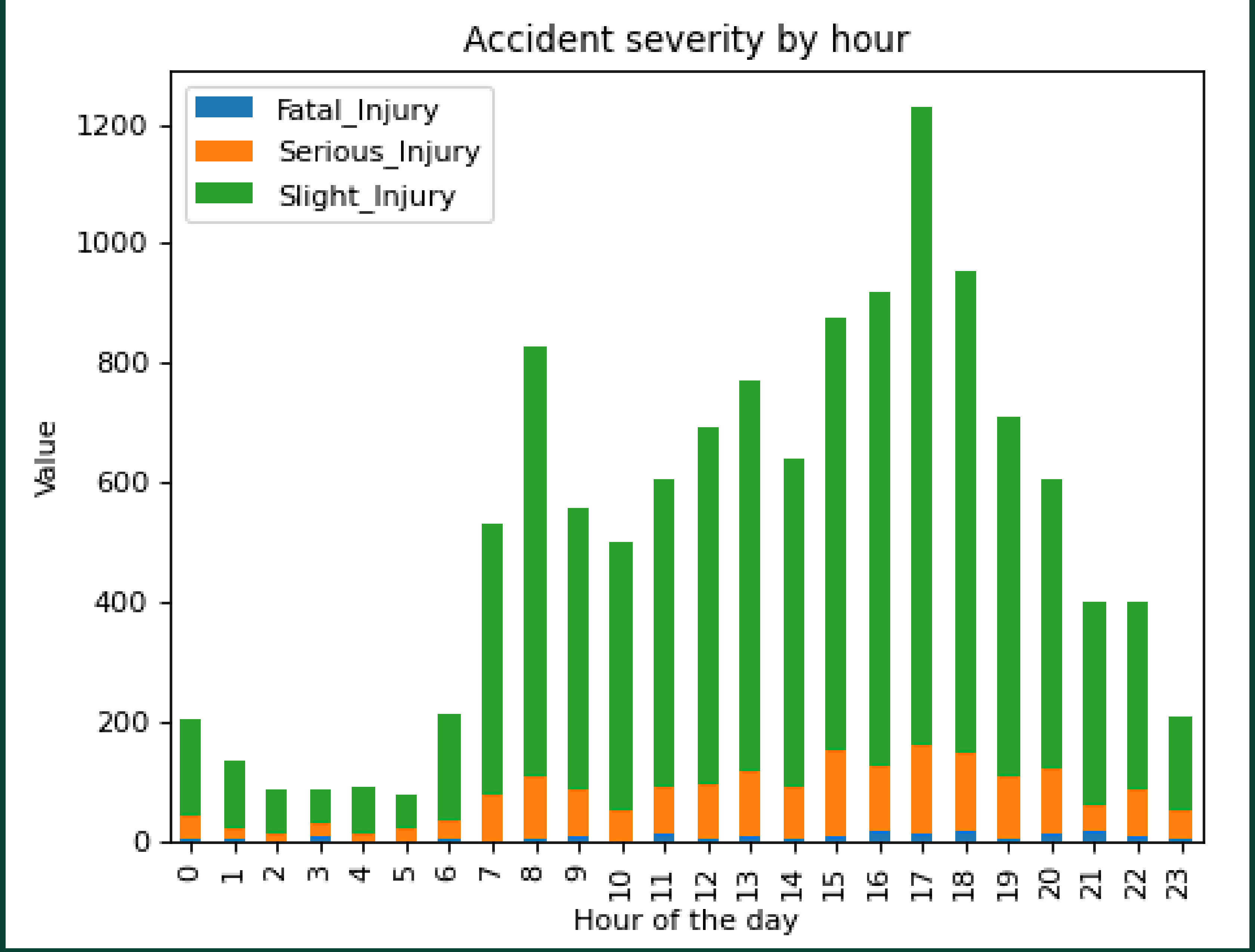


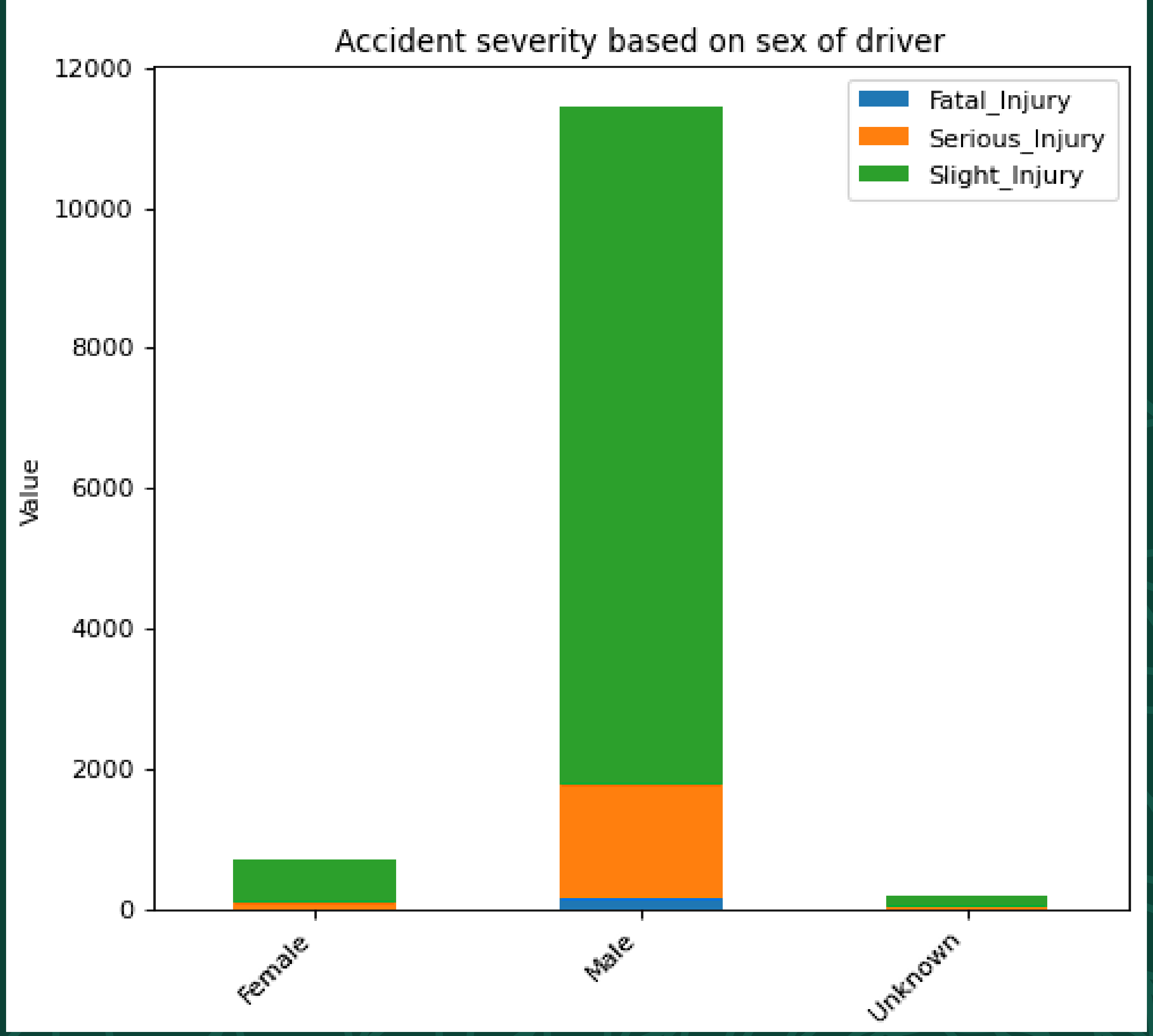
```
# Print classification report
print(classification_report(y_pred,y_test))
```

```
              precision    recall  f1-score   support

           0       0.98      0.87      0.92      2938
           1       0.17      0.52      0.25       132
           2       0.11      0.44      0.17         9

    accuracy                           0.86      3079
   macro avg       0.42      0.61      0.45      3079
ighted avg       0.94      0.86      0.89      3079
```

Visualizations

Back to Agenda



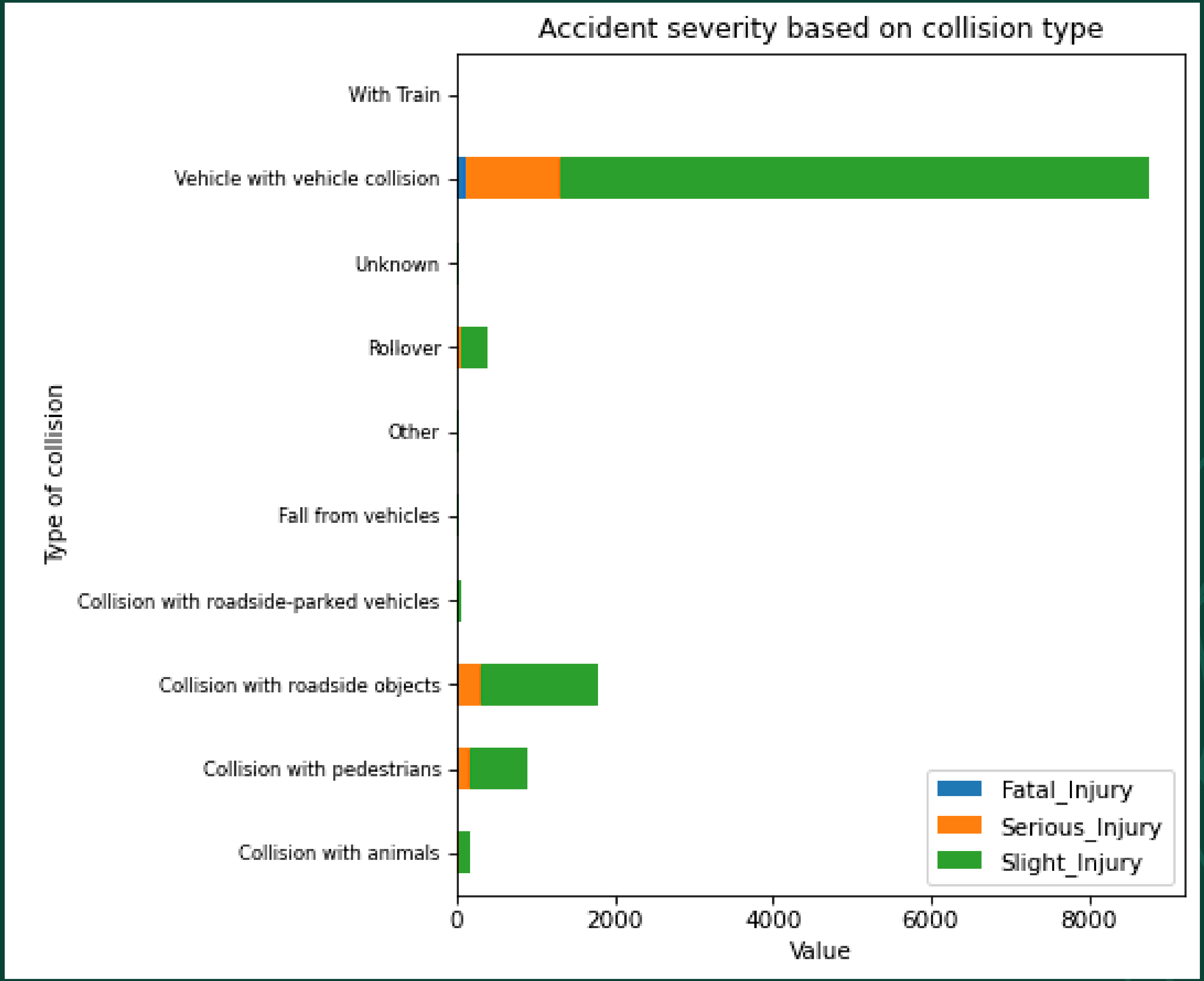


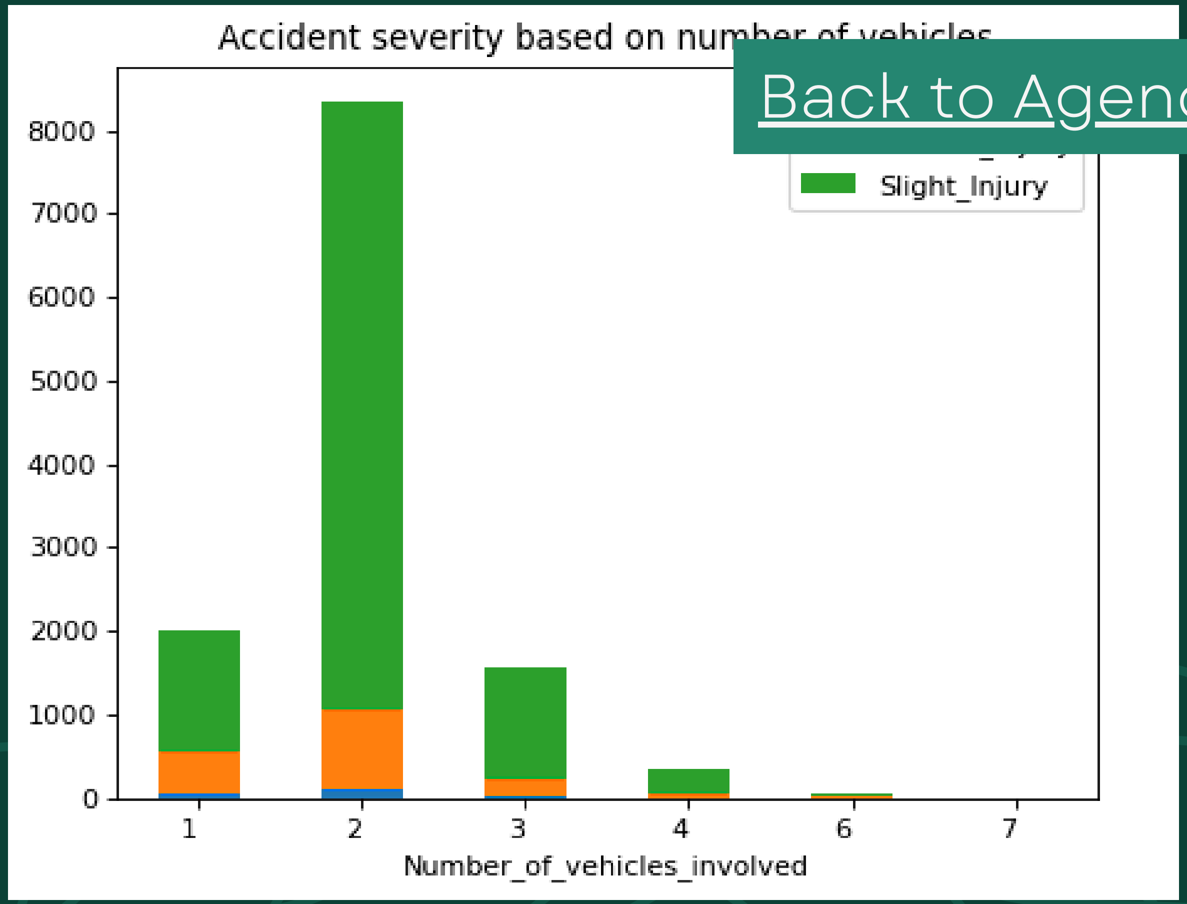
5PM is statistically the worst time to drive bases on this dataset

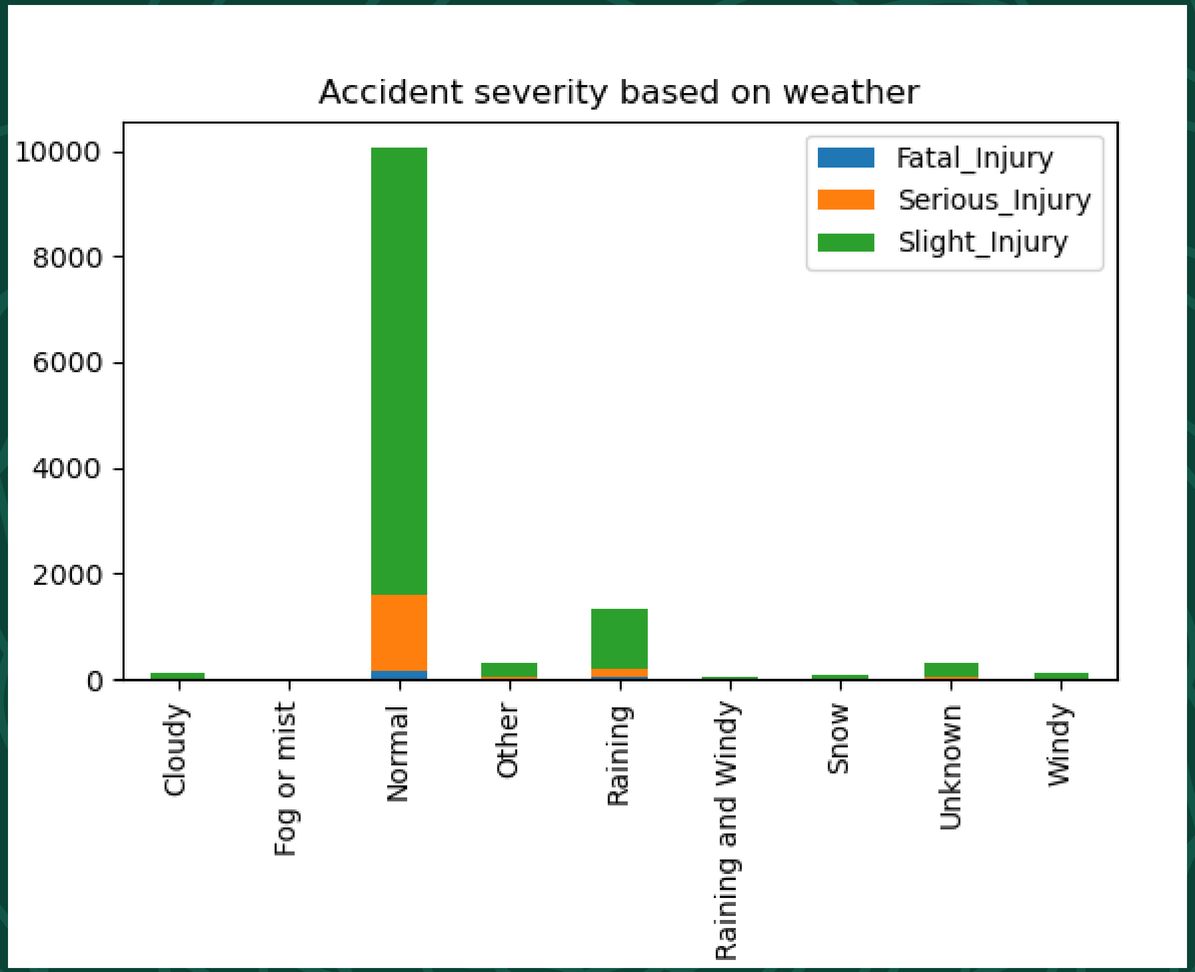
There is an extreme difference between the amount of accidents females and males are involved in. Based off this dataset

Visualizations

Back to Agenda







Thank you for attending our presentation