CAPSTONE PROJECT - CAR ACCIDENT REPORT

MARIA RENEE CARRASCO

September 2020

1. Introduction

1.1. Background

A car *accident* is an unplanned event that sometimes has inconvenient or undesirable consequences. There are too many situations that can cause an accident, some of them are our responsibility but others not.

1.2. Problem

The Seattle government is interested to develop an algorithm that could help to avoid car accidents, considering many variables that could affect the prediction like Weather, Road conditions and light conditions.

1.3. Objective

The principal objective of the project is try to define if there are a possibility and what is the level of probability to be involved in a car accident, taking in account some attributes that could add or rest reasons to change the circumstances.

2. Data acquisition

2.1. Data sources

Data is obtained from the Seattle police department and accident traffic record department from 2004.

First, it is necessary to understand the problem and the data which we are going to work. For that, all the attributes will be analyzed and prepared to construct a data set, considering missing data, type of data, correlations between data, etc. It is necessary to construct a supervised model and perform an evaluation to confirm the accuracy of it.

3. Exploratory Data Analysis

The data contains 37 attributes, where the independent variables are “WEATHER”, “ROADCOND” and “LIGHTCOND”, and the dependent variable is the “SEVERITY CODE”.

```
[10]: df['SEVERITYCODE'].value_counts().plot(kind= 'bar')

[10]: <AxesSubplot:>
```

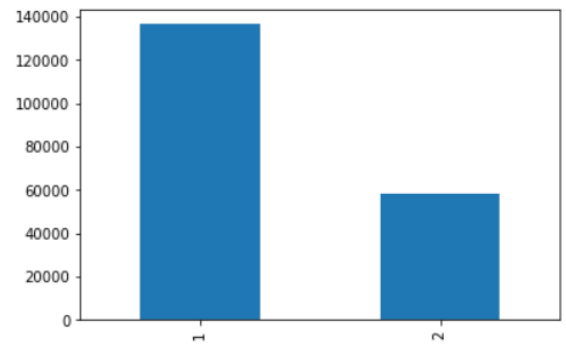


Fig.1

```
[8]: df['ROADCOND'].value_counts().plot(kind= 'bar')
```

```
[8]: <AxesSubplot:>
```

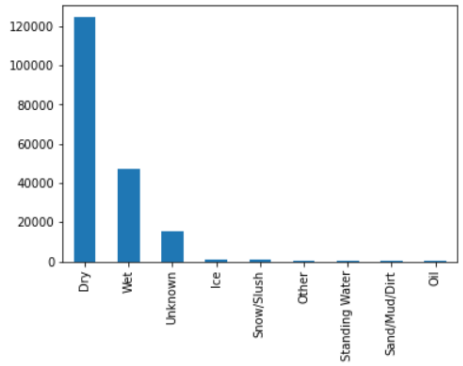


Fig.2

```
[9]: df['WEATHER'].value_counts().plot(kind= 'bar')
```

```
[9]: <AxesSubplot:>
```

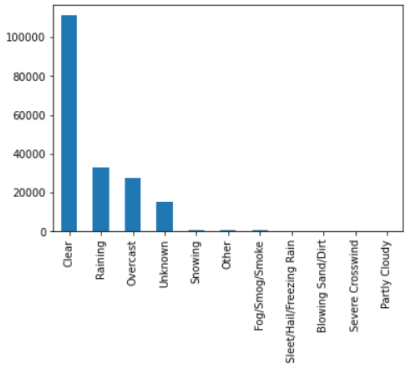


Fig.3

```
[11]: df['LIGHTCOND'].value_counts().plot(kind= 'bar')
```

```
[11]: <AxesSubplot:>
```

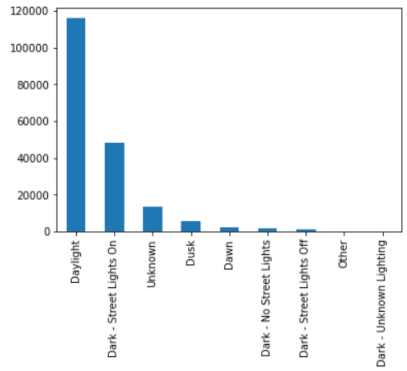


Fig.4

4. Methodology

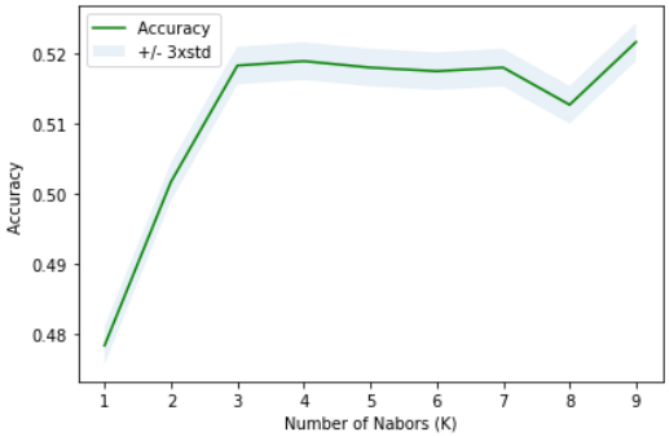
4.1. K-Nearest Neighbor (KNN)

Algorithm for supervised learning to solve both regression and classification problems. The value K (Knn_k) is the number of nearest neighbors to examine. It is necessary to increase the value of k, and see which k is the best for your model.

K-Nearest Neighbor

```
[29]: from sklearn.neighbors import KNeighborsClassifier
Knn_k = 25
#Train Model and Predict
neigh = KNeighborsClassifier(n_neighbors = Knn_k).fit(x_
neigh
Knn_yhat = neigh.predict(x_test)
Knn_yhat[0:5]
```

```
[29]: array([1, 1, 1, 2, 2])
```



4.1.1. KNN Evaluation

```
[33]: from sklearn.metrics import f1_score
print('KNN-F1Score',f1_score(Knn_y_test, Knn_yhat, avera
```

```
KNN-F1Score 0.5156463978535297
```

```
[34]: !pip install jaccard-index
```

```
Collecting jaccard-index
  Downloading https://files.pythonhosted.org/packages/e
7/66/a066229192ef1323b5a36bfc68a7d2e850227f96c0754349072
369470255/jaccard_index-0.0.3-py3-none-any.whl
Installing collected packages: jaccard-index
Successfully installed jaccard-index-0.0.3
```

```
[35]: from sklearn.metrics import jaccard_score
jaccard_score(Knn_y_test, Knn_yhat)
```

```
[35]: 0.3972503879045935
```

4.2. Decision tree

Decision trees are built are built using recursive partitioning to classify the data. For that, it is necessary to splitting the training set into distinct nodes, where one node contains all of most of one category of the data.

4.2.1. Decision Tree Evaluation

```
[40]: from sklearn.metrics import f1_score
      print('DT-F1Score',f1_score(Knn_y_test, Knn_yhat, averag
      DT-F1Score 0.4975537747091788

[41]: from sklearn.metrics import jaccard_score
      jaccard_score(Knn_y_test, Knn_yhat)

[41]: 0.3839371422476624
```

4.3. Logistic Regression

The dependent variable ("SEVERITYCODE") is finite or categorical. Logistic regression has been applied to understand the relationship between dependent variable ("SEVERITYCODE") and other attributes ("WEATHER", "ROADCOND", "LIGHTCOND"). In this way, it is possible to predict the car accident severity according to the variable selected to be analyzed.

4.3.1. Logistic Regression Evaluation

```
[45]: from sklearn.metrics import f1_score
      print('LR-F1Score',f1_score(Knn_y_test, LR_yhat, average
      LR-F1Score 0.5201486120944421

[46]: from sklearn.metrics import jaccard_score
      jaccard_score(Knn_y_test, LR_yhat)

[46]: 0.28388544278738287
```

5. Conclusion

Based on the data, it is possible to conclude that particular conditions have impact at a different scale that could result in property damage or injury. The following table shows that the model classification KNN is the best model to predict car accident.

MODEL	F1 SCORE	JACCARD SCORE	ACCURACY
KNN	0.52	0.40	0.51
Decision Tree	0.50	0.38	0.56
Logistic regression	0.52	0.28	0.53