# Capstone Project - CAR ACCIDENT SEVERITY

Sasith Rajasooriya, Ph.D.

This Project report is organized as follows.

# 1. INTRODUCTION

## 1.1 Background of the project

While the number of fatal motor vehicle accidents record a 2% decline from 2018 in USA, approximately 38800 people lost their lives in accidents. In addition about 4.4 million people have been injured enough to require medical attention. Therefore, while the decreasing trend is a positive, observing insights into accident data is important so that suitable proposals could be made to implement policy decisions and informing people to reduce possible accidents further.

The main observation affecting a particular driver on the road is "possible delay caused due to an accident". The severity of a particular accident can be assessed by the delay it would cause on the traffic. It is clear that the impact of an accident on traffic is highly proportionate to the severity of the accident. Therefore, if people (drivers) can be informed about the severity of possible accidents based particular factors such as weather condition, Wind Speed, Visibility etc. they can be more careful in driving and the choice of their driving routes.

In this project I try different techniques to develop several machine learning models that can predict the severity of road accidents given a set of variables relevant to accidents recorded. Then we will compare the performances of each of these different models and then chose a best model out of what I developed.

A machine learning model which is successful in predicting the severity with a great accuracy will make sure following important informative measures are taken efficiently.

(a). Successful machine learning model will elaborate a set of conditions (ex: temperature, wind speed, other weather conditions etc.) which are found to be highly related to accidents with higher level of severity (ex: degree 4 accidents. Then, if these conditions are present in a particular time (day, week, month etc.), people in the considered areas can be informed to take extra care, to avoid particular routes, to use optional rules. Weather news alerts and radio news broadcasting stations in the areas can inform/alert drivers about these conditions and to take extra care.

(b). Traffic Police officials and patrolling security officials can be informed about these conditions prior and make sure additional security and road/highway support teams are allocated on these areas.

(c). Signboard alerts, information boards on the roads can present alerts to drives accordingly.

(d). Road workers and related utility construction workers can also be informed prior for caution.

## 1.2 The Problem (Classification of Accidents based on severity)

Severity of road accidents is an important indication on the choice of the driving route and the attention on careful driving in specific environment conditions. Given the historical data on accidents, if there are successful models to predict the "severity" of a possible accident these can be used in informing people as mentioned under section 1.1. However, to do this we need to have effective models to "classify" accidents based on their severity.

Objective of this project is to address this problem using available data and to develop a machine learning model to answer key problems given below.

1. "What is the severity of the probable accident in the USA under a set of certain conditions (features)?
2. How can we classify to train and test effective machine learning models?

## 2. DATA SOURCE AND THE DATASET

### 2.1 Data Source

In this project I use the accident data in the United States in the "US Accidents" data set available in kaggle.com. This data set is openly available for research purposes hence usable in my project. The data set is developed from "Traffic Streaming Report APIs" and collected using streaming reports on important traffic events. The data set is maintained by its owner Sobhan Moosavi. The data set has 3513740 accident data recorded from 2016-02-01 to 2019-12-31, and further updated annually.

From over 3.5 million of accident data all over the United States, for this project I have chosen random samples to comply with the memory capacity of my machine that runs the algorithm. Original data set has 49 attributes including both categorical and quantitative types.

### 2.2 Dataset

Before developing machine learning models, in the section 3 (Preprocessing), proper data cleaning and formatting will be conducted followed by a preliminary and descriptive analysis on the main variables. Then aiming at the perdition model development, a "Feature Set" will be obtained. To ensure the balance of the data set, severity column shall be observed for the number of accidents in each different category. Higher priority will be given to the accidents with the highest severity class. If the high severity accidents are a minority class, needed under sampling or oversampling techniques will be applied.

Among the variables, attributes Weather condition, Visibility, Wind Speed, Temperature, Air Pressure are key factors that would be considered in the model building. Attribute "Weather condition" in the data set has many categories of the inputs such as "clear, partly cloudy, Scattered cloudy, Mostly cloudy, Drizzle, Fair, Windy, Fog, Snow, Heavy Snow, Heavy rain, Heavy rain/windy, Windy". These variables are clear indicators on the risk associated with the driving in a particular area. Therefore my feature set would hopefully would consist of these attributes. However, if there are too many levels of this variable priority will be given to the variables which has the highest frequencies. In addition, check for independence among categorical variables will be conducted in case more than one categorical attribute is included in the model feature set.

After conducting a suitable "exploratory analysis" and before the model selection procedures are applied "dimensionality reduction" approach also will be considered to check if that helps to reduce the dimension and the focus of the results.

Once the feature set is selected, suitable different machine learning approaches will be applied and assessed. Finally, after evaluating the model performances using different indices, the best model or models will be selected and presented with results.

## 3. METHODOLOGY PART I : PREPROCESSING

In this section preprocessing steps taken before applying the classification algorithms will be discussed. This project was conducted using a Jupiter Notebook on the “Anaconda Navigator” on my PC. Python version 3.6 was applied in the notebook to conduct all the analytical and data preprocessing and processing work.

### 3.1 Loading and having an initial look into the Data set

First the data set was read as a csv file in a pandas data frame in to the notebook. Using relevant commands size (shape) of the data set and headings of the data set, were observed. The data set is of the dimension of 3513740 rows and 49 columns. Each row had measured and recorded attributes of an accident occurred. Different data types was observed as mentioned below.

**Table 01: Data Types**

| | | | |
|---|---|---|---|
| ID | object | Humidity(%) | float64 |
| Source | object | Pressure(in) | float64 |
| TMC | float64 | Visibility(mi) | float64 |
| Severity | int64 | Wind_Direction | object |
| Start_Time | object | Wind_Speed(mph) | float64 |
| End_Time | object | Precipitation(in) | float64 |
| Start_Lat | float64 | Weather_Condition | object |
| Start_Lng | float64 | Amenity | bool |
| End_Lat | float64 | Bump | bool |
| End_Lng | float64 | Crossing | bool |
| Distance(mi) | float64 | Give_Way | bool |
| Description | object | Junction | bool |
| Number | float64 | No_Exit | bool |
| Street | object | Railway | bool |
| Side | object | Roundabout | bool |
| City | object | Station | bool |
| County | object | Stop | bool |
| State | object | Traffic_Calming | bool |
| Zipcode | object | Traffic_Signal | bool |
| Country | object | Turning_Loop | bool |
| Timezone | object | Sunrise_Sunset | object |
| Airport_Code | object | Civil_Twilight | object |
| Weather_Timestamp | object | Nautical_Twilight | object |
| Temperature(F) | float64 | Astronomical_Twilight | object |
| Wind_Chill(F) | float64 | | |

### 3.2 Data Cleaning

After the initial look into the data set, data cleaning steps were taken to make the data set ready for the objectives of running and training for classification techniques.

### 3.2.1 Removing irrelevant and unusable variables.

This is a big data set with over 3.5 million data points and 49 attributes. However, my analysis is conducted without cloud or parallel computing techniques. Metadata information about this data set is also available on kaggle.com and the same is uploaded with this project on GitHub. Reading the Metadata we can understand that some variable recorded are relevant to the accident only from the administrative point of view. Such variables can be disregarded. First column of Table 2 below shows the list of variables that was removed because of that.

**Table 2: Removed variables and the reason for removal**

| Irrelevant for the project objectives | | Large portion of values are missing | To avoid Multicolinearity |
|---|---|---|---|
| Source | Airport_Code | Wind_Chill(F) | Sunrise_Sunset |
| TMC | Weather_Timestamp | Precipitation(in) | |
| Start_Time | Amenity | | |
| End_Time | Bump | | |
| Start_Lat | Crossing | | |
| Start_Lng | Give_Way | | |
| End_Lat | Junction | | |
| End_Lng | No_Exit | | |
| Description | Railway | | |
| Number | Roundabout | | |
| Street | Station | | |
| Side | Stop | | |
| City | Traffic_Calming | | |
| County | Traffic_Signal | | |
| State | Turning_Loop | | |
| Zipcode | Civil_Twilight | | |
| Country | Nautical_Twilight | | |
| Timezone | Astronomical_Twilight | | |

Further, there are geo spatial variables recorded in the data set. It should be noted that, some of these might be considered and could be important in a model development with a larger scope of qualitative and geo spatial context. But, my project scope mainly focus on the weather conditions and variables relevant to weather. The variable “Weather_Timestamp” is a nominal variables with the time of the accident recorded hence not useful in my analysis and model development. Other variables in the first column are mainly nominal nature with no direct relevant to our analysis.

### 3.2.2 Dealing with the Missing data

After removing irrelevant variables, a check for missing values were executed. Below is a list of the variables with the number of missing values given in the argument output “TRUE” for missing values.

**Table 03: Counts of missing data values in each column**

| **Missing values present** |
|---|
| Temperature(F)<br>False 3448004- True 65736 |
| Wind_Chill(F)<br>True 1868256- False 1645484 |
| Humidity(%)<br>False 3444049- True 69691 |
| Pressure(in)<br>False 3457856- True 55884 |
| Visibility(mi)<br>False 3437879- True 75861 |
| Wind_Direction<br>False 3454863- True 58877 |
| Wind_Speed(mph)<br>False 3059127- True 454613 |
| Precipitation(in)<br>True 2025881- False 1487859 |
| Weather_Condition<br>False 3437597- True 76143 |
| Sunrise_Sunset<br>False 3513624- True 116 |

**Summary of the information obtained on missing data.**

Out of the 21 columns, 9 columns did not have any missing values, which is very good. Other 12 columns had missing values that we have to deal with. After the removal of irrelevant variables, only 10 left with the missing values to deal with.

1. There are few several important quantitative variables that we have priority consideration in building our model. Some other variables have other importance but not directly involved in our model. Therefore, we will first check the effect of the missing data in each of these attribute according to their nature.
2. Variables “Precipitation” and “Wind_Chill” were missing a large portion of data values. However, there are other attributes which illustrates the similar nature of effects that would be expected from these two attributes (such as “Weather_condtion”, “Wind_speed” and Temperature). Therefore, these two variables were dropped from the data frame.
3. Attribute Wind_speed is also missing over 10% of the values recorded. We will replace these missing values by the average wind speed.
4. Other records with missing attributes values will be removed.

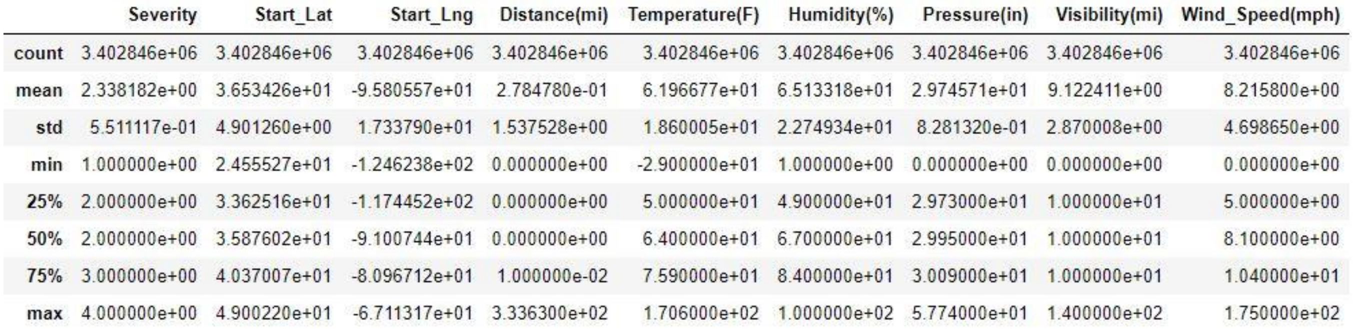
### 3.2.3. Possible data entry errors (Wind speed)

Record of the Wind speed with 984 mph must be a data entry mistake. After looking into the column it was found that there are several data entry errors. As per real records, the highest recorded Wind Speed in the history of US is around 231 mph, which is in 1984. Therefore, keeping that in mind, we removed accident records with “Wind speed” recorded over 200 mph for better accuracy. After that, missing data values in the Wind_Speed column was replaced with the average Windspeed.

### 3.3 Exploratory Analysis on the variables before choosing the feature set

After missing data was managed descriptive statistics on the variables were obtained. Table 4 below present the numerical statistics on each quantitative variable considered.

Table 04: Descriptive Statistics



| | Severity | Start_Lat | Start_Lng | Distance(mi) | Temperature(F) | Humidity(%) | Pressure(in) | Visibility(mi) | Wind_Speed(mph) |
|---|---|---|---|---|---|---|---|---|---|
| count | 3.402846e+06 | 3.402846e+06 | 3.402846e+06 | 3.402846e+06 | 3.402846e+06 | 3.402846e+06 | 3.402846e+06 | 3.402846e+06 | 3.402846e+06 |
| mean | 2.338182e+00 | 3.653426e+01 | -9.580557e+01 | 2.784780e-01 | 6.196677e+01 | 6.513318e+01 | 2.974571e+01 | 9.122411e+00 | 8.215800e+00 |
| std | 5.511117e-01 | 4.901260e+00 | 1.733790e+01 | 1.537528e+00 | 1.860005e+01 | 2.274934e+01 | 8.281320e-01 | 2.870008e+00 | 4.698650e+00 |
| min | 1.000000e+00 | 2.455527e+01 | -1.246238e+02 | 0.000000e+00 | -2.900000e+01 | 1.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 25% | 2.000000e+00 | 3.362516e+01 | -1.174452e+02 | 0.000000e+00 | 5.000000e+01 | 4.900000e+01 | 2.973000e+01 | 1.000000e+01 | 5.000000e+00 |
| 50% | 2.000000e+00 | 3.587602e+01 | -9.100744e+01 | 0.000000e+00 | 6.400000e+01 | 6.700000e+01 | 2.995000e+01 | 1.000000e+01 | 8.100000e+00 |
| 75% | 3.000000e+00 | 4.037007e+01 | -8.096712e+01 | 1.000000e-02 | 7.590000e+01 | 8.400000e+01 | 3.009000e+01 | 1.000000e+01 | 1.040000e+01 |
| max | 4.000000e+00 | 4.900220e+01 | -6.711317e+01 | 3.336300e+02 | 1.706000e+02 | 1.000000e+02 | 5.774000e+01 | 1.400000e+02 | 1.750000e+02 |

Further, to have a look into the distribution of each quantitative variable, a histogram was obtained as given in Figure 01.

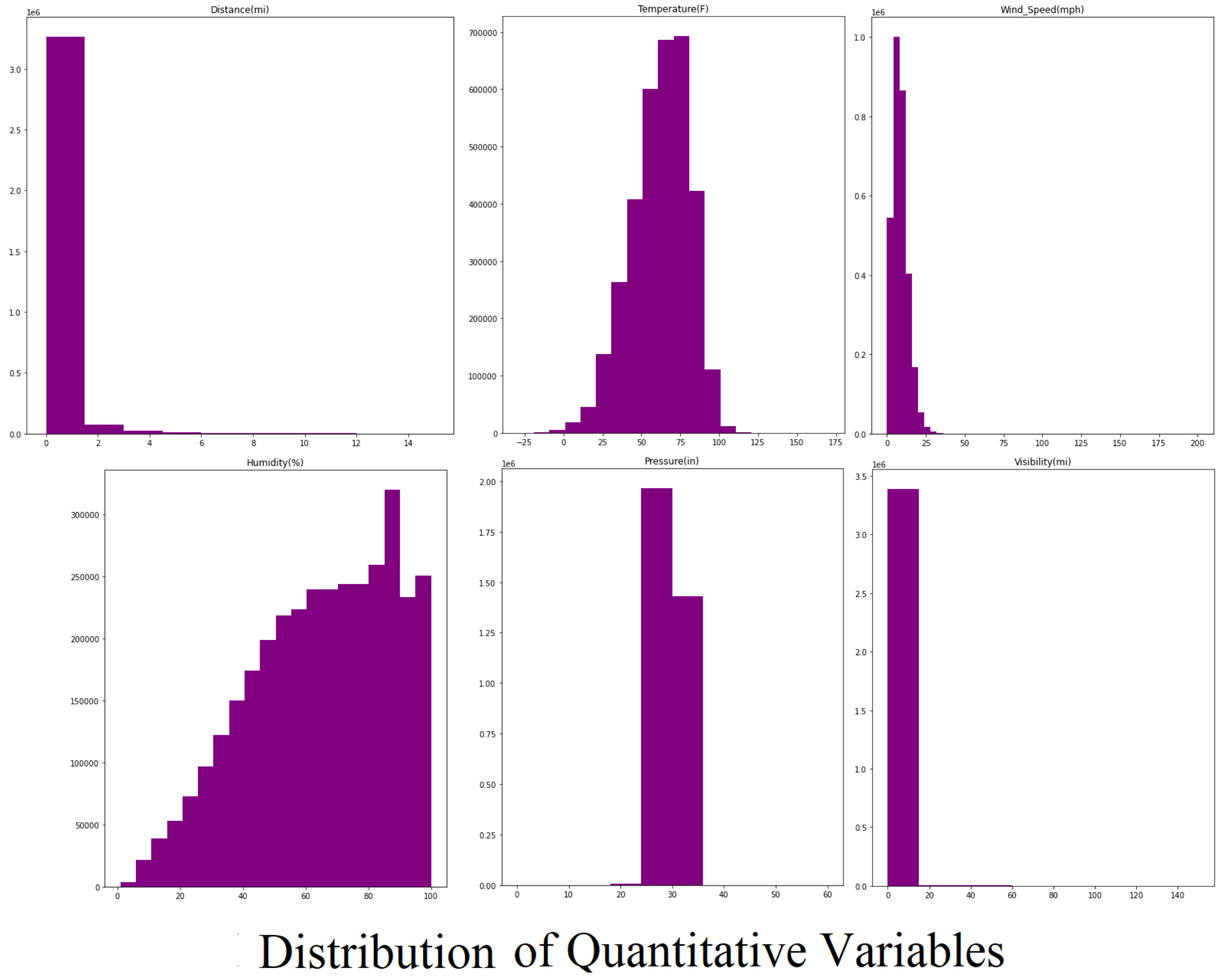
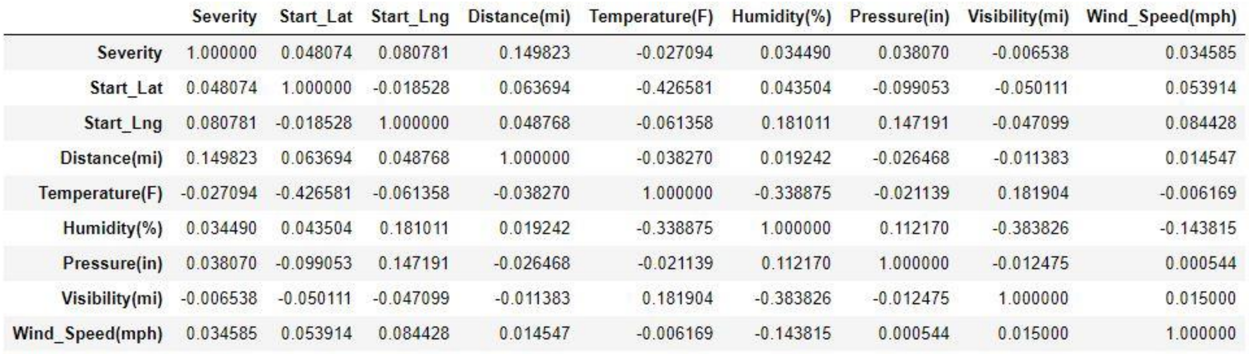


**Figure 01: Distribution of Quantitative variables**

**Table 05: Correlation Matrix**



| | Severity | Start_Lat | Start_Lng | Distance(mi) | Temperature(F) | Humidity(%) | Pressure(in) | Visibility(mi) | Wind_Speed(mph) |
|---|---|---|---|---|---|---|---|---|---|
| **Severity** | 1.000000 | 0.048074 | 0.080781 | 0.149823 | -0.027094 | 0.034490 | 0.038070 | -0.006538 | 0.034585 |
| **Start_Lat** | 0.048074 | 1.000000 | -0.018528 | 0.063694 | -0.426581 | 0.043504 | -0.099053 | -0.050111 | 0.053914 |
| **Start_Lng** | 0.080781 | -0.018528 | 1.000000 | 0.048768 | -0.061358 | 0.181011 | 0.147191 | -0.047099 | 0.084428 |
| **Distance(mi)** | 0.149823 | 0.063694 | 0.048768 | 1.000000 | -0.038270 | 0.019242 | -0.026468 | -0.011383 | 0.014547 |
| **Temperature(F)** | -0.027094 | -0.426581 | -0.061358 | -0.038270 | 1.000000 | -0.338875 | -0.021139 | 0.181904 | -0.006169 |
| **Humidity(%)** | 0.034490 | 0.043504 | 0.181011 | 0.019242 | -0.338875 | 1.000000 | 0.112170 | -0.383826 | -0.143815 |
| **Pressure(in)** | 0.038070 | -0.099053 | 0.147191 | -0.026468 | -0.021139 | 0.112170 | 1.000000 | -0.012475 | 0.000544 |
| **Visibility(mi)** | -0.006538 | -0.050111 | -0.047099 | -0.011383 | 0.181904 | -0.383826 | -0.012475 | 1.000000 | 0.015000 |
| **Wind_Speed(mph)** | 0.034585 | 0.053914 | 0.084428 | 0.014547 | -0.006169 | -0.143815 | 0.000544 | 0.015000 | 1.000000 |

### 3.3.1 Check for correlation and possible Multicollinearity among quantitative attributes

To check for possible high correlation and the risk of having multicolinearity between pairs of variables matrix of correlation coefficients was obtained. As shown in Table 05, there were no strong correlation between any pairs of quantitative variables, which made our analysis easier.

### 3.3.2 Check for possible Multicolinearity among categorical attributes

In addition to these numerical variables, we had two possible categorical variables those were possible candidates to be in the feature set. They are “Weather condition” and the “Sunrise_Sunset”. A Chi-Square procedure was applied to check for the independence of these two variables.

***Chi-Square Test for independence:***

There are two categorical attributes left in our updated data set. They are 'Weather_Condition' and ‘Sunrise_Sunset'. Before I proceed we need to check if they are independent from each other to keep in the model. If not, one of them should be dropped. Let's conduct a Chi-Square test for the independence.

Chi-Square test results were statistically significant.

p-value: [0. 0.]
Significance level: 0.05
Degree of freedom: 125

As the p-value of the Chi-Square test is close to 0, we reject the null hypothesis and concluded that, “weather condition” and “Sunrise_Sunset” (day or night) variables are not independent. Therefore, we dropped "Sunrise_Sunset" from and proceeded with weather condition only.

### 3.3.3 Exploring the "target" (Label) attribute - "Severity"

Our objective is to predict the "Severity". Let’s first explore this attribute column and its distribution.

Table 06: Severity counts and relative counts

| Severity | Count | % |
|---|---|---|
| 1 | 28540 | 0.008387 |
| 2 | 2302403 | 0.676611 |
| 3 | 964484 | 0.283435 |
| 4 | 107419 | 0.031567 |

We see that severity level 2 and 3 cover over 95% of the data set. Severity level 1 is below 0.8% of all recorded accidents. **However, 3.1% of all the accidents are categorized into Severity 4.** Indeed, our focus must be higher on Severity level 4 than any other category since our models ability to predict severity 4 is very important. Since, Severity 1 accidents are of a negligible amount and prediction power of other categories by the model is of higher importance than severity level 1, for simplicity we merged severity 1 and 2 together.

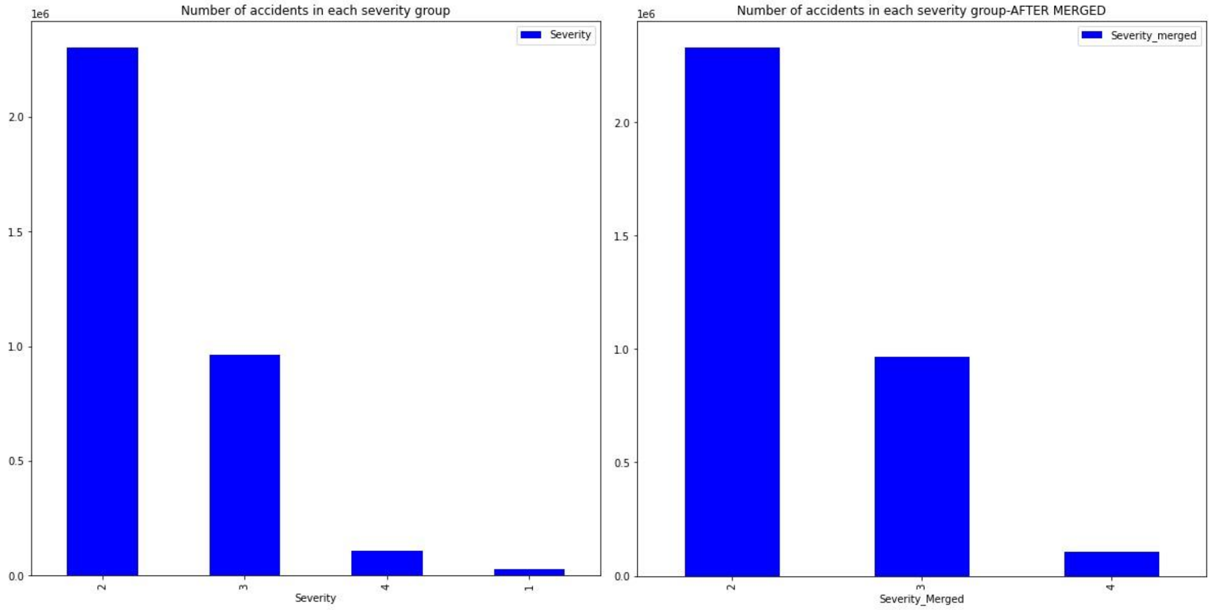


**Figure 02: Severity and Merged Severity**

### 3.3.4 Preprocessing categorical feature "Weather_Condition" for model building

As evident in the Python output given below, only categorical variable, “Weather condition” we have in our data frame has 126 different categories (levels). Handling such a big number of levels is difficult and mostly unnecessary. When looked into the value counts, it was found that out of 126 categories only 8 categories have occurred at frequencies higher than 100,000. There were many levels which occurred only very few times including only once in some cases. Therefore, this column was re-organized by bringing down from 126 to 20 levels with the highest frequencies and adding all other conditions in to one category called "Other Weather Conditions".

```
Clear                800907
Fair                 538522
Mostly Cloudy         484855
Overcast              378965
Partly Cloudy         342143
Cloudy                210335
```

```
Scattered Clouds        203398
Light Rain                 175377
Light Snow                   50081
Other_Weather_Condition         49089
Rain                          41406
Haze                           38087
Fog                          30625
Heavy Rain                    15154
Light Drizzle                 12325
Fair / Windy                   7879
Snow                         5707
Light Thunderstorms and Rain      4908
Mostly Cloudy / Windy           4425
Thunderstorm                  4358
Cloudy / Windy                 4300
```

### 3.4 Making the Feature set X and Target set

Feature set has both the numerical part and the categorical part of the attribute “weather condition”. Before proceeding, these two parts were combined into one data frame including the target set of the "Severity_merged" using concatenating commands in pandas library.

### 3.5 Dimensionality Reduction effort (PCA)

After obtaining the complete and cleaned data set before model building, a PCA (Principle Component Analysis) were considered with the objective of reducing the dimension of the data set.

**PCA unsuccessful**

As the plot and the explained variances of the chosen components suggest PCA did not help us well in REDUCING THE DIMENTIONALITY of the accidents severity feature set. Explained variances did not add up to at least 95% of the variance (first component about 81.6% and the second component 5.6% didn’t add up to at least 95% of the variation). Also the scatter plot given in the Figure 03 illustrates that no clear classification or separation. This is probably because of the principle components chosen does not do a good job in explaining the severity of an accident, or it may be attributes in the data set does not account well for the severity of an accident that we would find out in later sections.

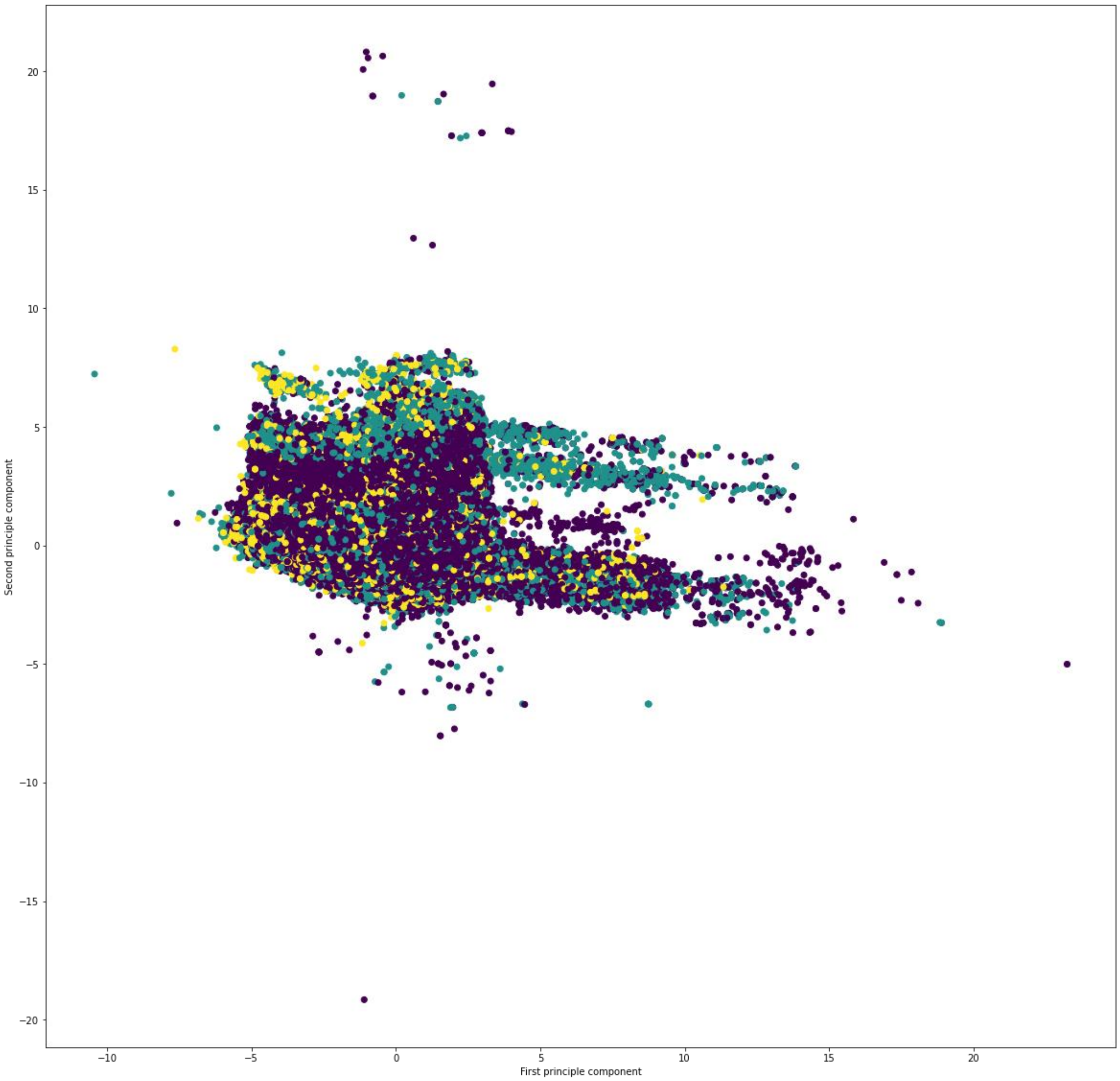


Figure 03: PCA Outcome and Classification :( First and Second Components)

## 4. METHODOLOGY PART II : SAMPLING AND BALANCING THE SAMPLES

At this point, the preprocessing of the dataset is complete and we are ready for applying and training classification algorithms. However, before that there is one more step to take. That is to execute a proper sampling procedure. The data set is too large for running classification algorithms, in a PC platform without higher memory capacity. Therefore suitable random samples were obtained.

### 4.1 Balancing the Data

Let’s remind that our data set is not a balanced one. It is mostly observed that accident severity data is highly imbalanced. Therefore, it was needed to take steps to make the data set balanced using steps such as Random “Under-sampling” from majority subsets.

Let’s remind the distribution plot of the accident severity value counts where we merged severity levels one and 2 together (Figure 02). We know that the high severe accident category (severity 4) is now the minority in our data.

### 4.2 Making a Main Random Sample

For processing convenience, first a random sample of the size 15000 was obtained from our cleaned data frame. Taking this step was compelled since when the entire data set or a larger sample was run without high power cloud computing or similar techniques, the Kernel did not run due to memory limitations. However, sample of size 15000 including all categories is a reasonable and is a representative random sample. Therefore, proceeded with our model development with that.

**Severity value count of the Main Random Sample**

```
2    10121
3     4374
4      505
```

### 4.3 Generating the undersample based on the Minority class

In order to treat the imbalance nature of classes in the target variable Severity, every classification model was trained using two different variations of the dataset.

The first variation is the main sample. In this case, what we have is an imbalanced dataset. The second variation is an under-sampled version of the dataset as obtained in this section below. In this variation, only the class with the least amount of samples will remain as it. In this case, that is the class of severity 4.

**Severity value count of Balanced Sample**

```
4    107419
3    107419
2    107419
Name: Severity, dtype: int64
```

### Train Test Split

For both the samples, Train_Test_Split were conducted with 70% for training and 30% for testing.

## 5. RESULTS: CLASSIFICATION

In this section execution of the algorithms of classification machine learning and performance of each will be discussed and compared. As mentioned above for each classifiers we trained and evaluate two models (one with the imbalanced data and the other is with the balanced data). Then comparison of the performances of these classifiers were done.

Following classifiers were applied.

1. KNN
2. SVM
3. Random Forrest

### 5.1 KNN (K- nearest neighbors)

K-Nearest Neighbors is an algorithm for supervised learning where the data is ‘trained’ with data points corresponding to their classification. Our objective is also to predict the class of "severity" in accidents. Therefore we can use KNN in our classification model building. Once a point is to be predicted, KNN takes into account the ‘K’ nearest points to it to determine its classification. Therefore, it is important to consider the value of K and obtained the best K before training the algorithm. We calculate the best "K" which has the highest accuracy and then run KNN with that best K value.

**KNN Performances**

**KNN_Mainsample Model**

```
            Precision    recall  f1-score   support

           2       0.68      0.96      0.80      3022
           3       0.46      0.08      0.13      1347
           4       0.00      0.00      0.00       131
    accuracy                           0.67      4500
   macro avg       0.38      0.35      0.31      4500
weighted avg       0.60      0.67      0.58      4500
```

**KNN_undersample Model**

```
            Precision    recall  f1-score   support

           2       0.41      0.52      0.46      1495
           3       0.42      0.42      0.42      1524
           4       0.64      0.46      0.54      1481
    accuracy                           0.47      4500
   macro avg       0.49      0.47      0.47      4500
weighted avg       0.49      0.47      0.47      4500
```

## 5.2 SVM (Support Vector Machines)

SVM (Support Vector Machines) works by mapping data to a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable. A separator between the categories is found, then the data is transformed in such a way that the separator could be drawn as a hyperplane. Following this, characteristics of new data can be used to predict the group to which a new record should belong.

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points. In this section we will train both the "Main sample" and "under sample" SVM.

### SVM Performances

### SVM_Mainsample Model

```
             precision    recall  f1-score   support

           2       0.68      1.00      0.81      3022
           3       0.55      0.02      0.03      1347
           4       0.00      0.00      0.00       131
    accuracy                           0.67      4500
   macro avg       0.41      0.34      0.28      4500
weighted avg       0.62      0.67      0.55      4500
```

### SVM_undersample Model

```
             precision    recall  f1-score   support

           2       0.69      0.91      0.78      3022
           3       0.40      0.15      0.22      1347
           4       0.26      0.04      0.07       131
    accuracy                           0.65      4500
   macro avg       0.45      0.36      0.35      4500
weighted avg       0.59      0.65      0.59      4500
```

## 5.3 Random Forest

Random forest classifier (RF) is an ensemble tree-based learning algorithm. It is a set of decision trees from randomly selected subset of training set. Random forest algorithm aggregates the votes from different decision trees to decide the final class of the test object. Compared to KNN and SVM, Random forest is a strong classifier with higher level of precision and accuracy in many aspects. In this section will train and test RF algorithm in both our main sample and balanced sample sets and evaluate.

**RF Performances**

**RF_Mainsample Model**

```
              precision    recall  f1-score   support

           2       0.69      0.91      0.78      3022
           3       0.40      0.15      0.22      1347
           4       0.26      0.04      0.07       131

    accuracy                           0.65      4500
   macro avg       0.45      0.36      0.35      4500
weighted avg       0.59      0.65      0.59      4500
```

**RF_undersample Model**

```
               precision    recall  f1-score   support

           2       0.50      0.47      0.49      1495
           3       0.51      0.40      0.45      1524
           4       0.68      0.89      0.77      1481

    accuracy                           0.58      4500
   macro avg       0.57      0.58      0.57      4500
weighted avg       0.57      0.58      0.57      4500
```

## 5.4 Grid Search for the BEST MODEL (undersample Random Forest)

It is clear that the undersampled Random Forest model was our best model with an overall accuracy of 0.58 and specially the especially higher precession, recall and f1-scores in for the severity 4 class of the accidents, which is our key objective in this project.

As per the confusion matrix also present very clearly, out of the total of true 1481 accidents of the category 4, 1313 have been correctly predicted by our machine learning algorithm. That is a very good performances compared to the five models we have.

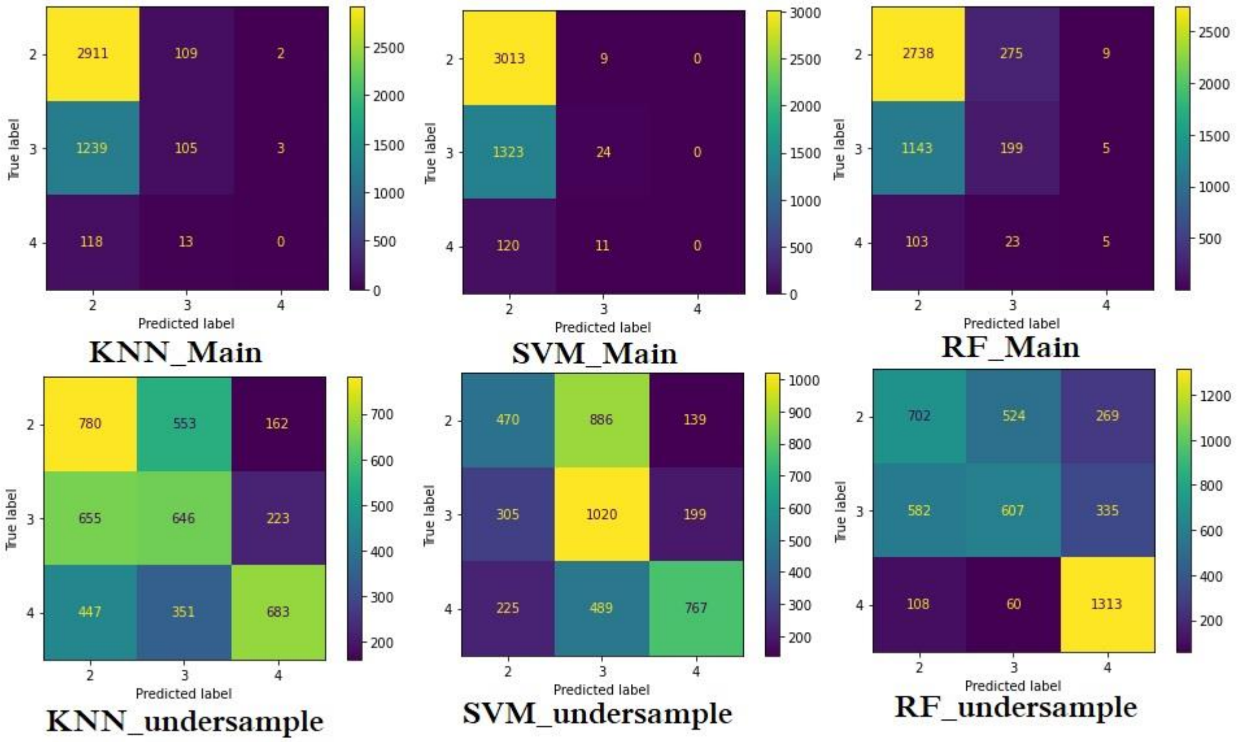
**RF_Grid_undersample Model Performances**

```
           precision    recall  f1-score   support

           2       0.53      0.47      0.50      1495
           3       0.55      0.43      0.48      1524
           4       0.68      0.91      0.78      1481

    accuracy                           0.60      4500
   macro avg       0.59      0.60      0.59      4500
weighted avg       0.59      0.60      0.59      4500
```

Confusion Matrices for first 6 models are is given in Figure 4. Looking at the diagonal values and how dark the cells are we get to see how models are performed. Values in the diagonal elements represents the correct predictions for each class of the accident severity. It is clear that under-sample balanced models outperforms all imbalanced methods in predicting severity class 4 accidents.



KNN_Main

| True label \ Predicted label | 2 | 3 | 4 |
|---|---|---|---|
| 2 | 2911 | 109 | 2 |
| 3 | 1239 | 105 | 3 |
| 4 | 118 | 13 | 0 |

SVM_Main

| True label \ Predicted label | 2 | 3 | 4 |
|---|---|---|---|
| 2 | 3013 | 9 | 0 |
| 3 | 1323 | 24 | 0 |
| 4 | 120 | 11 | 0 |

RF_Main

| True label \ Predicted label | 2 | 3 | 4 |
|---|---|---|---|
| 2 | 2738 | 275 | 9 |
| 3 | 1143 | 199 | 5 |
| 4 | 103 | 23 | 5 |

KNN_undersample

| True label \ Predicted label | 2 | 3 | 4 |
|---|---|---|---|
| 2 | 780 | 553 | 162 |
| 3 | 655 | 646 | 223 |
| 4 | 447 | 351 | 683 |

SVM_undersample

| True label \ Predicted label | 2 | 3 | 4 |
|---|---|---|---|
| 2 | 470 | 886 | 139 |
| 3 | 305 | 1020 | 199 |
| 4 | 225 | 489 | 767 |

RF_undersample

| True label \ Predicted label | 2 | 3 | 4 |
|---|---|---|---|
| 2 | 702 | 524 | 269 |
| 3 | 582 | 607 | 335 |
| 4 | 108 | 60 | 1313 |

Confusion Matrix: Model comparision

Figure 04: Confusion Matrix: Model comparison

However, in developing the RF model, we did not specify the model parameters hence we can check if we have the opportunity to further improve our model to obtain a best model. For this we can call for a Grid Search.

In the Grid search we will give a set of value ranges for important parameters, specially “n_estimators”. Once the Grid is run, we can call for the parameter combination with the optimized accuracy and performances. In this section this procedure is conducted and given below step by step.

**Undersampled Random Forest model ran with the optimized parameters obtained from the Grid search clearly improved our model. It is clear that, Grid search and application of optimum parameter values is a tuning procedure to improve the model performances of machine learning models. All the models including the Grid FR model that we developed are discussed in section section 6.**

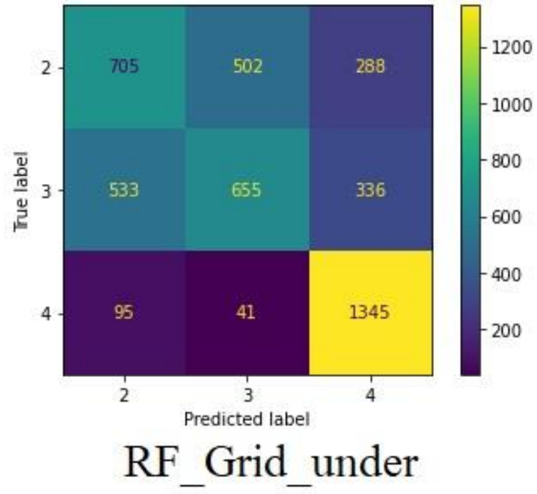
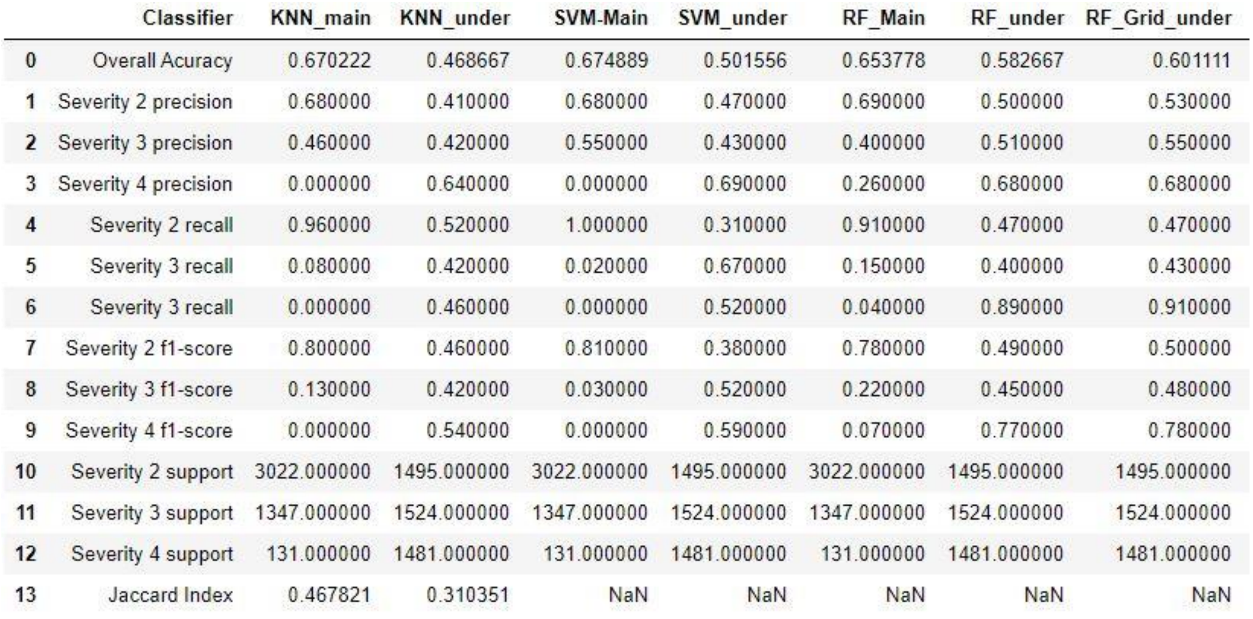


Figure 04: Confusion Matrix: RF-Grid_undersample model

According to Figure 04, RF_Grid model out performed all other models in correctly predicting the accidents of severity class 4. Out of true 1481 accidents of severity class 4, 1345 were correctly predicted by this model, which amount to over 90%. The model succeeded to improve the overall accuracy also to 0.60 while maintaining higher f1-score, recall and precession in all other dimensions.

Table 07: Comparison of model performances



| | Classifier | KNN_main | KNN_under | SVM-Main | SVM_under | RF_Main | RF_under | RF_Grid_under |
|---|---|---|---|---|---|---|---|---|
| 0 | Overall Acuracy | 0.670222 | 0.468667 | 0.674889 | 0.501556 | 0.653778 | 0.582667 | 0.601111 |
| 1 | Severity 2 precision | 0.680000 | 0.410000 | 0.680000 | 0.470000 | 0.690000 | 0.500000 | 0.530000 |
| 2 | Severity 3 precision | 0.460000 | 0.420000 | 0.550000 | 0.430000 | 0.400000 | 0.510000 | 0.550000 |
| 3 | Severity 4 precision | 0.000000 | 0.640000 | 0.000000 | 0.690000 | 0.260000 | 0.680000 | 0.680000 |
| 4 | Severity 2 recall | 0.960000 | 0.520000 | 1.000000 | 0.310000 | 0.910000 | 0.470000 | 0.470000 |
| 5 | Severity 3 recall | 0.080000 | 0.420000 | 0.020000 | 0.670000 | 0.150000 | 0.400000 | 0.430000 |
| 6 | Severity 3 recall | 0.000000 | 0.460000 | 0.000000 | 0.520000 | 0.040000 | 0.890000 | 0.910000 |
| 7 | Severity 2 f1-score | 0.800000 | 0.460000 | 0.810000 | 0.380000 | 0.780000 | 0.490000 | 0.500000 |
| 8 | Severity 3 f1-score | 0.130000 | 0.420000 | 0.030000 | 0.520000 | 0.220000 | 0.450000 | 0.480000 |
| 9 | Severity 4 f1-score | 0.000000 | 0.540000 | 0.000000 | 0.590000 | 0.070000 | 0.770000 | 0.780000 |
| 10 | Severity 2 support | 3022.000000 | 1495.000000 | 3022.000000 | 1495.000000 | 3022.000000 | 1495.000000 | 1495.000000 |
| 11 | Severity 3 support | 1347.000000 | 1524.000000 | 1347.000000 | 1524.000000 | 1347.000000 | 1524.000000 | 1524.000000 |
| 12 | Severity 4 support | 131.000000 | 1481.000000 | 131.000000 | 1481.000000 | 131.000000 | 1481.000000 | 1481.000000 |
| 13 | Jaccard Index | 0.467821 | 0.310351 | NaN | NaN | NaN | NaN | NaN |

## 6. DISCUSSION

### Model performances:

Table above illustrates performances of all 7 models we developed. KNN_Main model records highest the overall accuracy. But, we must not forget that all the main sample models used an imbalanced data set. Overall accuracy score is the sum of true positives and true negatives divided by the total number of samples. This is only accurate if the model is balanced. It will give inaccurate results if there is a class imbalance. Therefore, overall accuracy of models that used the main sample (imbalanced sample) should not be considered in evaluating the model performances. However, for models with the balanced samples, overall accuracy is a reasonable indicator of the performances. Therefore, both models from undersampled Random Forest algorithms give better results while RF_Grid_under model being the best.

Impact of having an imbalanced sample is very clear according to the results. KNN_Main, SVM_Main and even RF_Main performed very poor in almost all indicators in correctly predicting accidents of the class of Severity 4. In practical our key objective is to predict the class of a possible or probable accident under a given weather conditions. However, we must not forget that, the highest priority and the most important objective of our machine learning models must be the ability to predict probable accidents of the Severity category 4. Almost all imbalanced samples fail in performing for this objective. Both KNN and SVM models are highly affected by the imbalance of the samples as all the performance indicators with respect to the minority class is nothing but zero for both these models. Compared to them RF models have a higher robustness to imbalance samples also as the precision being 0.26. Still, the same model performs very much poorly in other indicators too.

However, RF with a balanced sample clearly performed well and predict much better. The best model therefore is the RF_Grid_under model. The same is well evident from comparing the confusion matrices for each model.

For all the imbalanced sample models lover diagonal value which indicates the correct predictions of accidents of the severity level 4 is 0 or very low. But, undersamples models performed well with higher values in the diagonal values. Specially the RF_under and RF_Grid_under.

## SUMMARY AND CONCLUSIONS

### Future Works

Considering these overall results it is clear the Best model out of these 7 is RF_Grid_Under. It must be noted that we only took the approach of undersampling but not the oversampling method with SMOTE in this project. Since our data set was very big and also our random samples consisted with 15000 data points statistically we can be satisfied with the sample size as long as we preserve the randomness. Therefore additional efforts were not taken to conduct oversampling method. But those also can be taken into consideration for any future work of this project or a similar project. There are further steps we can take to improve the performances of the model and fine tune. There are also new approaches such as XG Boost that might outperform RF models. Such approaches are also may be considered in future improvements of this project or any similar projects.

**Thank you!**