Predict severity of Accident using Seattle accident data

**INTRODUCTION:**

The Seattle government is going to prevent avoidable car accidents by employing methods that alert drivers, health system, and police to remind them to be more careful in critical situations.

The main reasons for car accidents are:-

*1] Distracted Driving.*

*2] Speeding.*

*3] Drunk Driving.*

*4] Reckless Driving.*

*5] Rain etc.*

Whilst some of these factors are human dependent, others can be predicted by revealing hidden patterns within data. We can provide warnings according to the patterns we find and thus attempt to save as much lives as possible.

The target audience of the project is local Seattle government, police, rescue groups, and last but not least, car insurance institutes. The model and its results are going to provide some advice for the target audience to make insightful decisions for reducing the number of accidents and injuries for the city.

**DATA**:

The data was collected by the Seattle Police Department and Accident Traffic Records Department.

The data consists of 37 independent variables and 194,673 rows. The dependent variable, “SEVERITYCODE”, contains numbers that correspond to different levels of severity caused by an accident from 0 to 4.

Severity codes are as follows:

*0: Little to no Probability (Clear Conditions)*

*1: Very Low Probability — Chance or Property Damage*

*2: Low Probability — Chance of Injury*

*3: Mild Probability — Chance of Serious Injury*

*4: High Probability — Chance of Fatality*

The NULL values in the data need to be processed before performing any operations on it.

**DATA PREPROCESSING:**

The dataset in the original form is not ready for data analysis. In order to prepare the data, first, we need to drop the non-relevant columns. In addition, most of the features are of object data types that need to be converted into numerical data types.

After analysing the data set, I have decided to focus on only four features, severity, weather conditions, road conditions, and light conditions, among others.

To get a good understanding of the dataset, I have checked different values in the features. The results show, the target feature is imbalance, so we use a simple statistical technique to balance it.

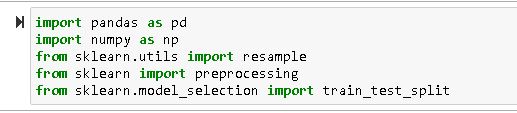
**Methodology:**

For implementing the solution, I have used GitHub as a repository and running Jupyter Notebook to pre-process data and build Machine Learning models. Regarding coding, I have used Python and its popular packages such as Pandas, NumPy and Sklearn.

Once I have load data into Pandas Dataframe, used ‘dtypes’ attribute to check the feature names and their data types. Then I have selected the most important features to predict the severity of accidents in Seattle. Among all the features, the following features have the most influence in the accuracy of the predictions:

* “WEATHER”,
* “ROADCOND”,
* “LIGHTCOND”

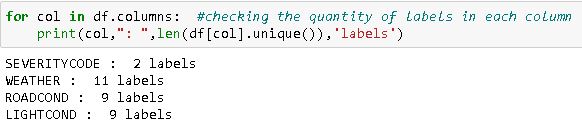
Step 1: Importing the libraries.

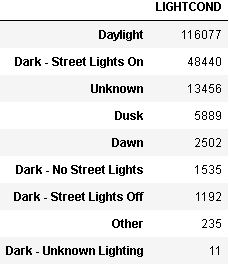
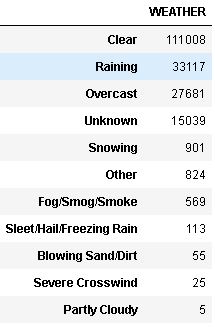


Step 2: Selecting specific columns from the dataset.

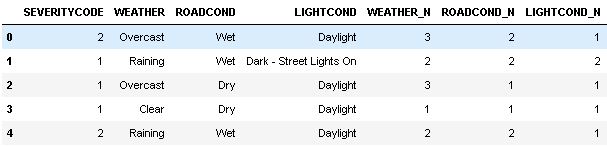


Step 3: Getting the info about the columns in the dataset.

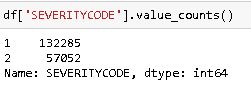




Step 4: Applying filters and adding columns to dataset.

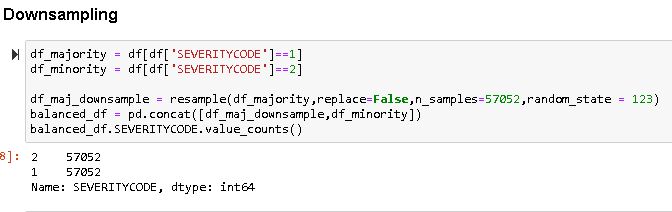


Step 5: Checking the sample size of the dataset.

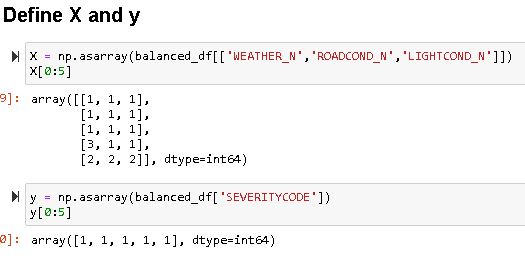


As we can see that this would require down sampling for normalisation of results.

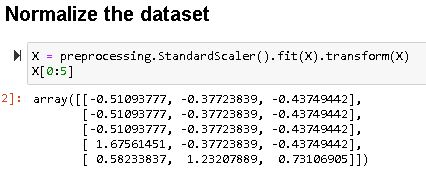
Step 6: Down-Sampling:



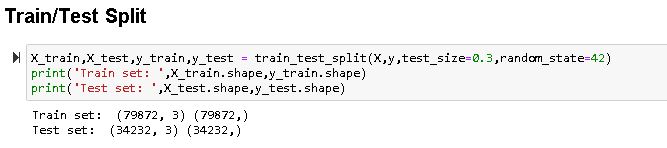
Step 7: Defining X and Y.



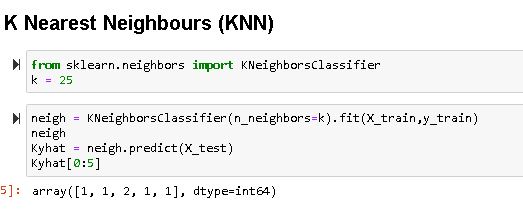
Step 8: Normalise the dataset.



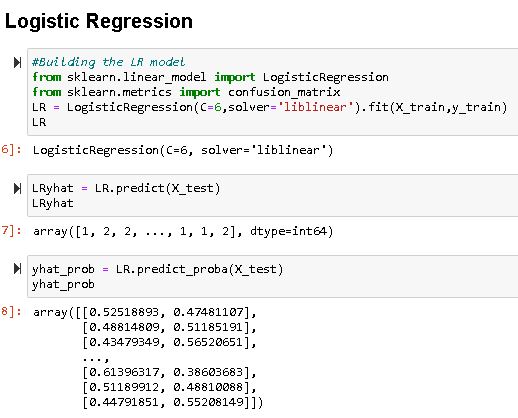
Step 9: Dividing the data into train and test parts.



Step 10: Applying KNN.

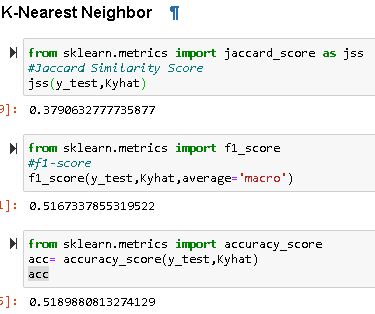


Step 11: Applying Logistic regression.

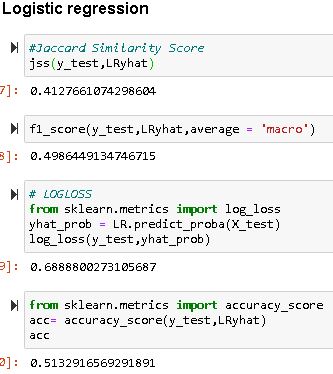


Step 12: Checking the accuracy of the models.

For KNN.



For Logistic Regression.



Conclusion:

It appears that both the models have given close results but the accuracy of KNN is more than that of Logistic Regression by a small extent.