**1. Introduction**

**1.1 Background**

Seattle is a seaport city on the West Coast of the United States. It is the seat of King County, Washington. Seattle is the largest city in both the state of Washington and the Pacific Northwest region of North America. According to U.S. Census data, the Seattle metropolitan area's population stands at 3.98 million, making it the 15th-largest in the United States. In July 2013, Seattle was the fastest-growing major city in the United States. The overall number of Seattle's car population is 435,000. That's more than 5,000 cars per square mile or 637 cars for every 1,000 residents. Crunching census data from between 2010 and 2015, it is reported that Seattle's population grew by 12 percent, the same increase as the number of personal vehicles owned by Seattle residents. With the increasing number of cars, the chances of accidents also rise. Around 1.35 million people die annually in traffic accidents globally, an average of 3,700 people risk their lives on the highways every day, and a further 20-50 million suffer non-fatal injuries, frequently resulting in long-term disability.

**1.2 Problem**

Road traffic collisions are a leading cause of death for many people in the United States and the leading cause of unnatural death for stable U.S. residents living or traveling abroad. In 2017, Seattle police reported 10,959 motor vehicle collisions on city streets. According to the report, in 2017, there were 187 fatal and severe injury collisions on Seattle streets. Data available from the Washington State Department of Transportation (WSDOT) reflect an even worse tally in 2018, with 212 crashes that resulted in severe injury or wrongful death. This project aims to forecast how the magnitude of accidents can be reduced, depending on a few factors.

**1.3 Interest**

The Seattle Public Development Authority, which aims to enhance these road factors, and the automobile drivers themselves, who can take steps to decrease the seriousness of injuries, can benefit from reducing the severity of injuries.

**2. Data acquisition and cleaning**

**2.1 Data Source**

The dataset used for this project is focused on car accidents that took place in Seattle city. The severity of each traffic crash, along with the time and circumstances in which each accident happened, is the details pertaining to car crashes. The link to the dataset is mentioned in the reference.

**2.2 Data Cleaning**

The dataset is imported to the notebook using the pandas library. The size of the dataset was calculated using the shape function. The dataset has a total of 1,94,673 observations for every feature. There a total of 38 feature columns present in the dataset. On exploring, it was found that the dataset has a lot of missing values.

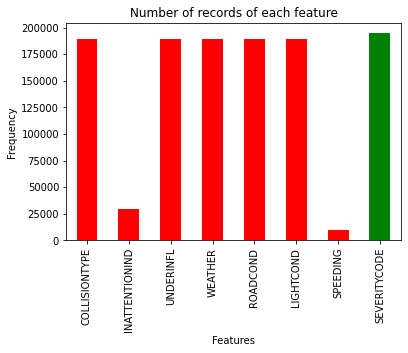


Fig: Data before cleaning

The aim is to make predictions of the severity of the accident using the dataset's features. The main predictor feature is categorized as 1 and 2 where 1 represents Property Damage, and 2 illustrates Collision Injury. Later, labels 1 and 0 were assigned to columns where Y and N were present. These columns include Inattention, Speeding and Under the influence. Furthermore, labels were assigned to rest of the features as most of the dataset contains categorical features. For Collision type, Parked Car is assigned 0, Angles as 1, Rear Ended as 2, Sideswipe as 3, Left Turn as 4, Pedestrian as 5, Cycles as 6, Right Turn as 7, and Head On as 8. For light condition, Light is given as 0, Medium as 1 and Dark as 2. For Road Condition, Dry is assigned 0, Wet is assigned 1, and Slush was given 2. Similarly, for Weather Condition, 0 is Clear, Rain and Snow as 1, Overcast and Cloudy as 2, and Windy as 3. Apart from these values, there were unique values for every variable: ‘Other’ or ‘Unknown’. Deleting these values would have caused a lot of data loss, so to deal with this issue data imputation method was carried out, assigning random values in place of these values.

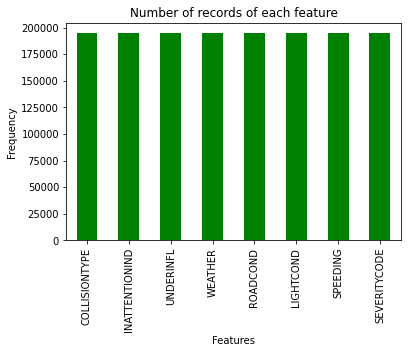
****

Fig: Data after cleaning

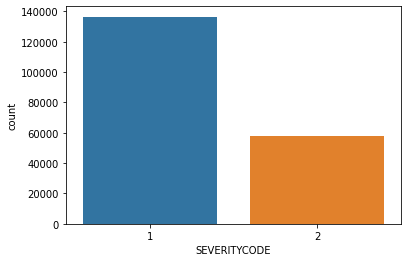
**2.3 Feature Selection**

After data cleaning, there were 1,94,673 samples and 38 features in the data. Upon examining each feature's meaning, it was clear that there was some redundancy in the features. For example, severity description, Object Id, Incident date, location, and other features after careful observation, it is conclusive that these features won't help make predictions. A total of 8 features were selected, including the target feature that is SEVERITYCODE.

|  |  |
| --- | --- |
| **Feature** | **Description** |
| COLLISIONTYPE | Type of Collision |
| INATTENTIONIND | Whether or not collision was due to inattention.  (Y/N) |
| UNDERINFL | Whether or not a driver involved was under the  influence of drugs or alcohol. |
| WEATHER | A description of the weather conditions during  the time of the collision. |
| ROADCOND | The condition of the road during the collision. |
| LIGHTCOND | The light conditions during the collision. |
| SPEEDING | Whether or not speeding was a factor in the  collision. (Y/N) |
| SEVERITYCODE | A code that corresponds to the severity of the  collision:   * 1 - Property Damage * 2 - Injury |

**3 Exploratory Data Analysis**

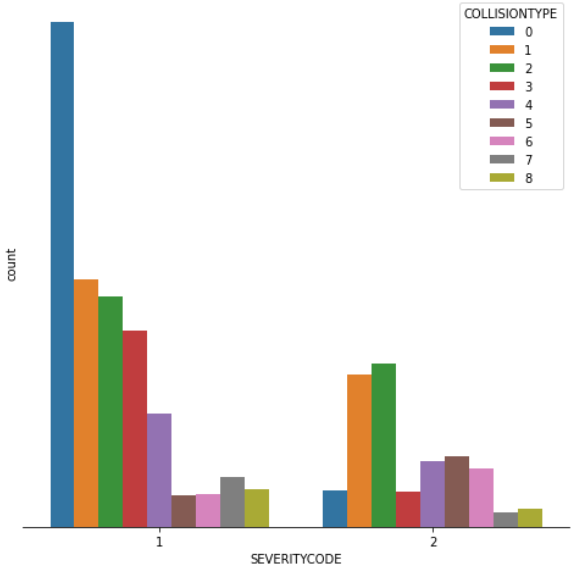
Exploratory analysis helps us understand more about the data. We’ve selected in total 8 features for our model. All of the selected features are categorical variables, so it is easy to represent them in a bar chart.



Looking at the above figure it is quite conclusive that the dataset is an unbalanced dataset where the distribution of the target variable is in almost 2:1 ratio in favour of property damage. To make better predictions it is recommended to have a balanced dataset. Synthetic Minority Oversampling Technique (SMOTE) is used to balance the dataset. SMOTE is used from the ‘imblearn’ library, it uses the oversampling method in order to balance the target variable in equal proportions in order to have an unbiased classification model. The below shown figures represent distribution of severity code against each feature.

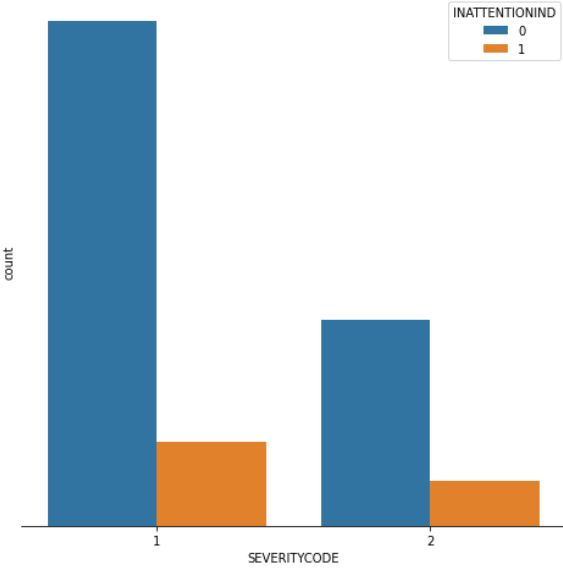
**3.1 Distribution of Collision Type against Severity Code**

From the following figure, it is observable that for the severity code 1 collision type 0 where collision type 0 represents ‘Parked Car’ is the most prominent. It is obvious that when a parked car has a collision incident then the one thing that for sure happens is the damage of property. For severity code 2 collision type 2 is the highest where collision type 2 represents ‘Rear Ended’. It is seen that most of the injury take place when a car collides with another car from behind as most of the time the driver is unaware of the situation.



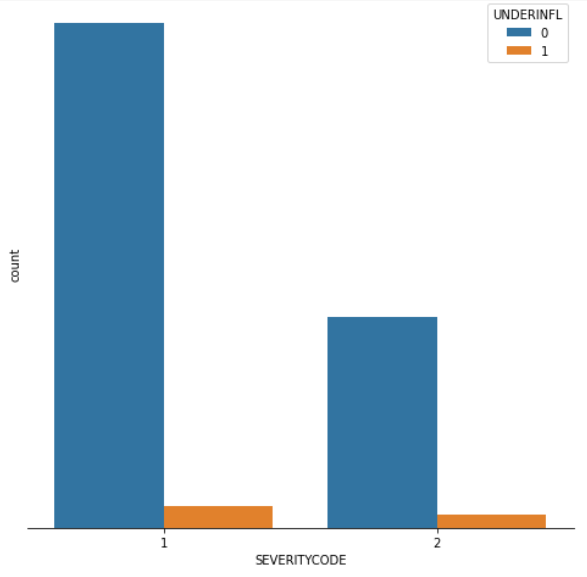
**3.2 Distribution of Inattention Type against Severity Code**

Many people talk on phone or use other devices when they are driving. This causes an distraction from driving which then lead to accidents. In this case, we can observe that most of the incidents took place when the driver was attentive. Very few cases were reported where the driver was inattentive. It is seen that, in the case where driver was inattentive it caused more damage to the property.



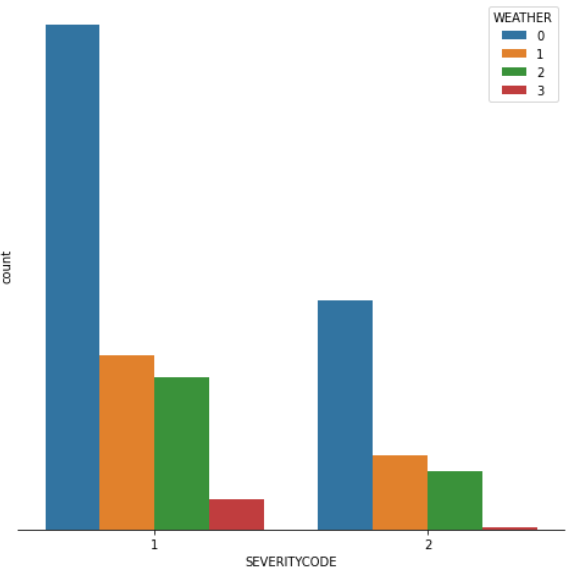
**3.3 Distribution of Under Influence Type against Severity Code**

All over the world drink and drive is the worst case of an road incident. Many people drive under the influence of either drugs or drinks which makes them inattentive on the road causing severe accidents. In this case, since the data being unbalanced it is seen that most of the incidents took place when the driver was not under any kind of influence. Very few cases were reported where the driver was drunk or under any drug influence. Most of the cases reported are for property damage.



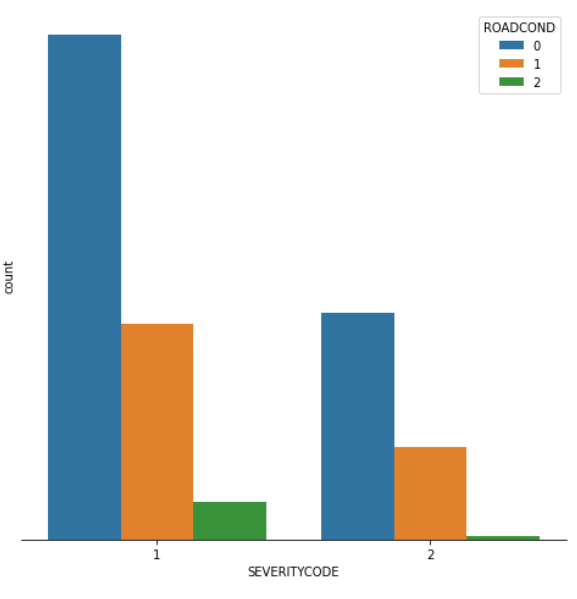
**3.4 Distribution of Weather Type against Severity Code**

Weather has been a major contributor to road accidents. Due to bad weather, sometimes the vision of driver is disturbed which causes road accidents. Looking at the plot, it is seen that type of weather did not affect much as we can see that most of the incident took place under clear weather. Less than 50% cases were reported for rain or snowy conditions. Compared to severity code 2, many cases were reported for code 1 where weather was either rainy or cloudy.



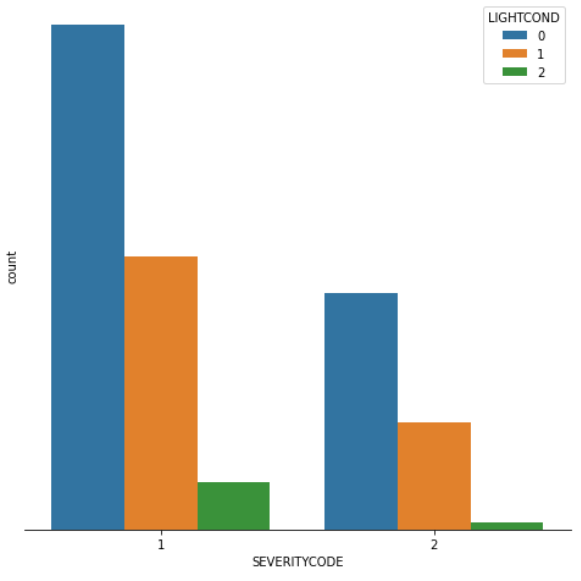
**3.5 Distribution of Road Condition Type against Severity Code**

Many times it is the road condition which causes an accident. Sometimes when it rains heavily or there is a heavy snowfall the road becomes wet and a bit slippery which causes road incidents. In out dataset it is observed that most of the incidents took place under dry road conditions. A fair amount of report were observed when the road conditions were wet or slippery. Road conditions didn’t led to many injuries.



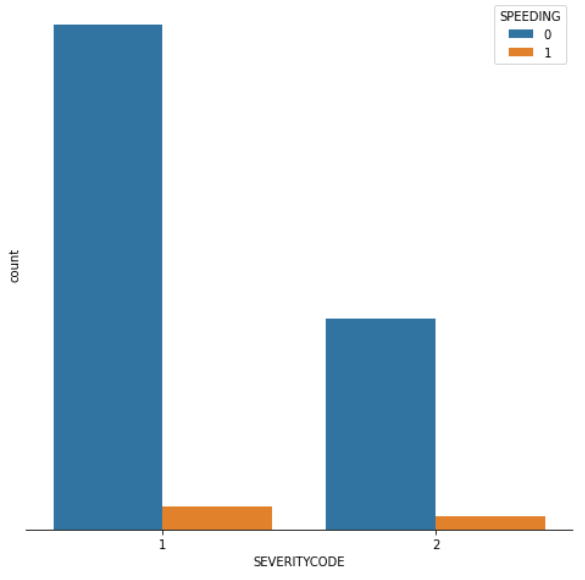
**3.6 Distribution of Light Condition Type against Severity Code**

Lightning conditions hardly matter many times. Usually the roads are well lit up which reduces the number of accidents. In this case, we can observe that very cases were report when the lightning conditions were dark. Most of the incidents took place under bright sunny day or well-lit area.



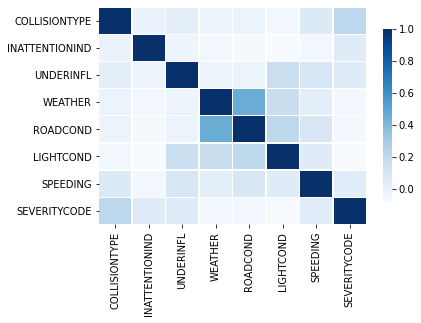
**3.7 Distribution of Speeding Type against Severity Code**

Speeding is one the worst type in reported cases. Many people tend to speed when they drive which lead to major car accidents. Since our dataset is imbalanced it is observed that very few cases have been reported when the driver was speeding.



**3.8 Correlation Analysis**

In the following figure, we can see that the correlation between features is very low. This might affect the predictions. It is seen that the correlation between weather and road condition is pretty strong as it is obvious that if it rains then the road gets wet. So the dependencies is observed in that case. We can see that correlation between severity code and other features is low, the highest amongst the features is the collision type.



**4. Predictive Modeling**

For this project, three machine learning models were used which are Logistic Regression, Decision Tree Analysis and k-Nearest Neighbor. Support Vector Machine (SVM) model was not used because it is inaccurate with large dataset. These models were used from two packages Sklearn and PyCaret. The goal is to make comparison and choose the best perfoming model.

**4.1 Logistic Regression**

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression).

For Logistic regression model from Sklearn package, Hyperparameter tuning was doing using GridSearchCV method. The optimum parameter came out as : Penalty – L1, C – 0.0001 with solver set to ‘liblinear’.

**4.1.1 Classification Report**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| 1 | 0.78 | 0.62 | 0.69 | 40847 |
| 2 | 0.40 | 0.59 | 0.48 | 17555 |
| Accuracy |  |  | 0.62 | 58402 |
| Macro Avg | 0.59 | 0.61 | 0.59 | 58402 |
| Weighted Avg | 0.67 | 0.62 | 0.63 | 58402 |

**4.2 Decision Tree**

Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute. This process is then repeated for the subtree rooted at the new node. By displaying a sequence of nodes, decision trees give an effective and easy way to visualize and understand the potential options of a decision and its range of possible outcomes. The decision tree also helps to identify every potential option and weigh each course of action against the risks and rewards each option can yield.

For Decision Tree model from Sklearn package, Hyperparameter tuning was doing using GridSearchCV method. The optimum parameter came out as : Criterion – gini, max\_depth – 11 with min\_sample\_leaf set to 5.

**4.3 k-Nearest Neighbor**

k-nearest neighbors (kNN) is a type of lazy learning algorithm. It is one of the simplest algorithm. kNN can be used for both regression as well as classification. For our case, kNN is used as a classifier. K-nearest neighbors (KNN) algorithm uses ‘feature similarity’ to predict the values of new datapoints which further means that the new data point will be assigned a value based on how closely it matches the ‘k’ neighbouring points in the training set.

For k-Nearest Neighbor model from Sklearn package, Hyperparameter tuning was doing using GridSearchCV method. The optimum values for n\_neighbor turned out as 17.

**4.4 Performance Evaluation**

Classification report for each model was generated along with confusion matrix. Results for each model are shown as follows.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1-Score |
| Logistic Regression | 0.62 | 0.67 | 0.62 | 0.63 |
|  |  |  |  |  |
| Decision Tree | 0.65 | 0.72 | 0.65 | 0.67 |
| K-Nearest Neighbor | 0.69 | 0.68 | 0.69 | 0.68 |

**References:**

https://en.wikipedia.org/wiki/Seattle

https://seattle.curbed.com/2017/8/10/16127958/seattle-population-growth-cars-transit