

Accident Analysis: Predicting Traffic Accident Duration



By Chase, Pablo, Victor, Ming, Hanson,
Kenneth, Swayam

Introduction

What specific factors significantly influenced the duration of traffic accidents across various cities?

13 DECEMBER 2023

CHASE HERTEL



Dataset

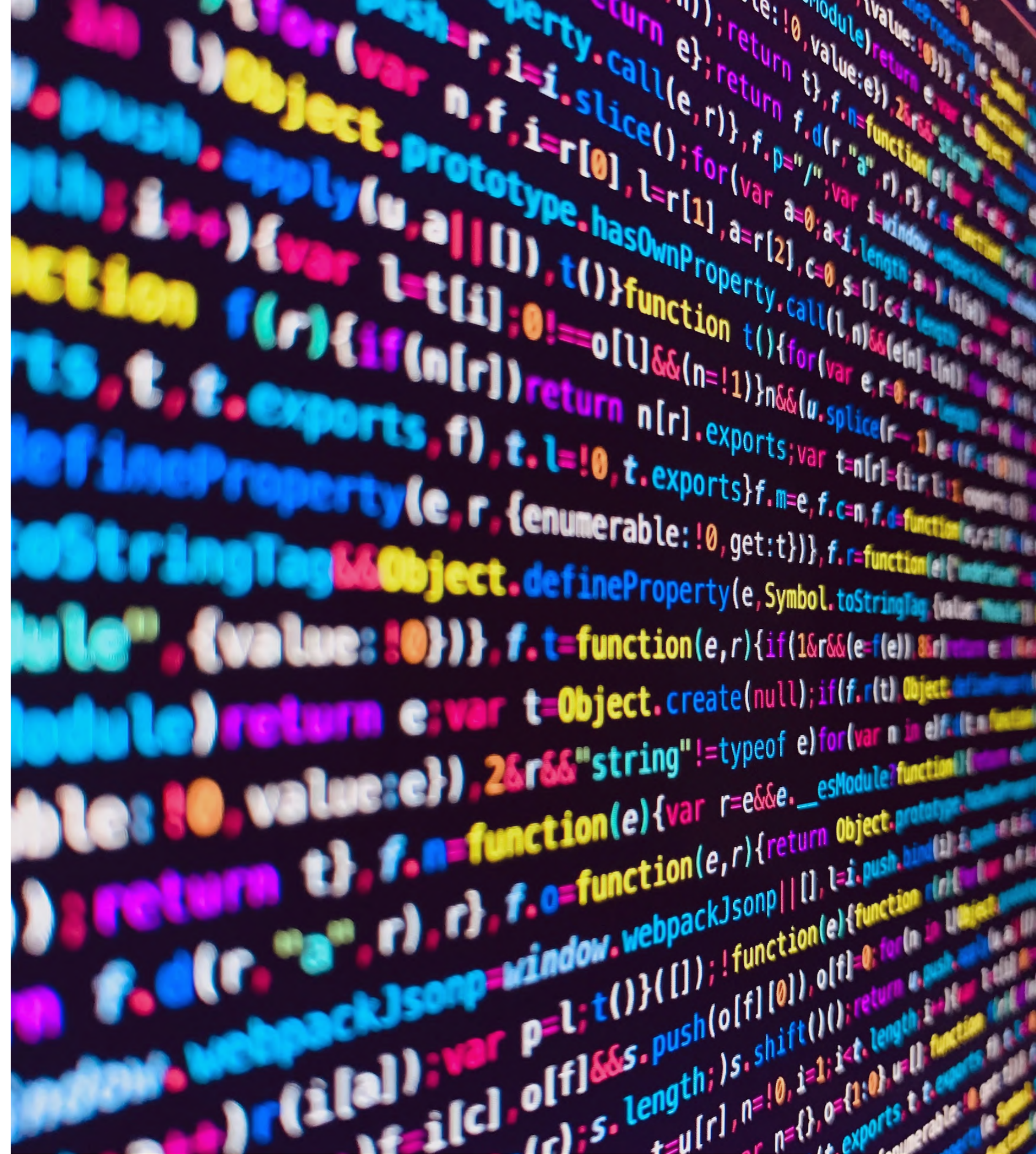
US Accidents (2016 – 2023): A Countrywide Traffic Accident Dataset from Kaggle

Contains car accident information from across 49 U.S. states, The data originates from sources such as traffic cameras, sensors, and government agencies, offering a comprehensive overview of approximately 7.7 million recorded accidents. These incidents were gathered through multiple APIs that stream real-time traffic event data.

Weather, time, and location-based features are some of the most important attributes and make up some of total 46 columns

13 DECEMBER 2023

CHASE HERTEL



Accident Patterns and Duration Analysis



Insights



Methods



Recommendations

Houston

Worst traffic in Texas, and ranked 2nd in the US for worst traffic



Dataset

Trained on historical logged accidents data in Houston, TX

Prediction

Predict accident duration in seconds using logged data

Insight

Predict accident duration as soon as accident data is initially logged for city and public usage

Home of one of the largest interchanges in the world



Houston

Decision Tree Random Forest

MLP

```
from sklearn.tree import DecisionTreeClassifier
best_dt = DecisionTreeClassifier(
    criterion='entropy',
    max_depth=40,
    max_features='log2',
    min_samples_leaf=12,
    min_samples_split=7
)
best_dt.fit(X_train, y_train)
```

Score: 0.24

Accuracy: 0.36

```
from sklearn.ensemble import RandomForestClassifier
best_rf = RandomForestClassifier(
    max_depth=30,
    max_features='sqrt',
    min_samples_leaf=1,
    min_samples_split=13,
    n_estimators=67
)
best_rf.fit(X_train, y_train)
```

Score: 0.27

Accuracy: 0.55

```
from sklearn.neural_network import MLPClassifier
best_mlp = MLPClassifier(
    hidden_layer_sizes=(50,),
    activation='tanh',
    solver='adam',
    verbose=1
)
best_mlp.fit(X_train, y_train)
```

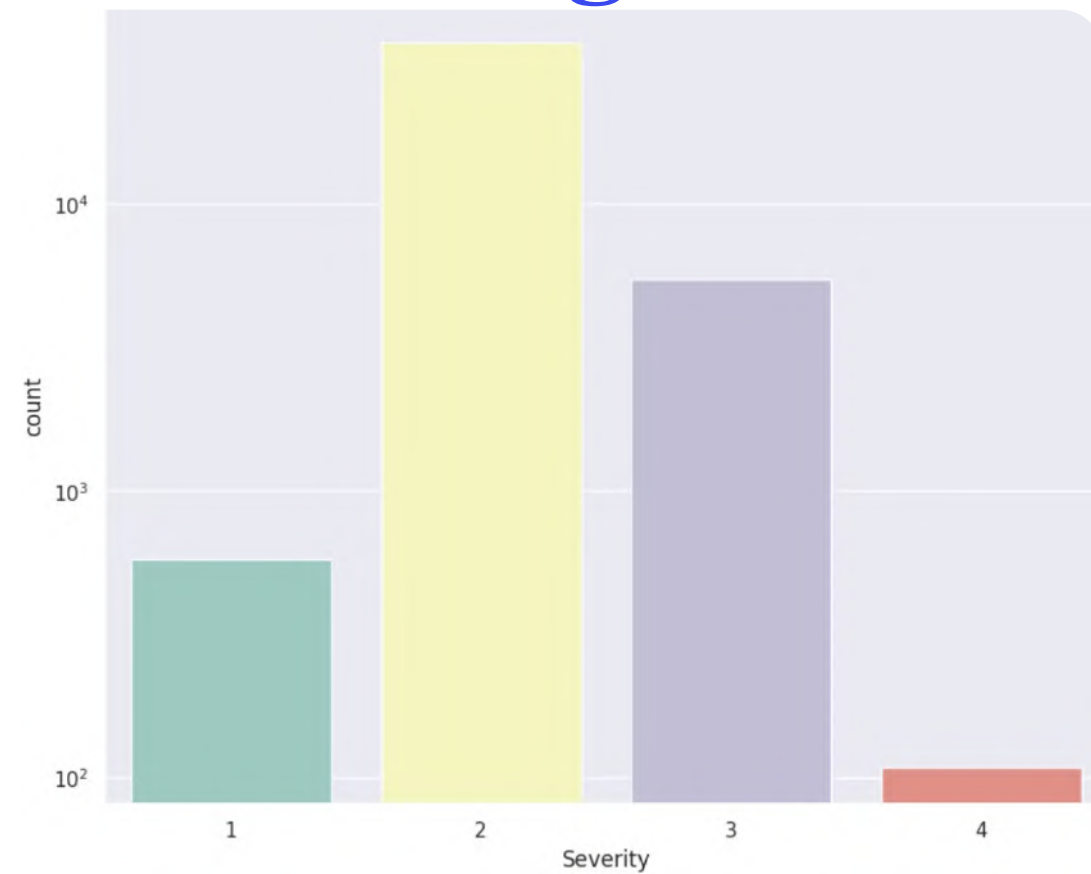
Score: 0.26

Accuracy: 0.36

Houston

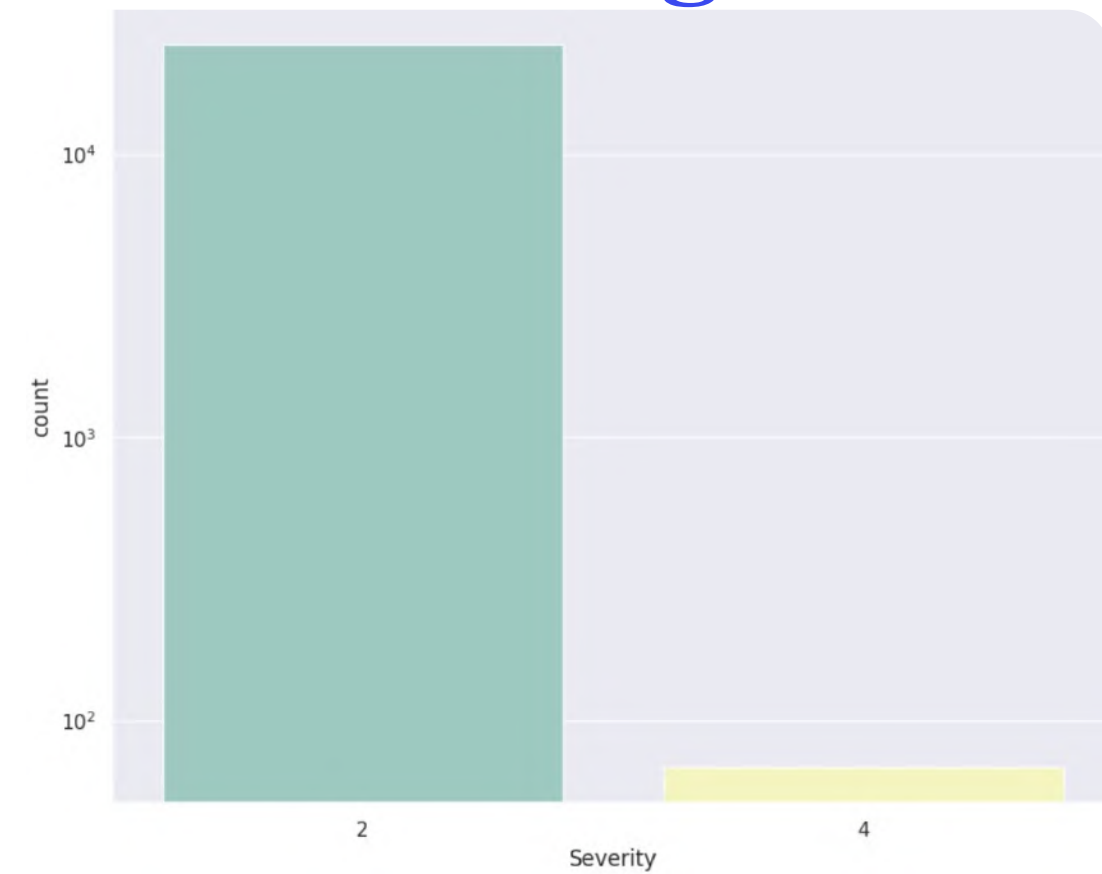
Severity of Accidents

Missing Data



Missing end longitude and latitude, however those are key features in predicting accident duration

No Missing Data

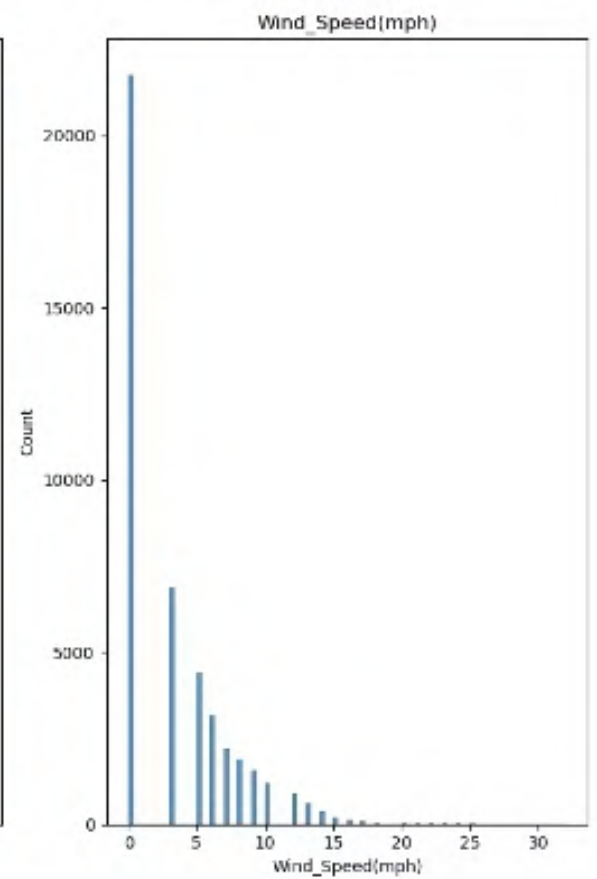
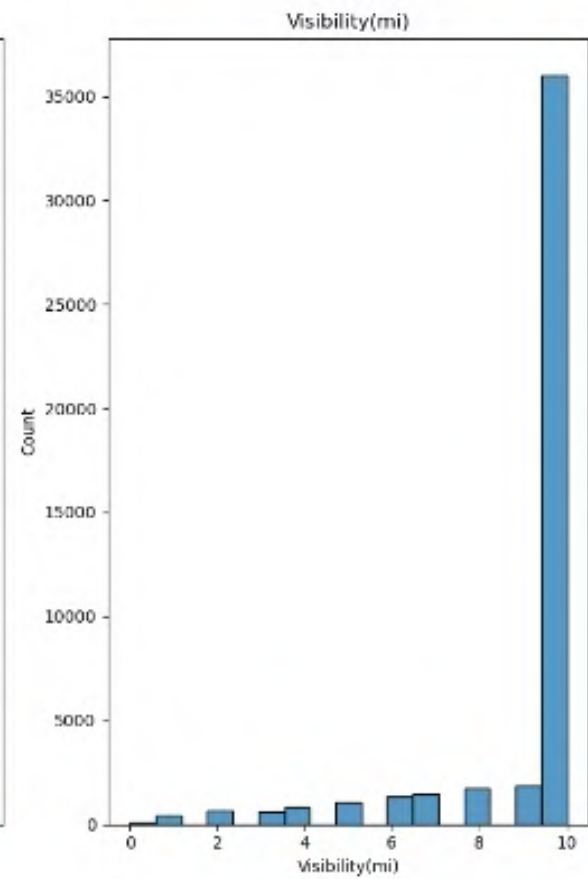
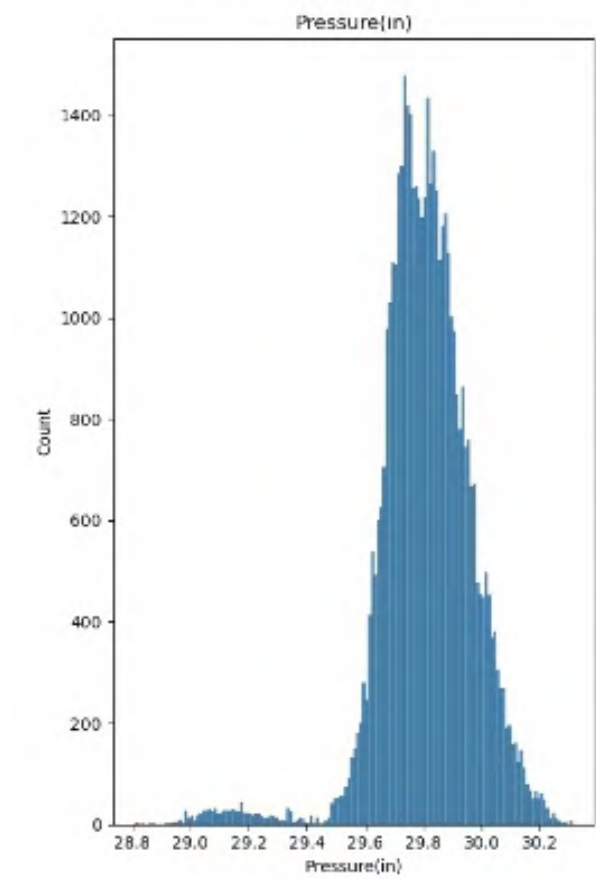
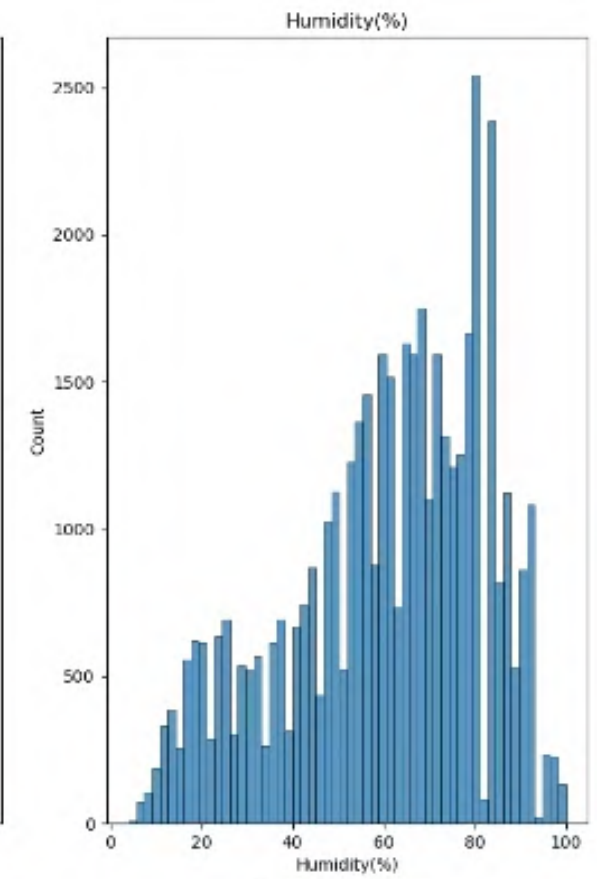
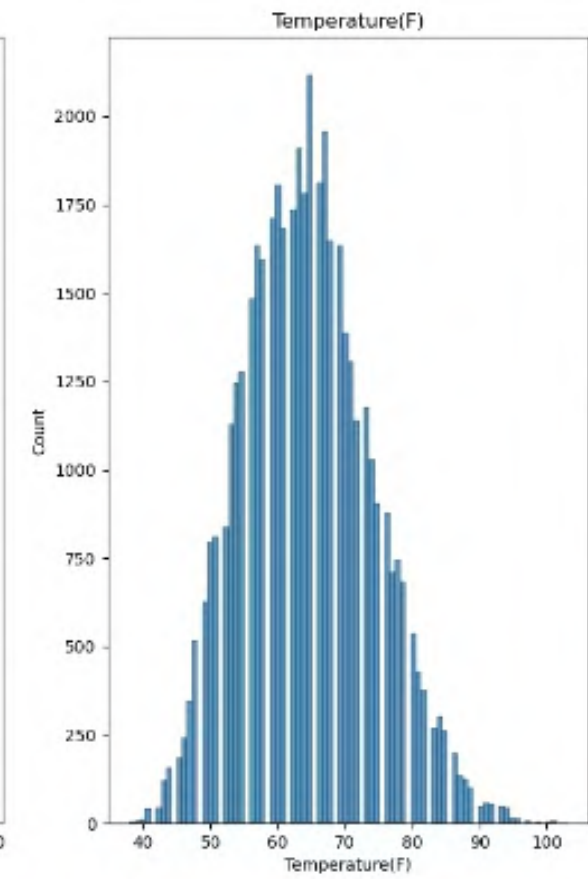
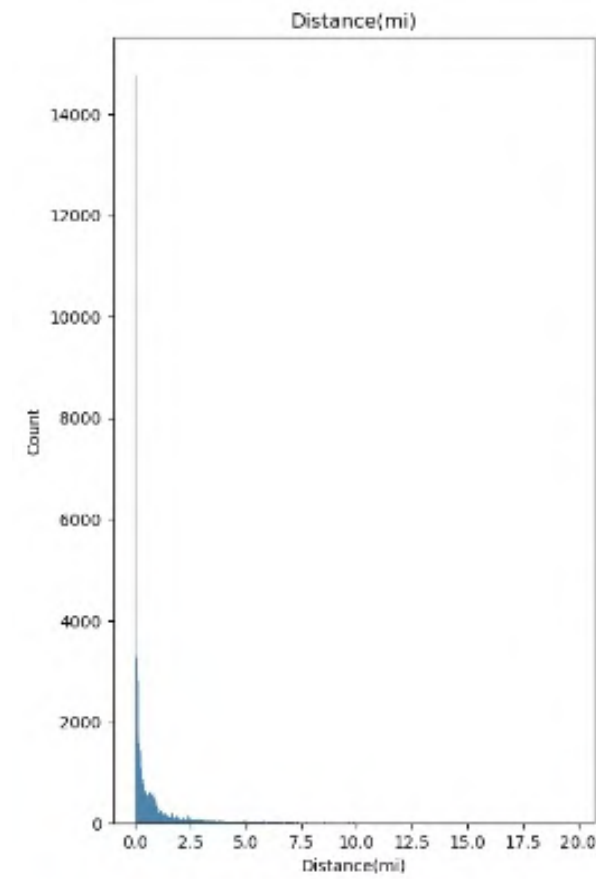
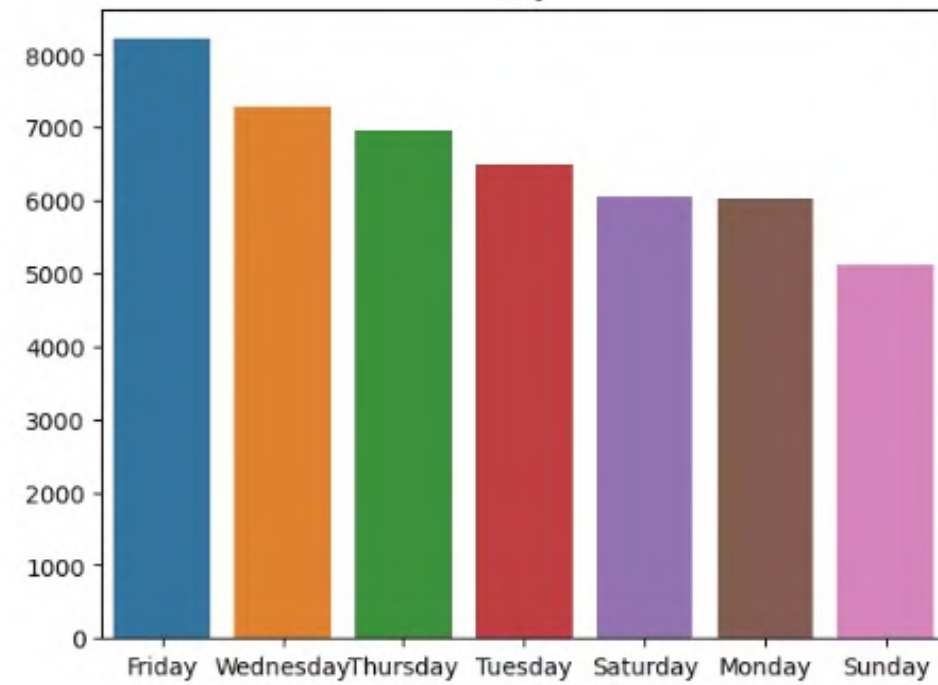
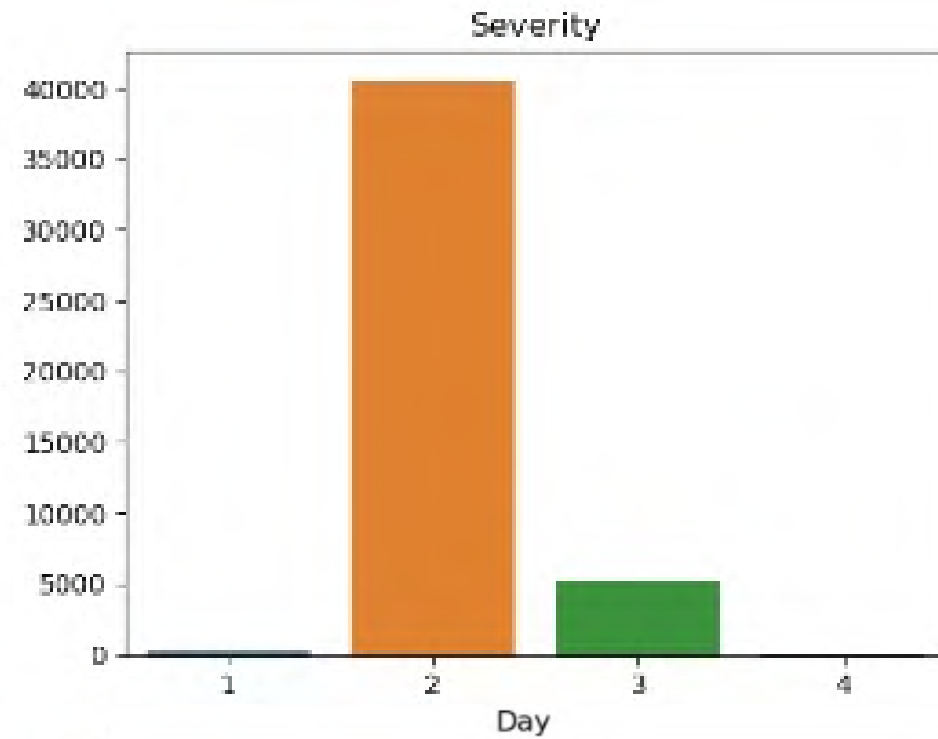


Los Angeles



1. One of the largest cities in the states(high population density)
2. Very car-dependent (city is structured around cars)
3. Attracts lots of people (e.g. tourism)
4. Poor road infrastructure (due to rapid growth)

EDA



Feature Engineering

```
# Converting Duration to categorys

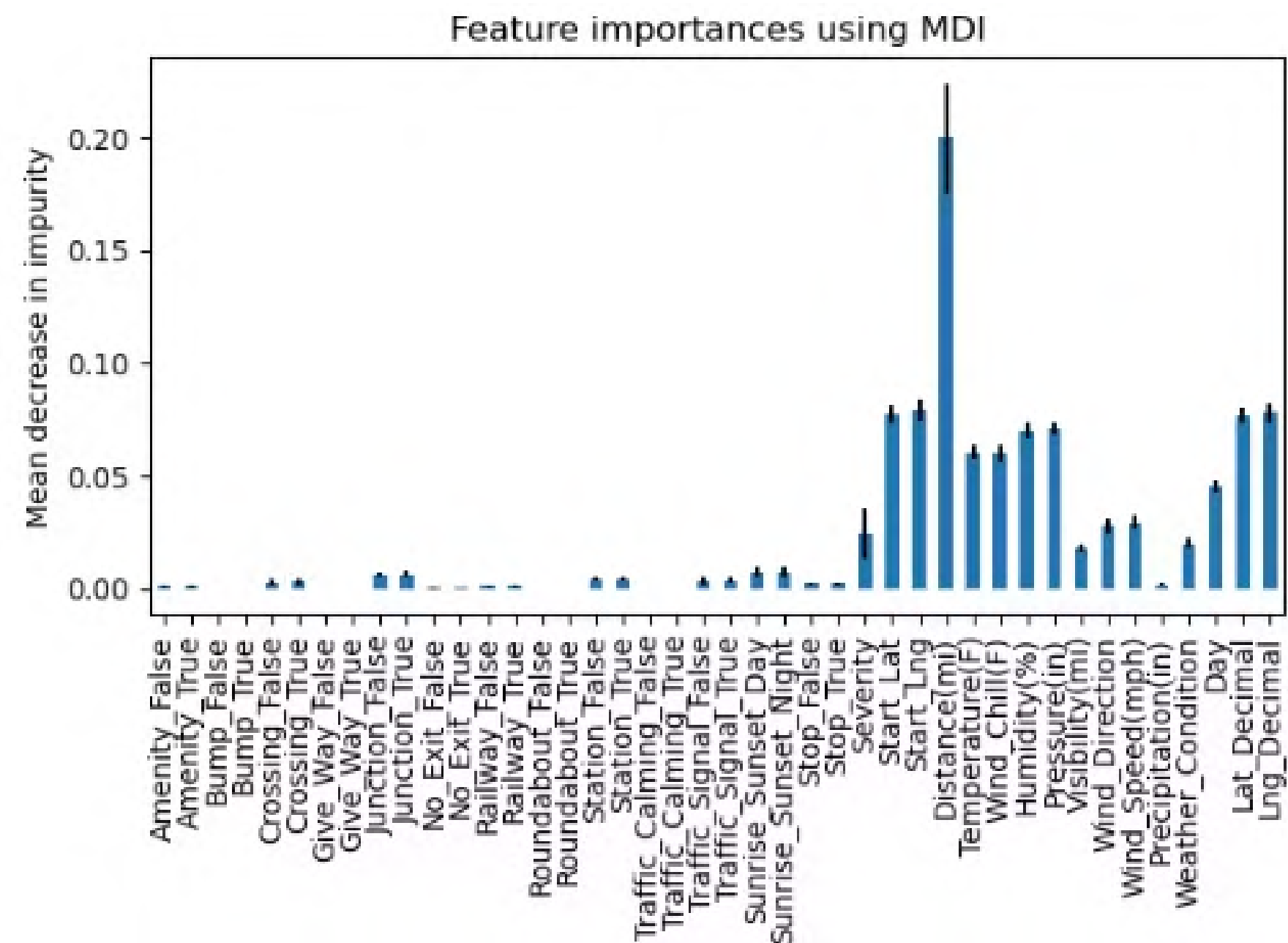
# 15 mins or less = 0
# 15 - 30mins = 1
# 30 mins - 1hr = 2
# 1 - 3hr = 3
# 3hr - 6hr = 4
# rest of day = 5

la['ETA'] = 0
la.loc[la['Accident_Duration'] <= 900, 'ETA'] = 0
la.loc[(la['Accident_Duration'] <= 1800) & (la['Accident_Duration'] > 900), 'ETA'] = 1
la.loc[(la['Accident_Duration'] <= 3600) & (la['Accident_Duration'] > 1800), 'ETA'] = 2
la.loc[(la['Accident_Duration'] <= 10800) & (la['Accident_Duration'] > 3600), 'ETA'] = 3
la.loc[(la['Accident_Duration'] <= 21600) & (la['Accident_Duration'] > 10800), 'ETA'] = 4
la.loc[la['Accident_Duration'] > 21600, 'ETA'] = 5
```

```
# Get the decimal points of Latitude and Longitude so it can be more sensitive to the model for Location
la['Lat_Decimal'] = la.Start_Lat.astype(str).str.extract('\.(.*)').astype(int)
la['Lng_Decimal'] = la.Start_Lng.astype(str).str.extract('\.(.*)').astype(int)
```

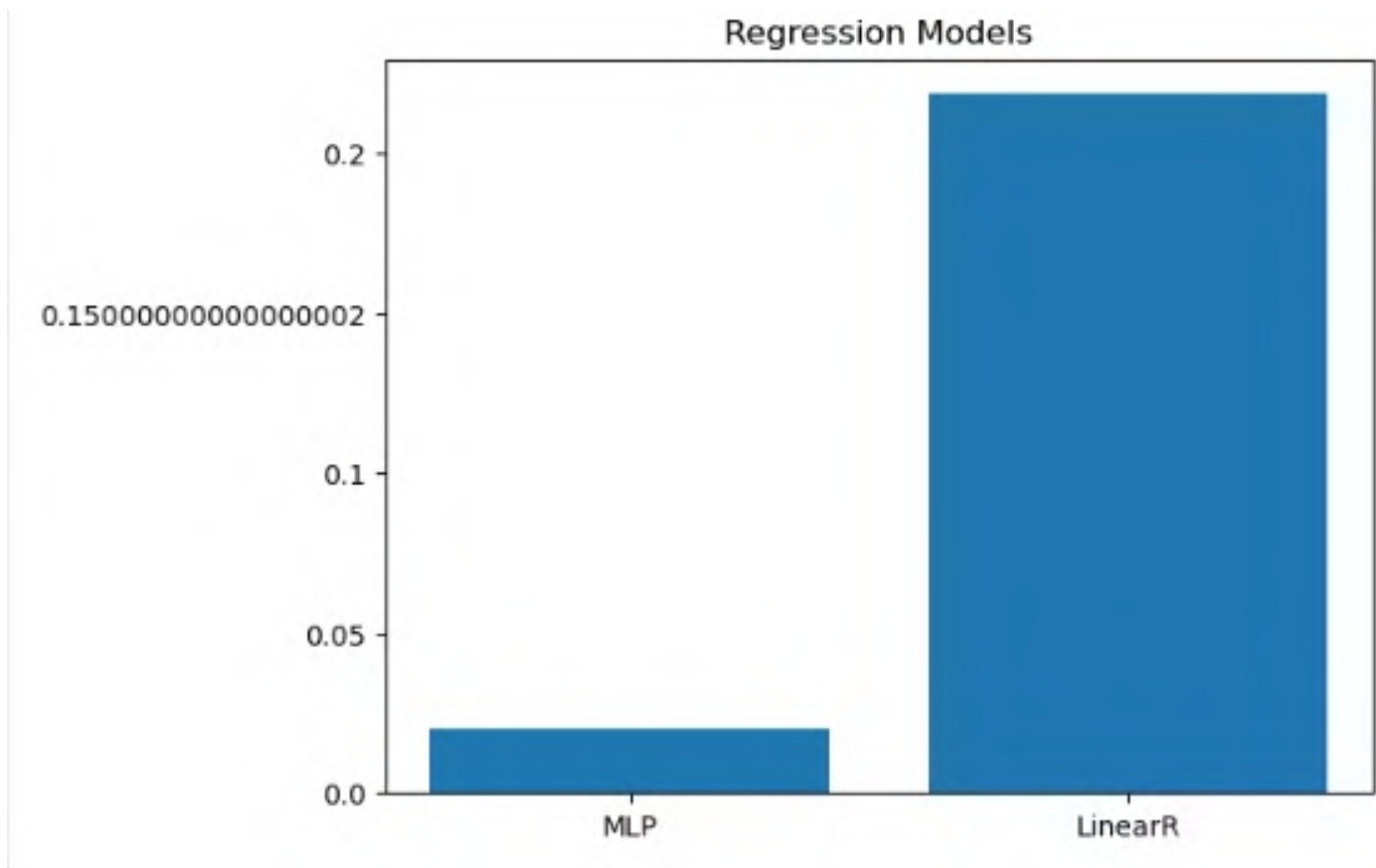
Lat_Decimal	Lng_Decimal
98748	137558
9289590000000000	388271
989342	256482
4945	270073
30895	217926

Feature Importance

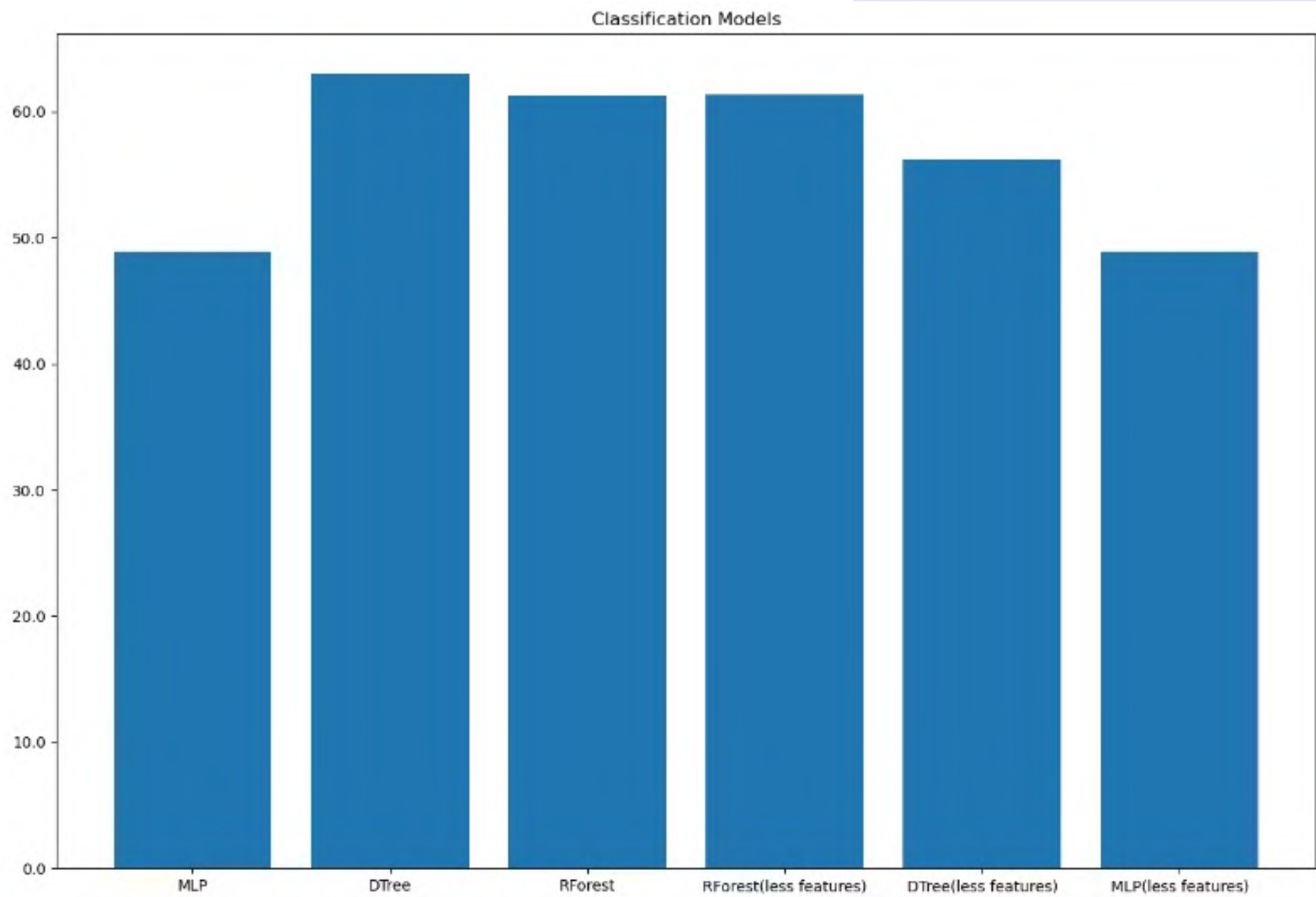


- Distance plays a large role in predicting accident duration
- Latitude, Longitude, Temperature, Wind Chill, Humidity, and Pressure plays a relatively significant role as well.
- Day and Severity doesn't contribute as much as I thought (expecting more congestion and delays due to certain days and severity of accidents)
- Try including these features only?

Results



- Best score is 63% from DecisionTreeClassifier.
- Potential future algorithm to consider is LSTM or RNN for Regression



Accuracy(%): 48.9 63.0 61.3 61.4 56.2 48.9

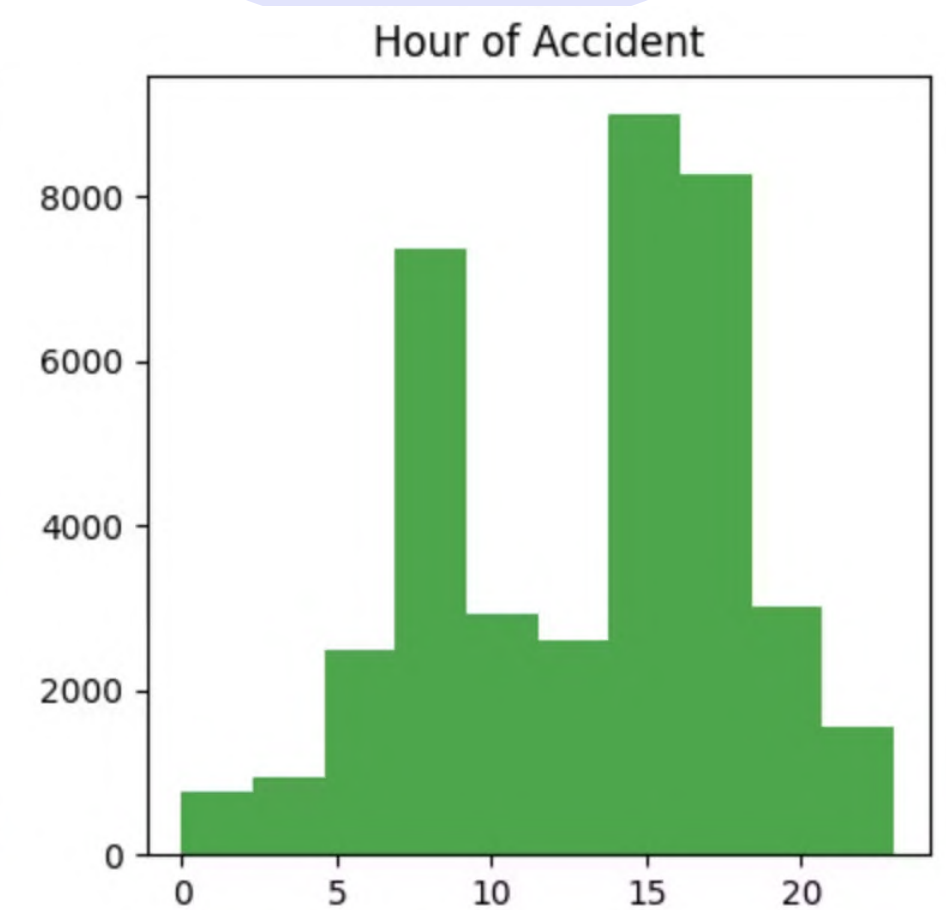
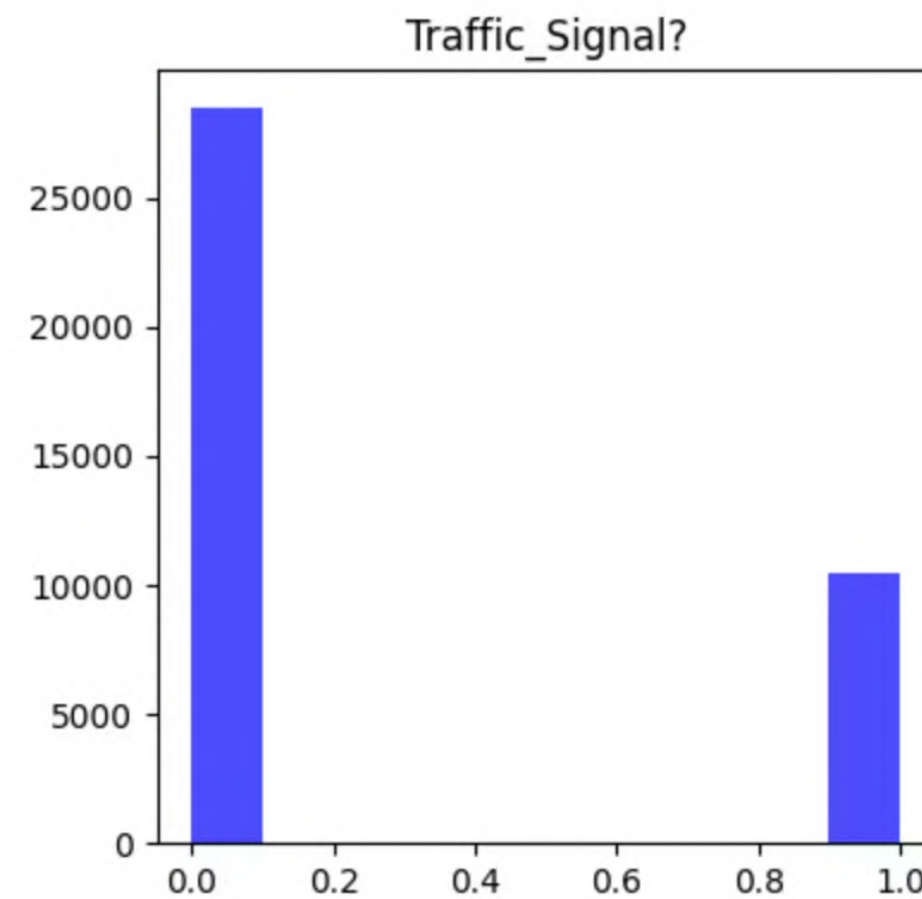
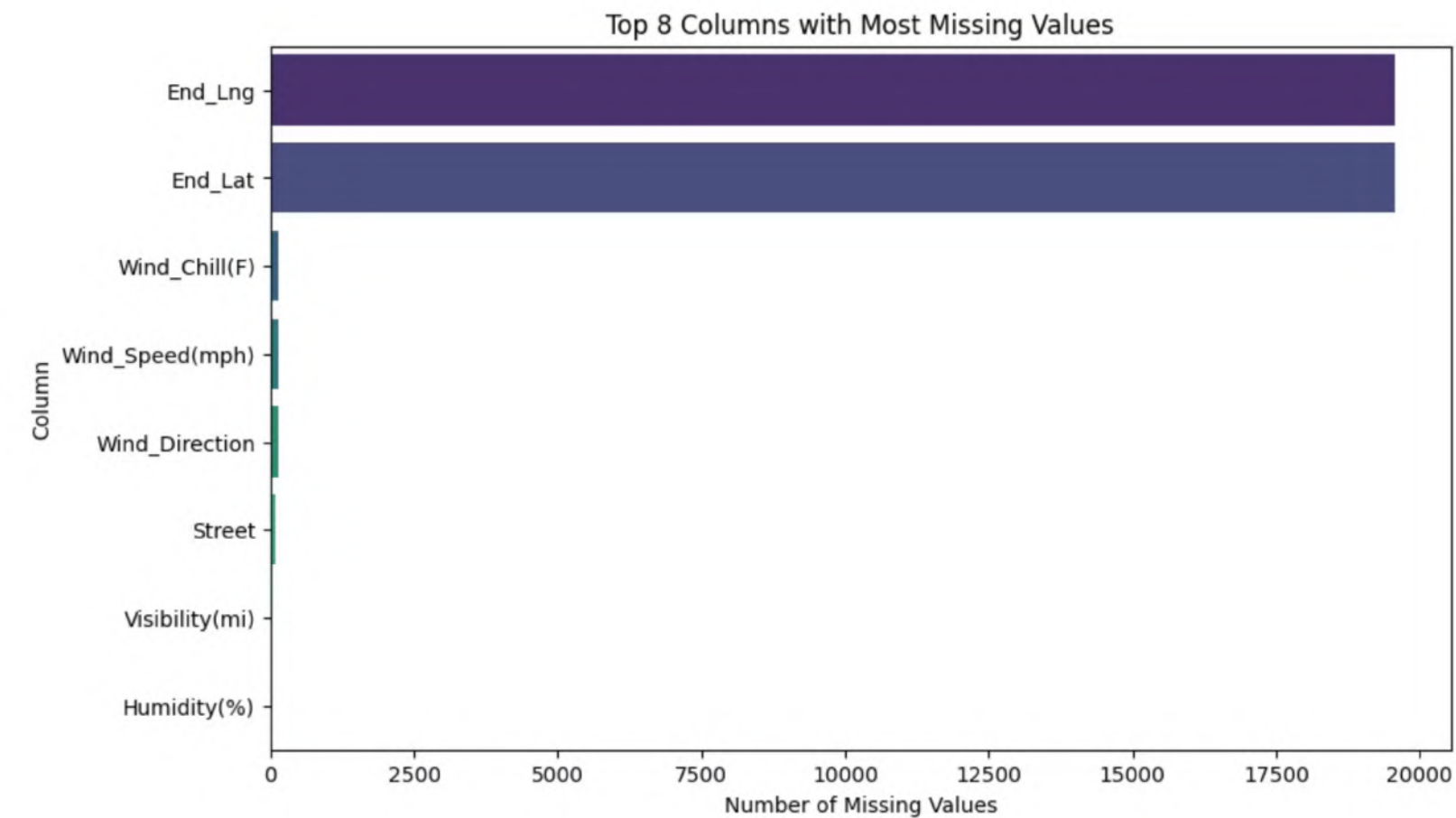
Charlotte, North Carolina



One of the top 5 cities with most accidents in the US
Past 2 years have seen 39000+ reported traffic accidents
Above average rush-hour traffic travel times
According to surveys, Charlotte ranks in top 10 worst public transportation systems

EDA and Data Cleaning

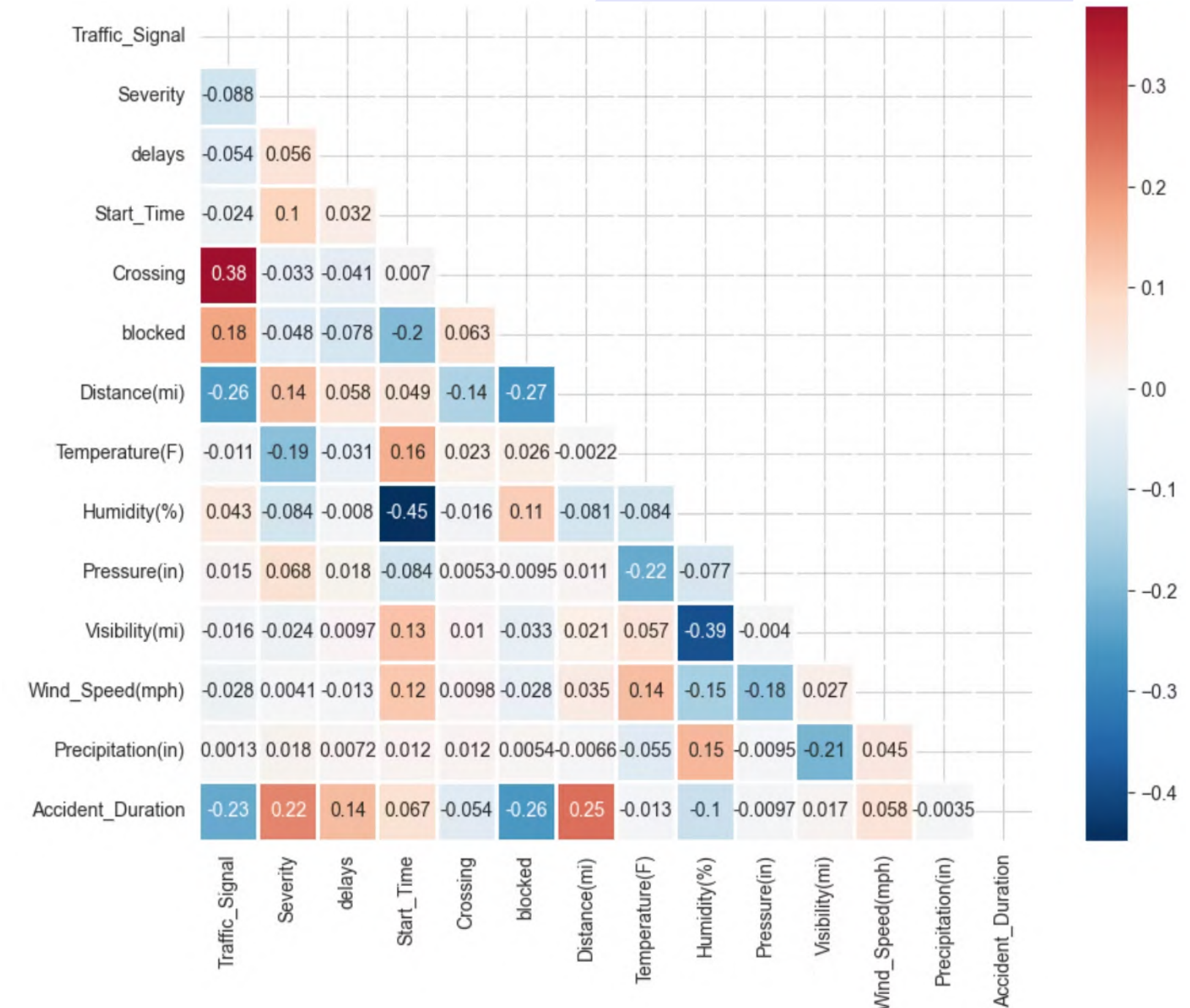
Accidents in dataset: 39150



Feature Engineering

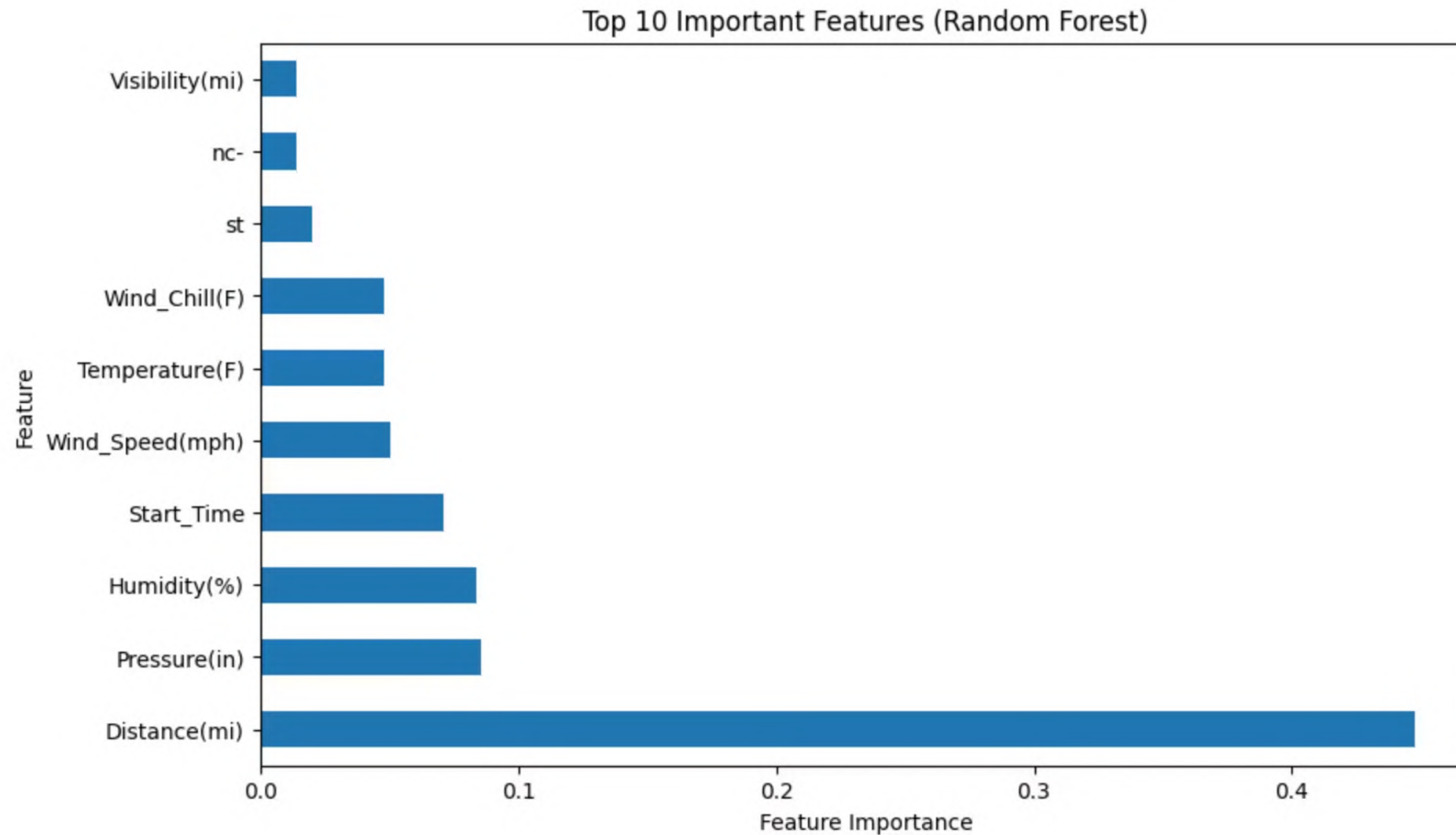
```
1 target_column = 'Accident_Duration'
2 numeric_columns = df.select_dtypes(include=['number'])
3 correlation_matrix = numeric_columns.corr()
4 correlations = correlation_matrix[target_column].drop(target_column)
5 sorted_features = correlations.abs().sort_values(ascending=False)
6 print(sorted_features)
```

blocked	0.254254
Distance(mi)	0.243359
Traffic_Signal	0.223407
Severity	0.221620
delays	0.145592
Humidity(%)	0.096498
Start_Time	0.066515
Crossing	0.065914
Wind_Speed(mph)	0.059779
.	0.000000

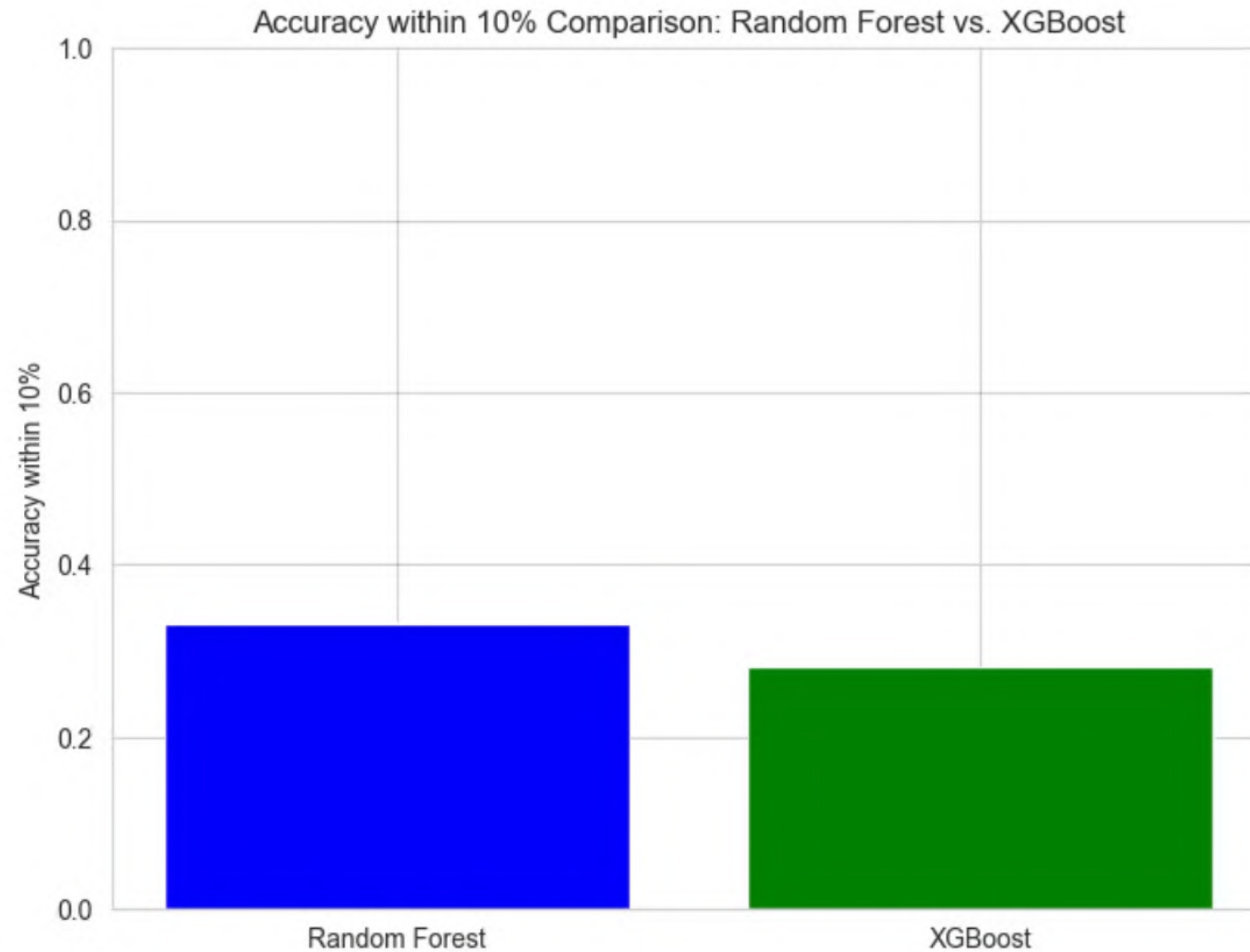


Features that I chose = ['Junction', 'blocked', 'Distance(mi)', 'Temperature(F)', 'Traffic_Signal', 'delays', 'Start_Time', 'Humidity(%)', 'Pressure(in)', 'Crossing', 'Wind_Speed(mph)']

Feature Engineering

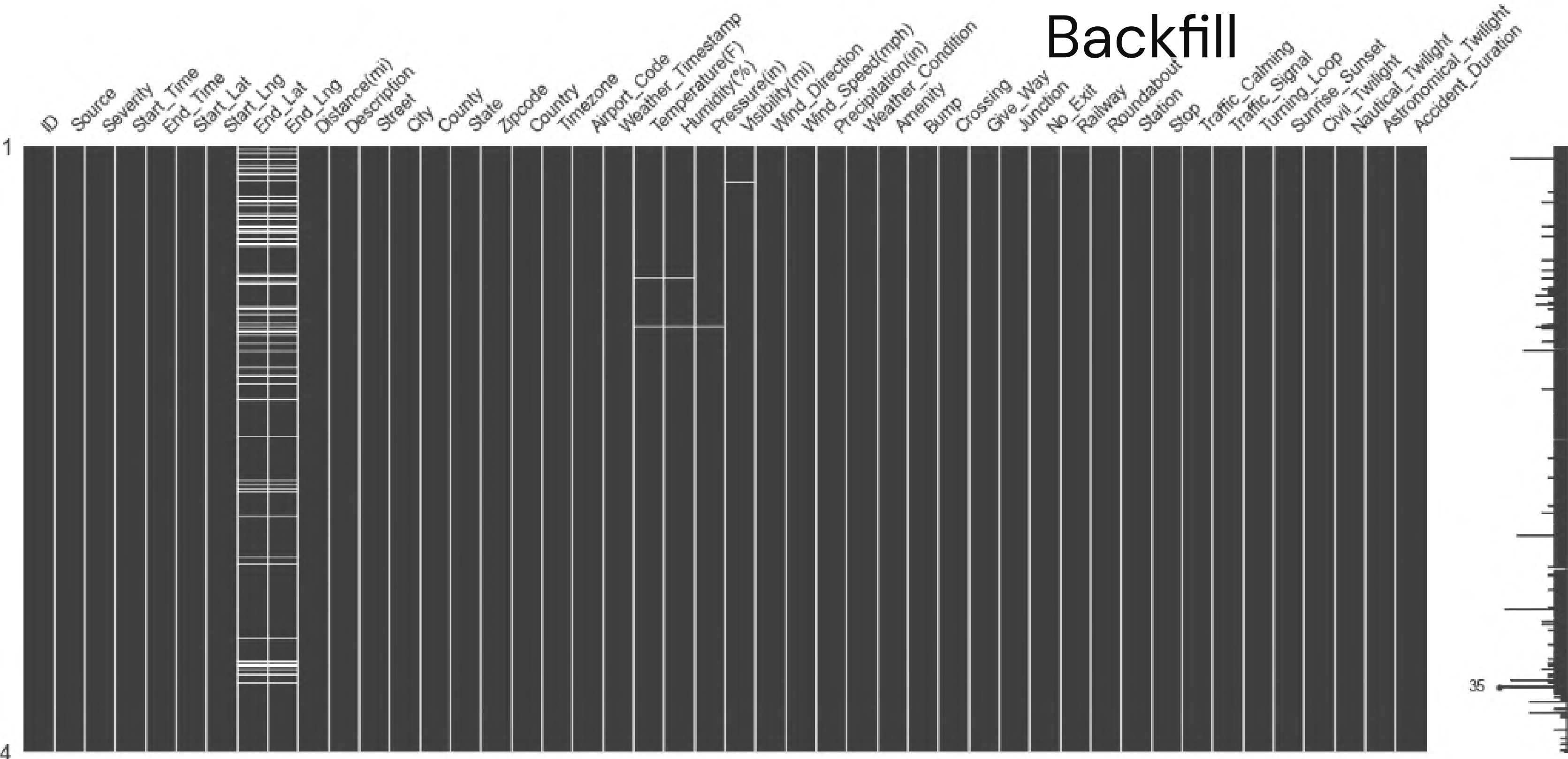


Model Performance



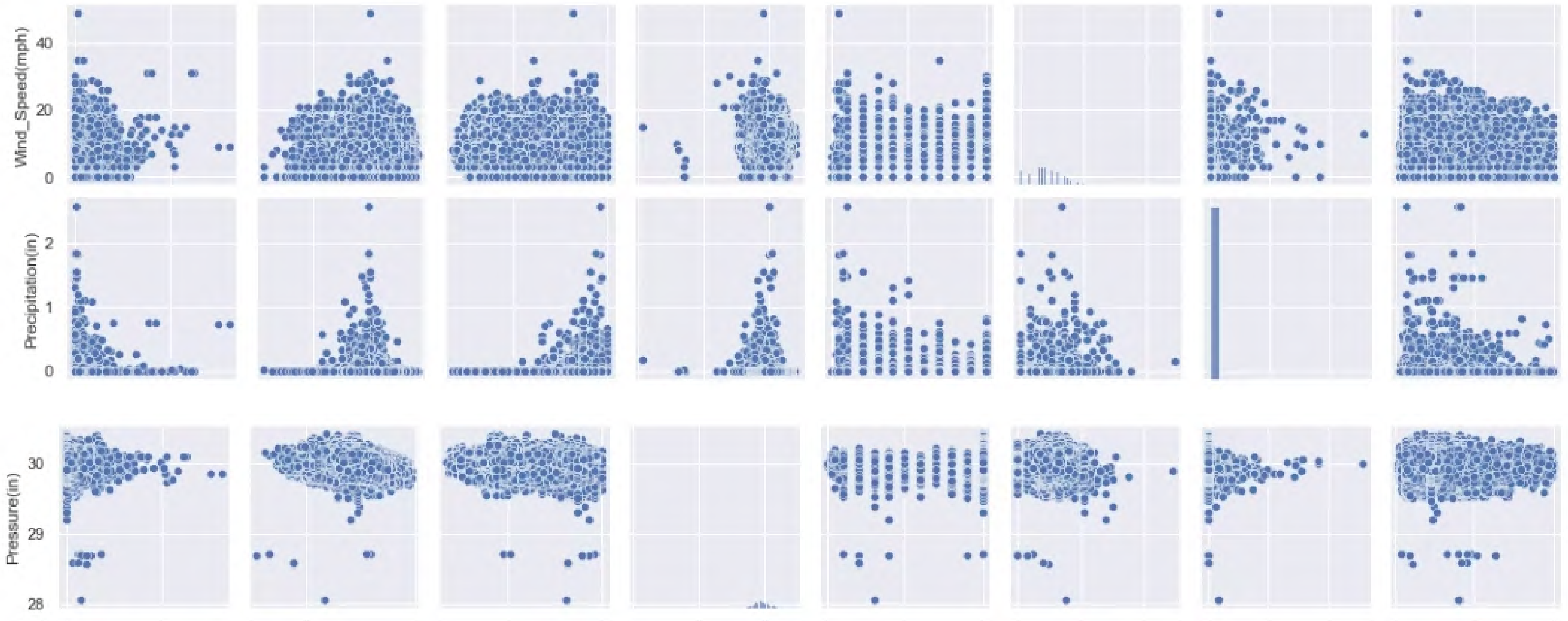
Orlando

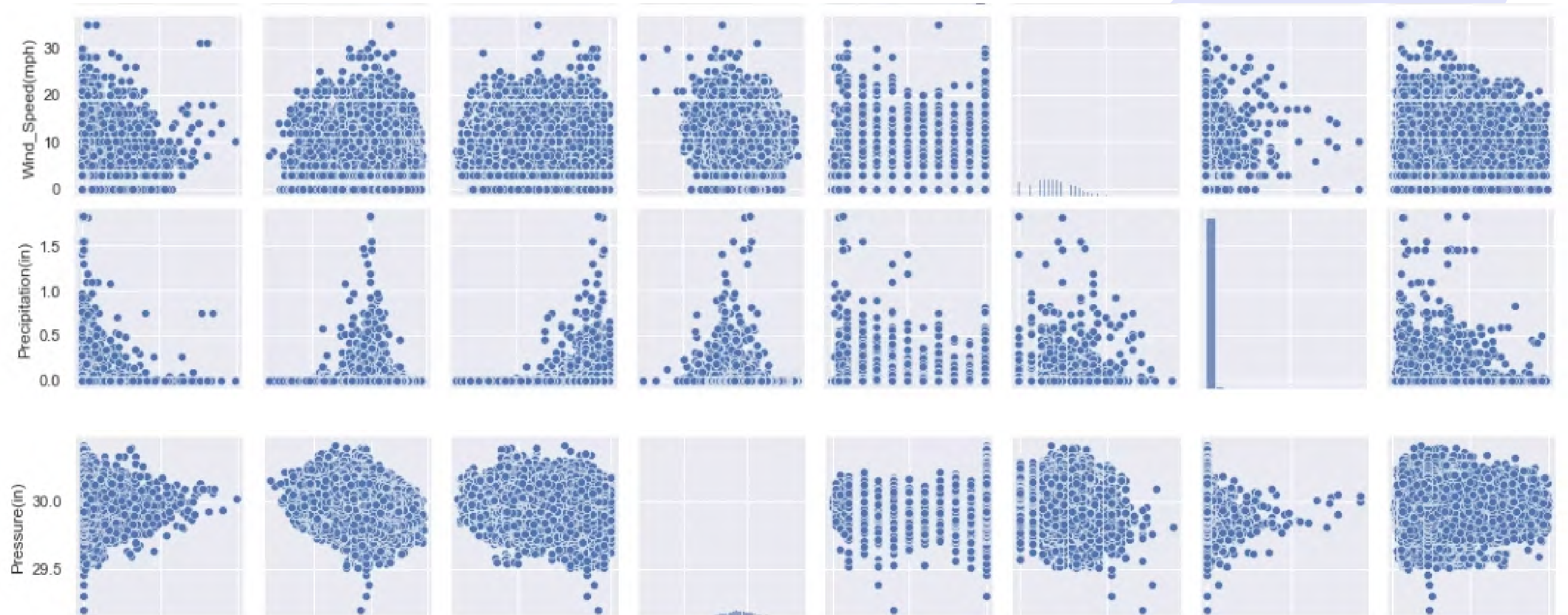
Backfill

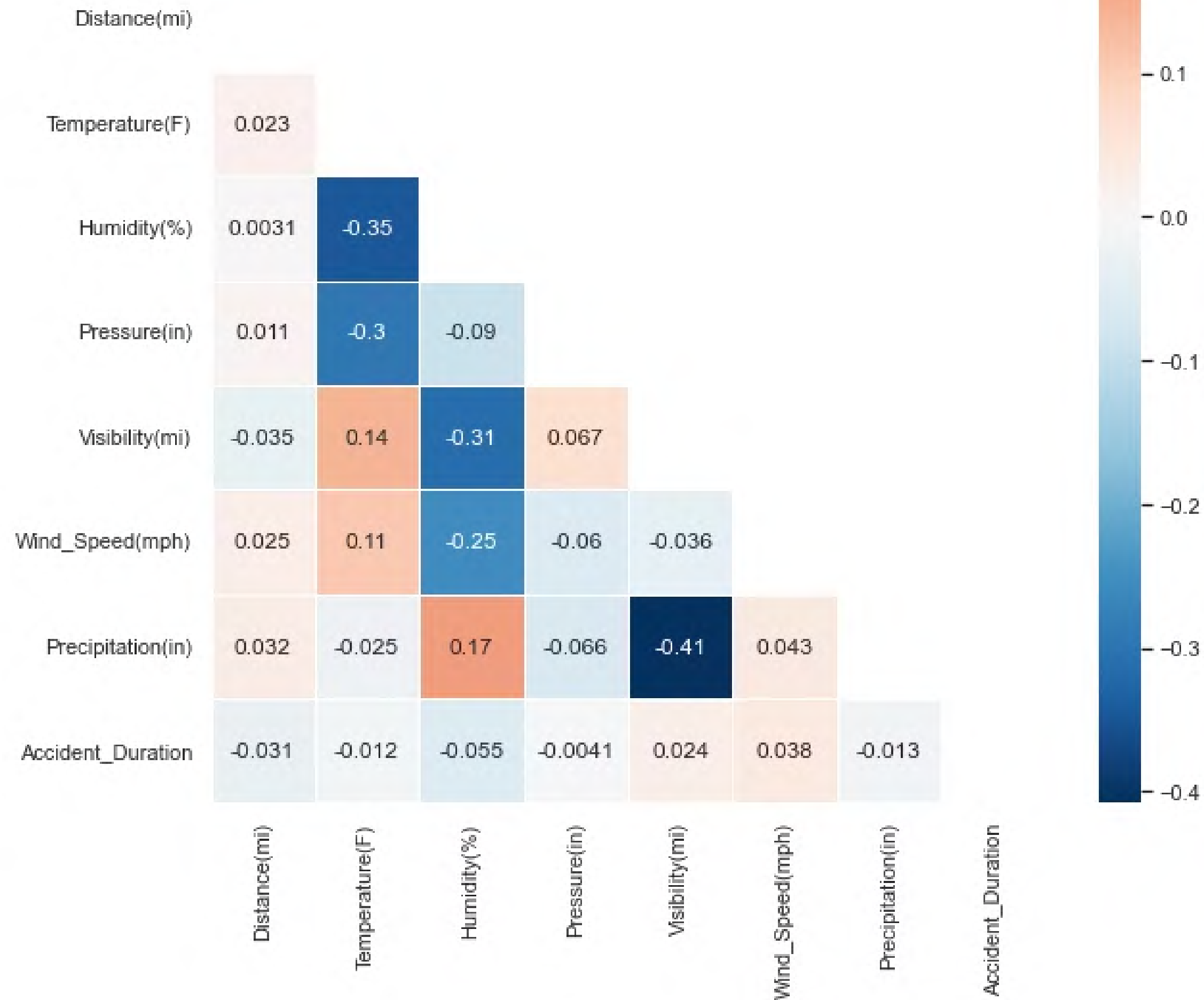


Drop Outliers

```
df = df.drop(df[df["Distance(mi)"]>10].index)
df = df.drop(df[df["Pressure(in)"]<29].index)
df = df.drop(df[df["Wind_Speed(mph)"]>40].index)
df = df.drop(df[df["Precipitation(in)"]>2].index)
```







Random Forest

```
rf_params = {'max_depth': randint(1,20), 'min_samples_split':  
rf = RandomForestRegressor()  
rf_random = RandomizedSearchCV(estimator=rf, param_distribut  
rf_random.fit(X_train_std, y_train)  
best_params = rf_random.best_params_  
best_rf = RandomForestRegressor(**best_params)  
best_rf.fit(X_train_std, y_train)
```

rmse: 2661.35

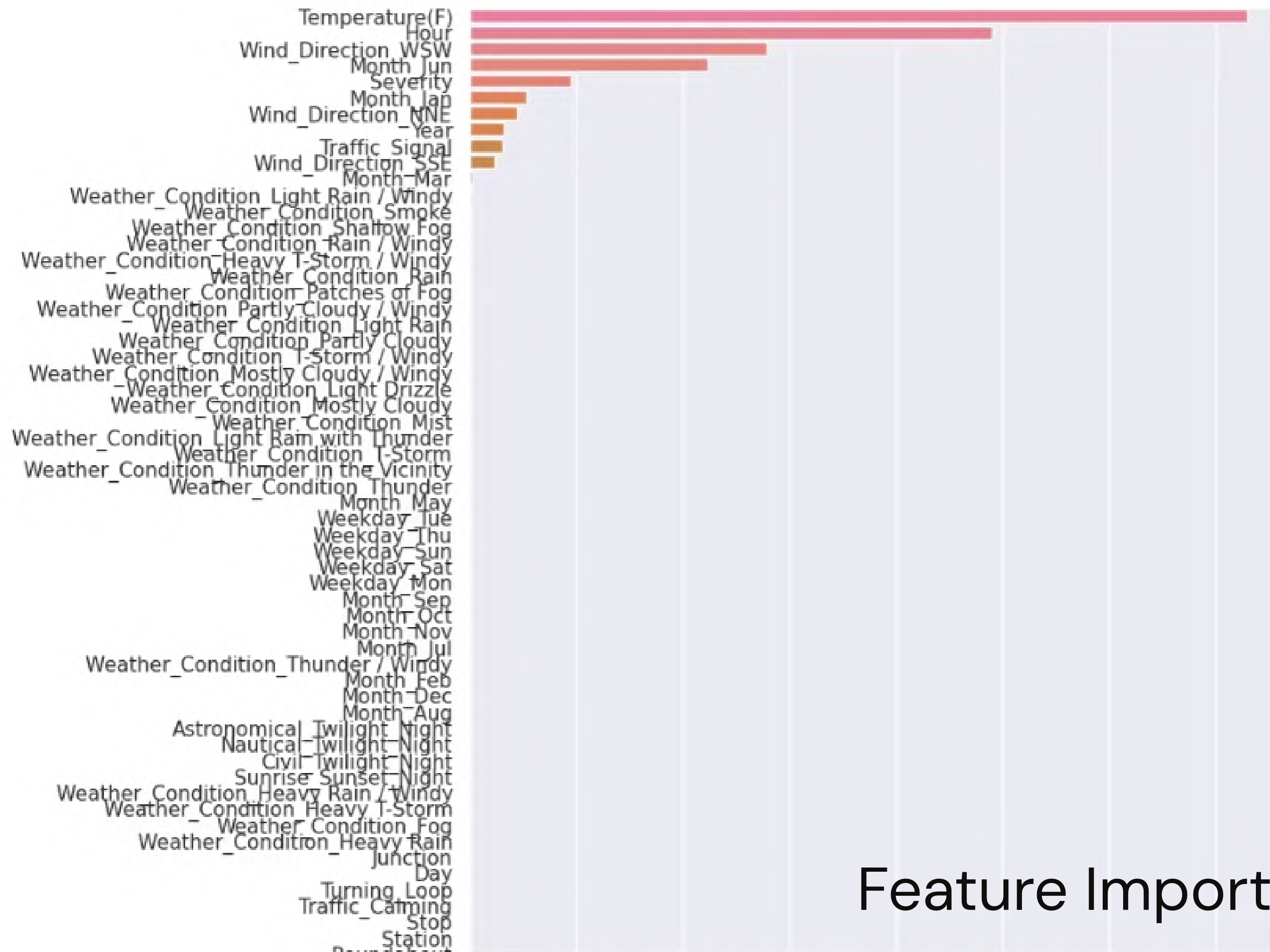
r2: 0.34

Neural Network

```
def build_nn_model():  
    model = Sequential()  
    model.add(Dense(128, activation='relu', input_shape=(X_train_std.shape[1],)))  
    model.add(Dense(64, activation='relu'))  
    model.add(Dense(1))  
    model.compile(optimizer='adam', loss='mean_squared_error')  
    return model
```

rmse: 2731.53

r2: 0.30



Feature Importance

Miami, Florida



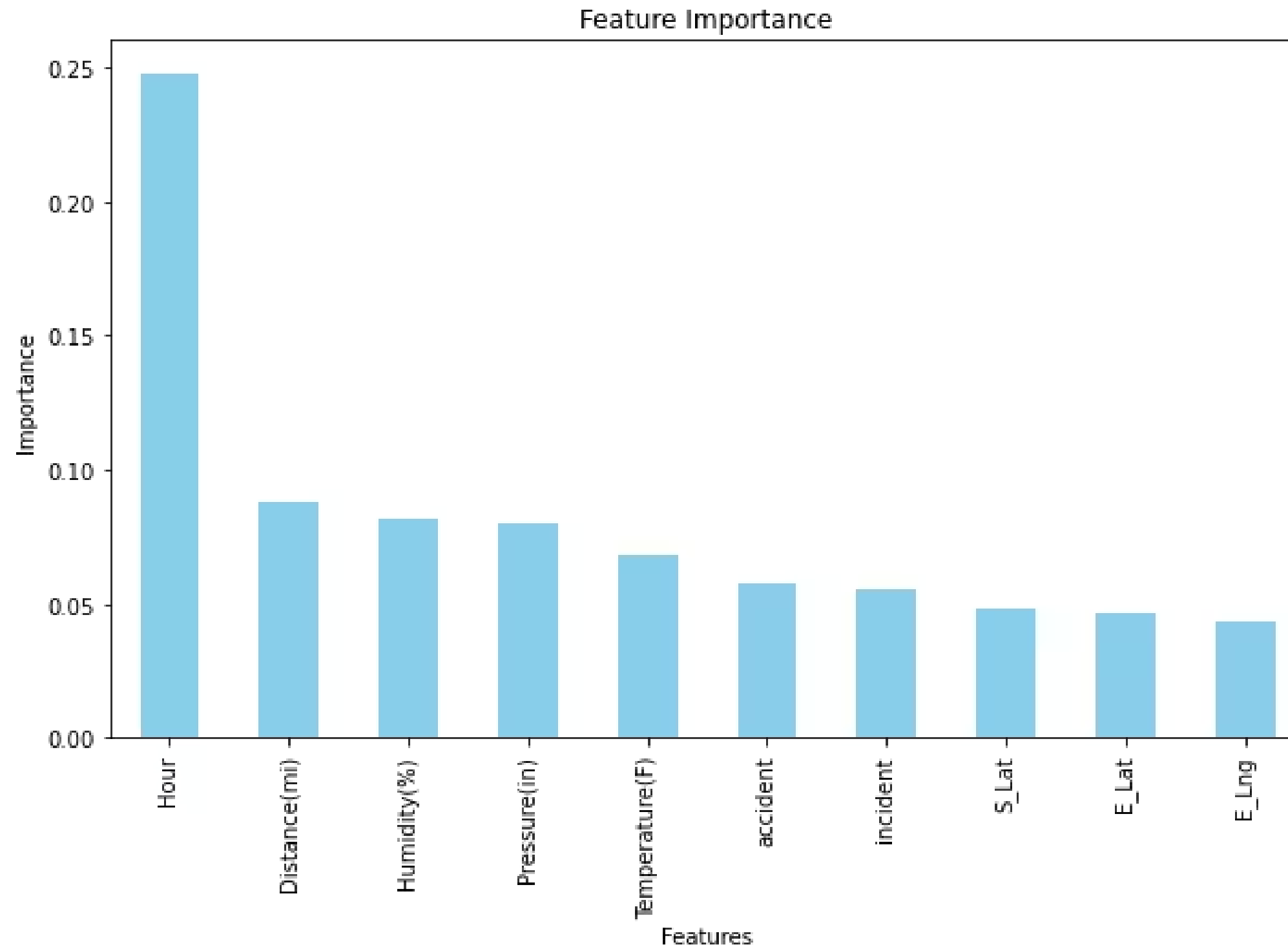
Most traffic accidents out of any city in the US
Past 2 years have seen 85,000+ reported traffic accidents
High tourism
Lots of congestion at peak hours

KENNETH APTE

Workflow

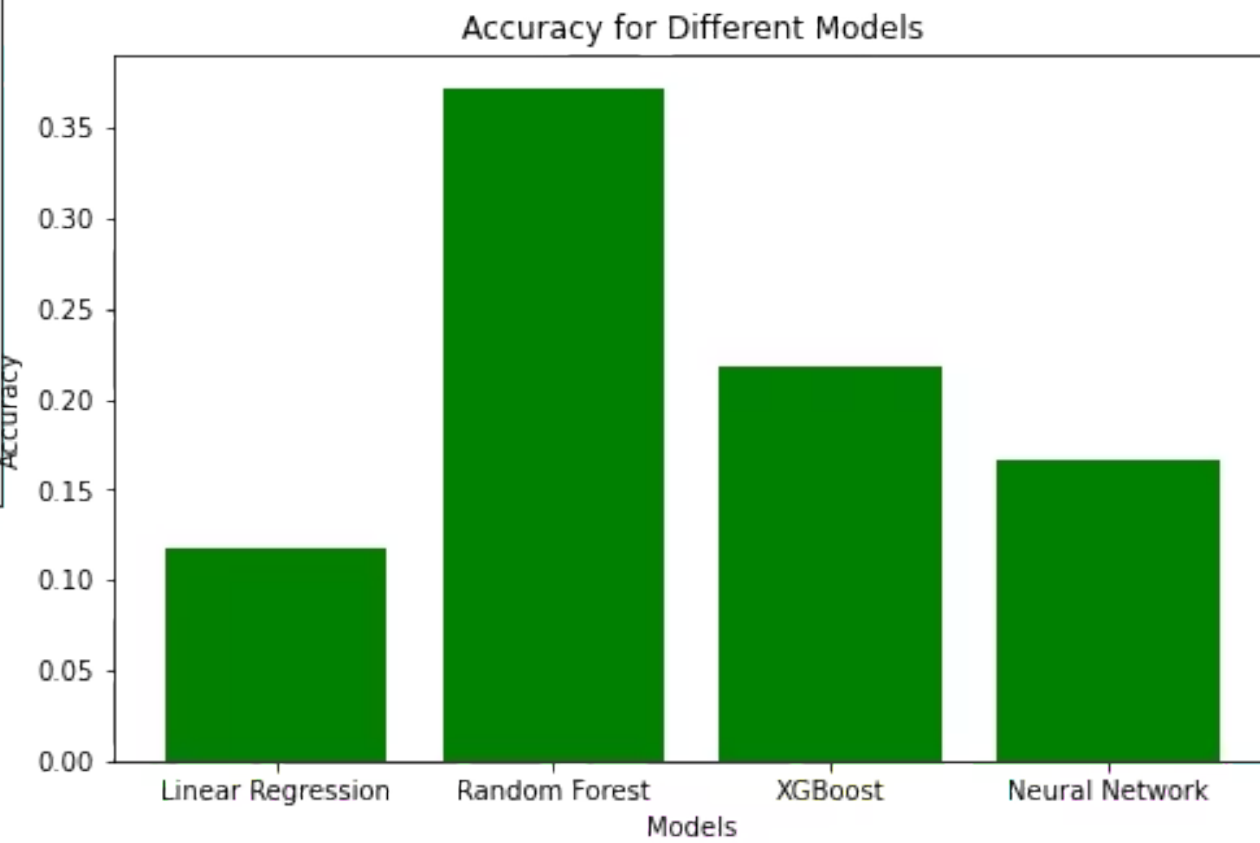
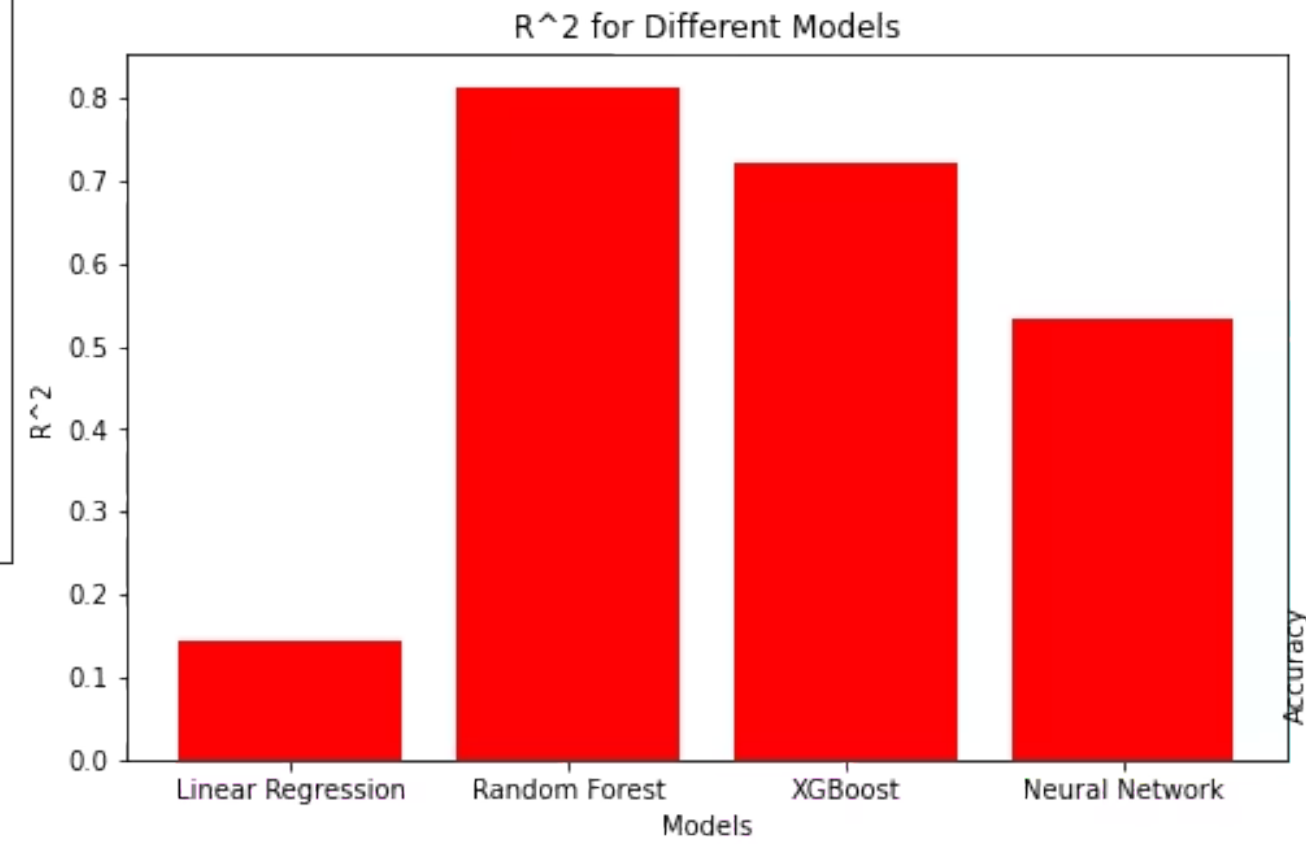
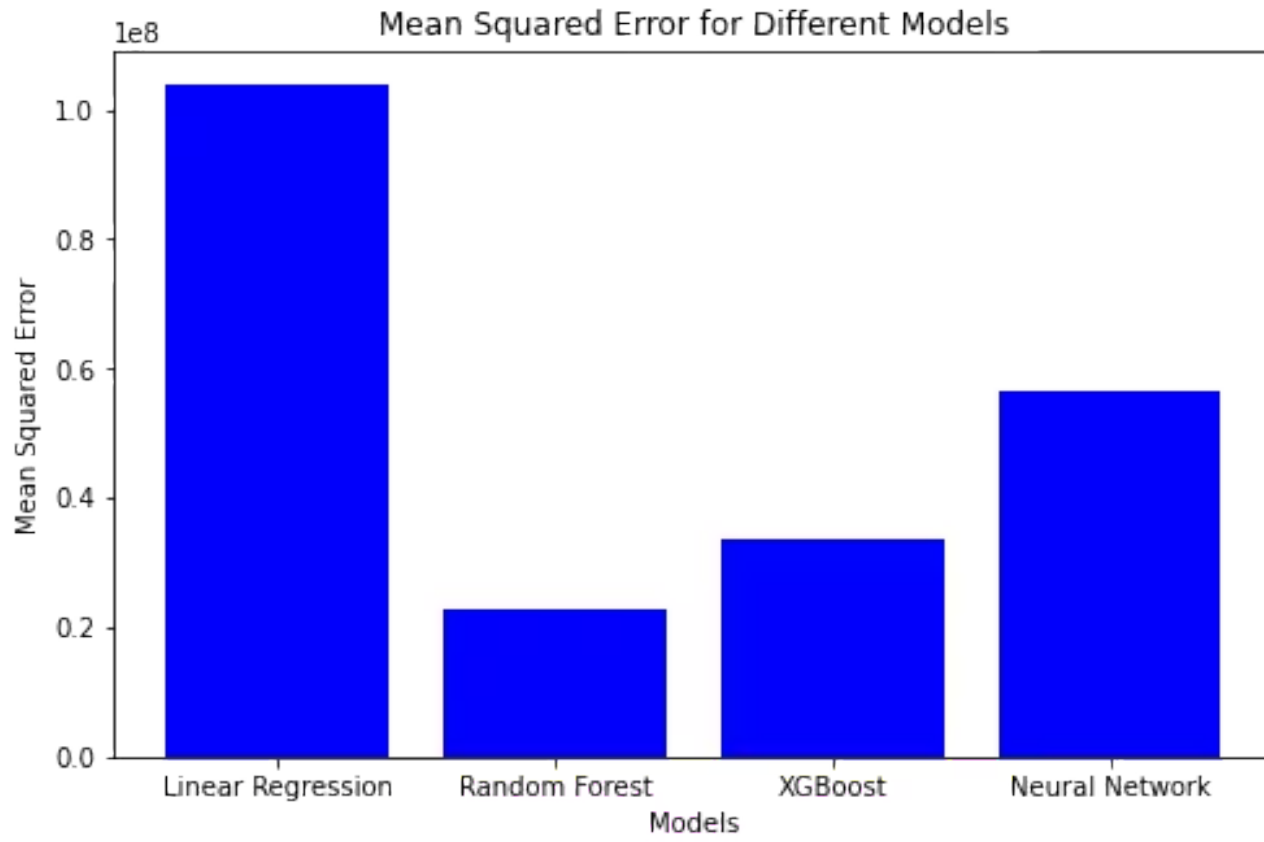
Procedure	Details
1. Exploratory Data Analysis	Understanding data, distributions, correlation coefficients with target variable
2. Wrangling/Cleaning	Removing NAs, nonsensical data, replacing outliers with 3+ z-score
3. Feature Engineering	Creating features from description text (Regex), discretization, creating day of week/hour features, one hot/binary encoding of non-numeric data, Lasso Regression and Decision Tree Regressors to identify salient variables/relative feature importance
4. Model Testing	Baseline linear regression, random forest, XGBoost, MLP regressor (neural network), mse, r square score, accuracy within 10%

Feature Importance



KENNETH APTE

Model Performance



KENNETH APTE

Real World Application of Models

- Using our models raw estimates of accident durations could be computed for different cities
- Useful for web mapping service creators such as Google and Apple to increase accuracy of arrival time
- Emergency Service organizations would be able to analyze whether certain areas have accident durations that are longer than expected
- Drivers, commuters, etc can be notified of accidents and projected accident duration to plan accordingly

General Limitations of Models

- Less than ideal model performance can be adjusted by training on a greater sample size of accidents
- Training/testing and hyperparameter tuning on more data would require much greater computational power and time which can be accomplished using distributed computing
- For most of the feature importance tables for Random Forests the distance feature dominated
- This can be adjusted by finding datasets with more detailed information geographic and driver information to add other important features



Want to make a presentation like this one?

Start with a fully customizable template, create a beautiful deck in minutes, then easily share it with anyone.

Create a presentation (It's free)