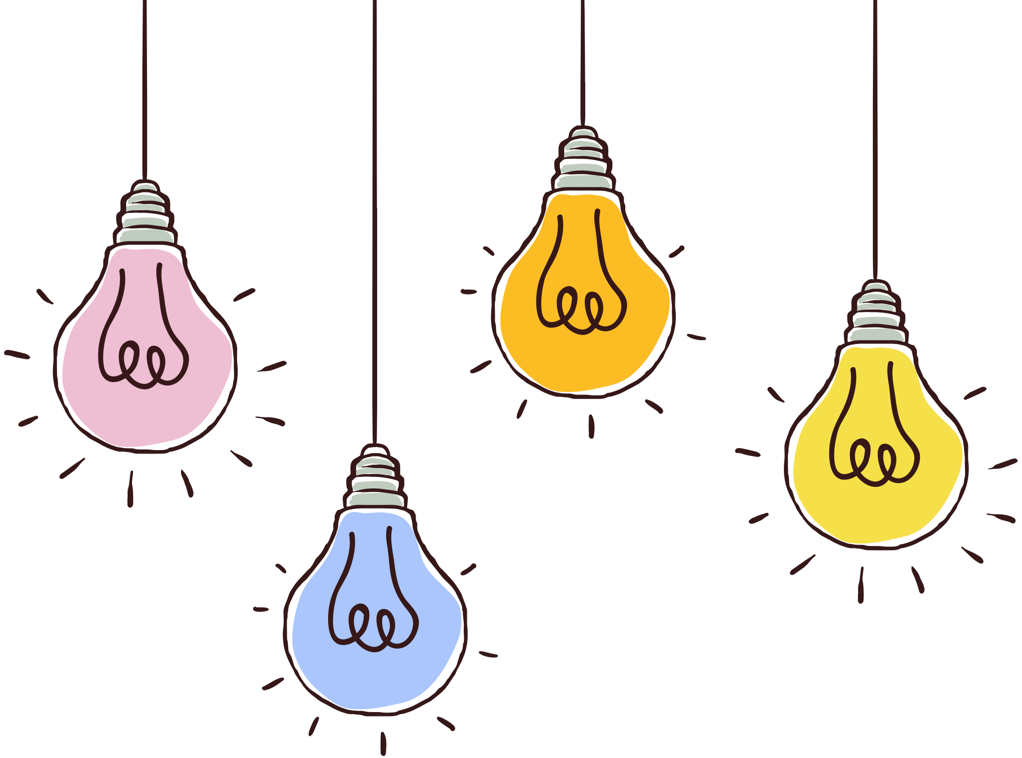


CAPSTONE PROJECT - CAR ACCIDENT REPORT

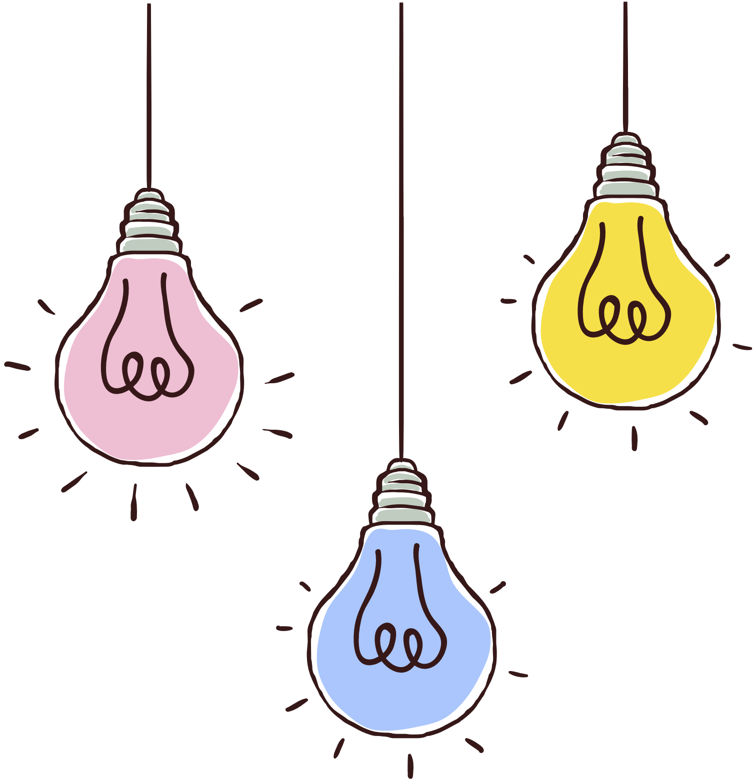
MARIA RENEE CARRASCO

INTRODUCTION

1. 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.

2. There are too many situations that can cause an accident, some of them are our responsibility but others not.

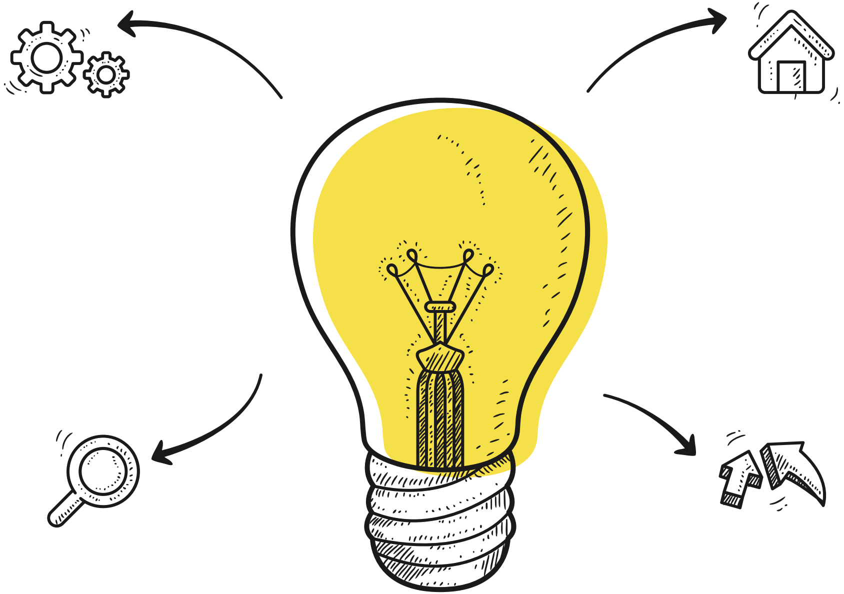
3. The Seattle government is interested to develop an algorithm that could help to avoid car accidents, considering many variables that could affect the prediction like Weather, Road conditions and light conditions.



EXPLORATORY DATA ANALYSIS

194,673 rows

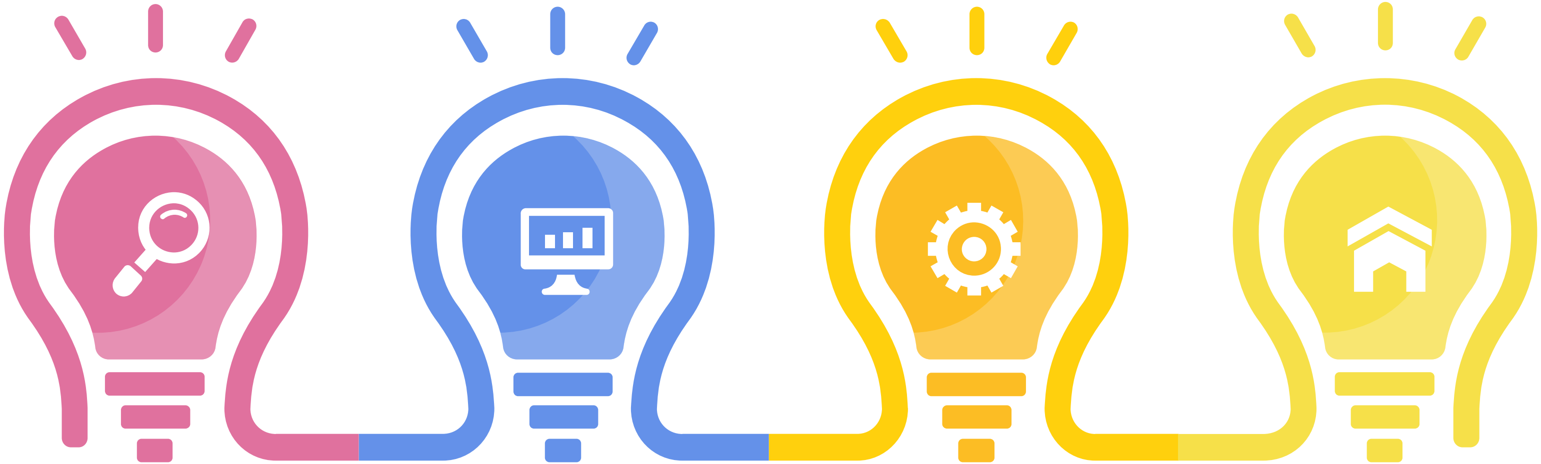
37 ATTRIBUTES



4 LEVELS OF SEVERITY

UNBALANCED DATA

ATRIBUTTES



SEVERITY CODE
Independent Variable

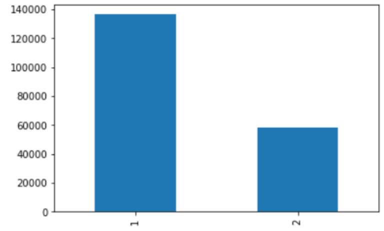
WEATHER
Dependent Variable

ROADCOND
Dependent Variable

LIGHTCOND
Dependent Variable

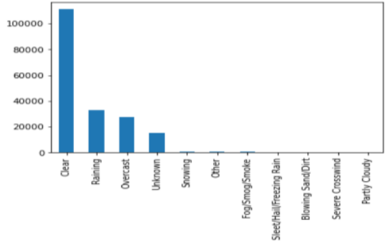
SEVERITYCODE

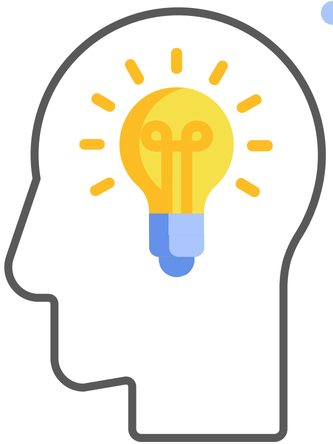
```
[10]: df['SEVERITYCODE'].value_counts().plot(kind= 'bar')
[10]: <AxesSubplot:>
```



WEATHER

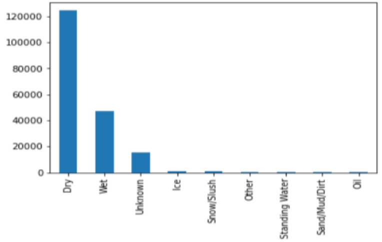
```
[9]: df['WEATHER'].value_counts().plot(kind= 'bar')
[9]: <AxesSubplot:>
```





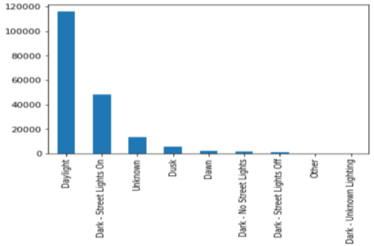
ROADCOND

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

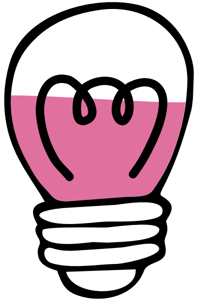


LIGHTCOND

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

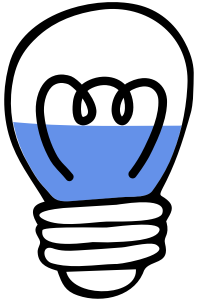


METHODOLOGY



01

K-NEAREST NEIGHBOR KNN



02

DECISIÓN TREE

03

LOGISTIC REGRESSION

01

K-NEAREST NEIGHBOR (KNN)

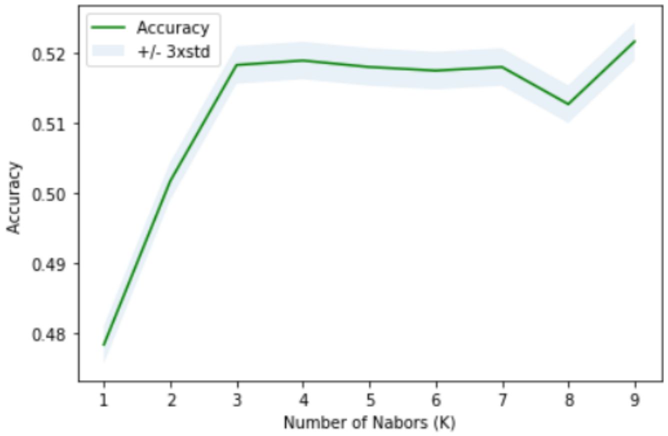
K-Nearest Neighbor

[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])
```



01

K-NEAREST NEIGHBOR (KNN)

[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/e7/66/a066229192ef1323b5a36bfc68a7d2e850227f96c0754349072369470255/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
```

02 DECISIÓN 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.

```
[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
```

03 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.

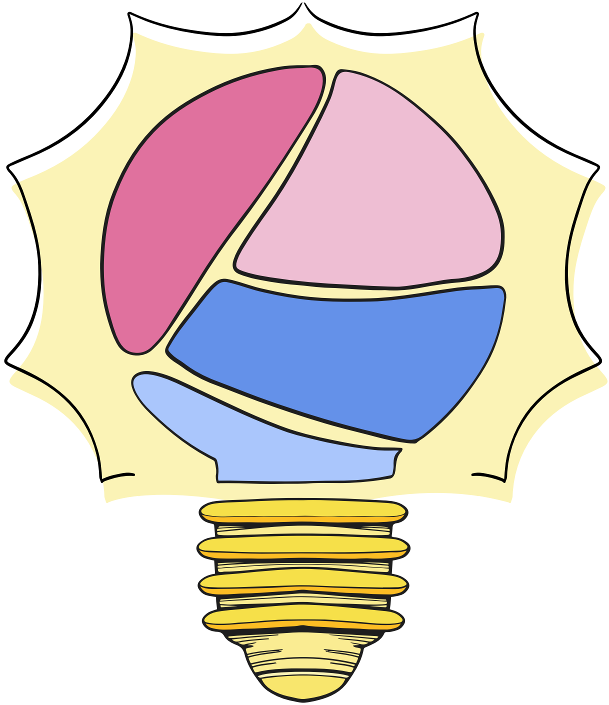
```
[45]: from sklearn.metrics import f1_score
      print('LR-F1Score',f1_score(Knn_y_test, LR_yhat, average=

      LR-F1Score 0.5201486120944421
```

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

[46]: 0.28388544278738287
```

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