DATA

Research Question

Predicting the occurrences of vehicular crashes on roadways of the State of Seattle based on Seattle Department of Transportation, provided data for all collisions and crashes that have occurred in the state from 2004 onwards.

DATA Source

The dataset is available as comma-separated values (CSV) files, KML files, and ESRI shapefiles that can be downloaded from the Seattle Open Geo-Data Portal

Link- https://opendata.arcgis.com/datasets/5b5c745e0f1f48e7a53acec63a0022ab_0.csv

download the dataset to my project directory and look at the data types and the dimensionality of the data. We can see that the dataset contains 221,389 records and 40 fields.

The metadata of the dataset can be found from the website of the Seattle Department of Transportation. On reading the dataset summary, we can determine the description of each of the fields and their possible values.

The data contains several categorical fields and corresponding descriptions which could help us in further analysis, attempt at understanding the data in terms of the fields that was considered for later stages of model building

```
data = pd.read_csv("data.csv")
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 221389 entries, 0 to 221388
Data columns (total 40 columns):
 #   Column           Non-Null Count   Dtype  
---  ------           --------------   -----  
 0   X                213918 non-null  float64
 1   Y                213918 non-null  float64
 2   OBJECTID         221389 non-null  int64  
 3   INCKEY           221389 non-null  int64  
 4   COLDETKEY        221389 non-null  int64  
 5   REPORTNO         221389 non-null  object 
 6   STATUS           221389 non-null  object 
 7   ADDRTYPE         217677 non-null  object 
 8   INTKEY           71884 non-null   float64
 9   LOCATION         216801 non-null  object 
 10  EXCEPTRSNCODE    100986 non-null  object 
 11  EXCEPTRSNDESC    11779 non-null   object 
 12  SEVERITYCODE     221388 non-null  object 
 13  SEVERITYDESC     221389 non-null  object 
 14  INATTENTIONIND   30188 non-null   object
 28  UNDERINFL        195179 non-null  object
 29  WEATHER          194969 non-null  object
 30  ROADCOND         195050 non-null  object
 31  LIGHTCOND        194880 non-null  object
```

The WEATHER field contains a description of the weather conditions during the time of the collision.

The SEVERITYCODE field contains a code that corresponds to the severity of the collision. and SEVERITYDESC contains a detailed description of the severity of the collision.

From the data we can conclude that there were 349 collisions that resulted in at least one fatality, and 3,102 collisions that resulted in serious injuries. The following table lists the meaning of each of the codes used in the SEVERITYCODE field:

SEVERITYCODE Value	Meaning
1	Accidents resulting in property damage
2	Accidents resulting in injuries
2b	Accidents resulting in serious injuries
3	Accidents resulting in fatalities
0	Data Unavailable i.e. Blanks

DATA Cleaning/Preparation

The data collected is real world data and contained missing values. The missing values were encoded in a number of different ways, such as ‘Unknown’, ‘N/A’, ‘Not Reported’, or ’’.

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 analyzing 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.

As the dataset has possibly been sourced from a database table, several unique identifiers and spatial features are present in the database which may be irrelevant in further statistical analysis. These fields are OBJECTID, INCKEY, COLDETKEY, INTKEY, SEGLANEKEY, CROSSWALKKEY, and REPORTNO. Other fields suchs as EXCEPTRSNCODE, SDOT_COLCODE, SDOTCOLNUM and LOCATION and their corresponding descriptions (if any) are categorical but have a large number of distinct values that shall not be that much useful for analysis. The INCDATE and INCDTTM denote the date and the time of the incident but may not be of use in further analyses. The data needs to be pre-processed.

Methodology

For implementing the solution, I have used Github as a repository and running Jupyter Notebook to preprocess data and build Machine Learning models. Regarding coding, I have used Python and its popular packages such as Pandas, NumPy and Sklearn and also user Teansor flow to check which prediction is best

Once I have load data into Pandas Data frame, 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”

Also, as I mentioned earlier, “SEVERITYCODE” is the target variable.

I have run a value count on road (‘ROADCOND’) and weather condition (‘WEATHER’) to get ideas of the different road and weather conditions. I also have run a value count on light condition (’LIGHTCOND’), to see the breakdowns of accidents occurring during the different light conditions.

Checking for blanks and duplicated records.
data.isna().sum()- Checking for blanks in the data
data.duplicated().sum()- Finding Duplicates if any in the dataset

Selecting relevant fields and dropping others. Example- WEATHER', 'ROADCOND', 'LIGHTCOND', 'SPEEDING', 'SEVERITYCODE', 'UNDERINFL', 'SERIOUSINJURIES', 'FATALITIES', 'INJURIES', 'PERSONCOUNT', 'PEDCOUNT', 'PEDCYLCOUNT', 'VEHCOUNT'

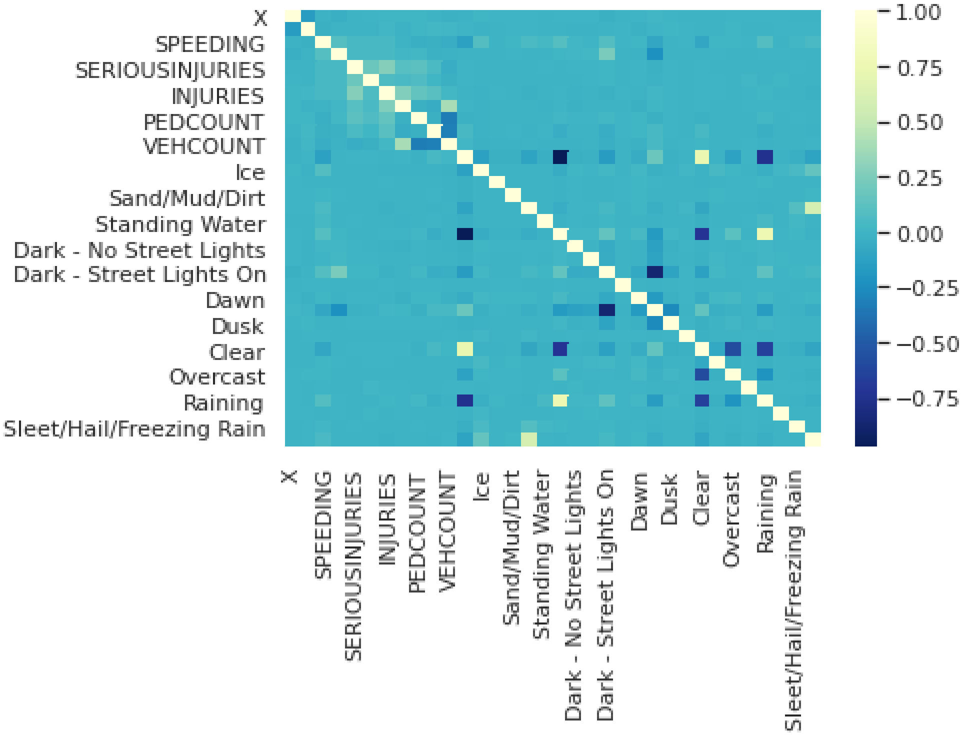
```
#   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   X                 213918 non-null  float64
 1   Y                 213918 non-null  float64
```

2	WEATHER	194969	non-null	object
3	ROADCOND	195050	non-null	object
4	LIGHTCOND	194880	non-null	object
5	SPEEDING	9928	non-null	object
6	SEVERITYCODE	221388	non-null	object
7	UNDERINFL	195179	non-null	object
8	SERIOUSINJURIES	221389	non-null	int64
9	FATALITIES	221389	non-null	int64
10	INJURIES	221389	non-null	int64
11	PERSONCOUNT	221389	non-null	int64
Sample data				

Fixing the SPEEDING field by encoding it to 0 for the blanks and 1 for the Y values.

Records containing values as **Unknown** and **Other** can be considered as null values. Severity Code of 0 corresponds to unknown severity, which can also be treated as null.

Finding the correlation among the features of the dataset helps understand the data better. For example, in the heatmap shown below, it can be observed that some features have a strong positive / negative correlation while most of them have weak / no correlation.



The datasets x and y are constructed. The set x contains all the training examples and y contains all the labels. Feature scaling of data is done to normalize the data in a dataset to a specific range.

After normalization, they are split into x_train, y_train, x_test, and y_test. The first two sets will be used for training and the last two shall be used for testing. Upon choosing a suitable split ratio, 80% of data is used for training and 20% of is used for testing.

I have employed three machine learning models:

Decision Tree

Decision Tree makes decision with tree-like model. It splits the sample into two or more homogenous sets (leaves) based on the most significant differentiators in the input variables. To choose a differentiator (predictor), the algorithm considers all features and does a binary split on them (for categorical data, split by category; for continuous, pick a cut-off threshold). It will then choose the one with the least cost (i.e. highest accuracy), and repeats recursively, until it successfully splits the data in all leaves (or reaches the maximum depth).

Information gain for a decision tree classifier can be calculated either using the Gini Index measure or the Entropy measure, whichever gives a greater gain. A hyper parameter Decision Tree Classifier was used to decide which tree to use, DTC using entropy had greater information gain; hence it was used for this classification problem.

```
              precision    recall  f1-score   support
           1       1.00      1.00      1.00     22543
           2       1.00      1.00      1.00     11090
          2b       1.00      1.00      1.00       589
           3       1.00      1.00      1.00        54

    accuracy                           1.00     34276
   macro avg       1.00      1.00      1.00     34276
weighted avg       1.00      1.00      1.00     34276
```

RFC- Random Forest Classifier

Random Forest Classifier is an ensemble (algorithms which combines more than one algorithms of same or different kind for classifying objects) tree-based learning algorithm. RFC is a set of decision trees from randomly selected subset of training set. It aggregates the votes from different decision trees to decide the final class of the test object. Used for both classification and regression.

Similar to DTC, RFT requires an input that specifies a measure that is to be used for classification, along with that a value for the number of estimators (number of decision trees) is required. A hyper parameter RFT was used to determine the best choices for the above mentioned parameters. RFT with 75 DT’s using entropy as the measure gave the best accuracy when trained and tested on pre-processed accident severity dataset.

```
              precision    recall  f1-score   support
           1       1.00      1.00      1.00     22543
           2       1.00      1.00      1.00     11090
          2b       1.00      1.00      1.00       589
           3       1.00      0.98      0.99        54
```

```
    accuracy                           1.00     34276
   macro avg       1.00      1.00      1.00     34276
```

Logistic Regression

Logistic Regression is a classifier that estimates discrete values (binary values like 0/1, yes/no, true/false) based on a given set of an independent variables. It basically predicts the probability of occurrence of an event by fitting data to a logistic function. Hence it is also known as logistic regression. The values obtained would always lie within 0 and 1 since it predicts the probability.

```
              precision    recall   f1-score   support
           1       1.00      1.00      1.00     22543
           2       1.00      1.00      1.00     11090
          2b       1.00      0.98      0.99       589
           3       1.00      1.00      1.00        54

    accuracy                           1.00     34276
   macro avg       1.00      1.00      1.00     34276
weighted avg       1.00      1.00      1.00     34276
```

Result

The accuracies of all models was 100% which means we can accurately predict the severity of an accident. A bar plot is plotted below with the bars representing the accuracy of each model.