Car Accident Severity

Coursera Capstone

Ameya Sandeep Pange

Introduction

- According to the WHO, approximately 1.35 million people die each year as a result of road traffic crashes. Between 20 and 50 million more people suffer non-fatal injuries, with many incurring a disability as a result of their injury.
- Road traffic injuries cause considerable economic losses to individuals, their families, and to nations as a whole. These losses arise from the cost of treatment as well as lost productivity for those killed or disabled by their injuries, and for family members who need to take time off work or school to care for the injured. Road traffic crashes cost most countries 3% of their gross domestic product.

Business Problem

- The 2030 Agenda for Sustainable Development has set an ambitious target of halving the global number of deaths and injuries from road traffic crashes by 2020.
- This would be made easier if there was a way to analyze the main causes and areas that the accidents took place in.
- This would make it easier to take precautionary measures like placing traffic signs to warn people about the high accident risk in a particular area as well as allocate resources like medical and police assistance, etc.
- This project aims at using techniques like Data Science and Machine Learning to build a model which can predict the severity of accidents based on historical data. This would make people drive more carefully in accident-prone areas and would also help the government bodies manage and reduce the number of accidents and the deaths related to them more effectively.

Data

- The data used in the project is historical accident data for the city of Seattle.
- The raw dataset consists of 190000+ unique records and has 37 attributes, numerical (15) as well as categorical(22).
- The dataset includes date and time entries in 2 of the coloumns.
- The labelled data is the ‘severity’ of the accident which is the target variable.
- For feeding the categorical data into the Machine Learning models, it first needs to be cleaned and formatted which will be dealt with in the data preparation stage.
- Many columns can be seen to have missing data or ‘unknown’ data. These values too will be addressed in the data preparation stage.

Data Preparation

- This stage involves the cleaning of the dataset and the significant feature selection.
- The raw dataset has many impurities such as null and unknown values, duplicate coloumns and some unnecessary attributes. There are no duplicate records but the duplicate coloumn 'SEVERITYCODE.1' is dropped from the dataset.

Data Cleaning or Pre-processing

In [8]:

```
#dropping a duplicated coloumn 'SEVERITYCODE.1'
main_df.drop('SEVERITYCODE.1',axis=1,inplace= True)
main_df.describe()
```

Out[8]:

	SEVERITYCODE	X	Y	OBJECTID	INCKEY	COLDETKEY	INTKEY	PERSONCOUNT	PEDCOUNT	PEDCYLCOUNT	VEH
count	194673.000000	189339.000000	189339.000000	194673.000000	194673.000000	194673.000000	65070.000000	194673.000000	194673.000000	194673.000000	194673
mean	1.298901	-122.330518	47.619543	108479.364930	141091.456350	141298.811381	37558.450576	2.444427	0.037139	0.028391	1
std	0.457778	0.029976	0.056157	62649.722558	86634.402737	86986.542110	51745.990273	1.345929	0.198150	0.167413	0
min	1.000000	-122.419091	47.495573	1.000000	1001.000000	1001.000000	23807.000000	0.000000	0.000000	0.000000	0
25%	1.000000	-122.348673	47.575956	54267.000000	70383.000000	70383.000000	28667.000000	2.000000	0.000000	0.000000	2
50%	1.000000	-122.330224	47.615369	106912.000000	123363.000000	123363.000000	29973.000000	2.000000	0.000000	0.000000	2
75%	2.000000	-122.311937	47.663664	162272.000000	203319.000000	203459.000000	33973.000000	3.000000	0.000000	0.000000	2
max	2.000000	-122.238949	47.734142	219547.000000	331454.000000	332954.000000	757580.000000	81.000000	6.000000	2.000000	12

- Then the attributes which will be used for EDA and the model are selected, namely- ‘SEVERITYCODE’, ‘X’ , ‘Y’, ‘PERSONCOUNT’, ‘VEHCOUNT’, ‘INATTENTIONIND’, ‘UNDERINFL’, ‘ROADCOND’, ‘LIGHTCOND’, ‘WEATHER’, ‘ADDRTYPE’.

In [9]:
```
#extracting useful variables
df_use = main_df[['SEVERITYCODE','X','Y', 'PERSONCOUNT','VEHCOUNT','INATTENTIONIND','UNDERINFL', 'ROADCOND',
                  'LIGHTCOND', 'WEATHER','ADDRTYPE']]
df_use.head()
```

Out[9]:

	SEVERITYCODE	X	Y	PERSONCOUNT	VEHCOUNT	INATTENTIONIND	UNDERINFL	ROADCOND	LIGHTCOND	WEATHER	ADDRTYPE
0	2	-122.323148	47.703140	2	2	NaN	N	Wet	Daylight	Overcast	Intersection
1	1	-122.347294	47.647172	2	2	NaN	0	Wet	Dark - Street Lights On	Raining	Block
2	1	-122.334540	47.607871	4	3	NaN	0	Dry	Daylight	Overcast	Block
3	1	-122.334803	47.604803	3	3	NaN	N	Dry	Daylight	Clear	Block
4	2	-122.306426	47.545739	2	2	NaN	0	Wet	Daylight	Raining	Intersection

- In the ‘INATTENTIONIND’ coloumn, the null values i.e. which are not ‘Y’ are replaced with ‘N’. Then the null and ‘unknown’ values are dropped from the selected coloumns. The cleaned dataset now has 166705 records and 11 coloumns.
- Since most of the selected variables are categorical, they are first encoded to numerical variables so that they can be processed. The remining numerical variables are also encoded so that they have a similar impact.

In [12]:

```
#encoding the different categorical variables in the dataframe

e= LabelEncoder()

df_use['underinfl'] = e.fit_transform(df_use['UNDERINFL'])
df_use['inattention'] = e.fit_transform(df_use['INATTENTIONIND'])
df_use['roadcond'] = e.fit_transform(df_use['ROADCOND'])
df_use['lightcond'] = e.fit_transform(df_use['LIGHTCOND'])
df_use['weather'] = e.fit_transform(df_use['WEATHER'])
df_use['personcount'] = e.fit_transform(df_use['PERSONCOUNT'])
df_use['vehcount'] = e.transform(df_use['VEHCOUNT'])

df_use.head()
```

Out[12]:

IT	VEHCOUNT	INATTENTIONIND	UNDERINFL	ROADCOND	LIGHTCOND	WEATHER	ADDRTYPE	underinfl	inattention	roadcond	lightcond	weather	personcount	vehcount
2	2	N	N	Wet	Daylight	Overcast	Intersection	2	0	7	5	4	2	2
2	2	N	0	Wet	Dark - Street Lights On	Raining	Block	0	0	7	2	6	2	2
4	3	N	0	Dry	Daylight	Overcast	Block	0	0	0	5	4	4	3
3	3	N	N	Dry	Daylight	Clear	Block	2	0	0	5	1	3	3
2	2	N	0	Wet	Daylight	Raining	Intersection	0	0	7	5	6	2	[illegible]

The encoded categorical variables are stored in the same dataframe and will later be extracted as needed.

Exploratory Data Analysis (EDA)

- This stage involves analysis of the dataset, with visual methods like graphs and plots, to summarize the characteristics present in the data.
- First the different or unique values of the categorical variables ‘ROADCOND’, ‘LIGHTCOND’, ‘WEATHER’ and ‘ADDRTYPE’ are observed.
- Next a Subplot containing 3 histograms and 1 pie chart is created. The histograms depict the frequency of accidents for different weather, light and road conditions, grouped by the attribute and then the severity of the accident.

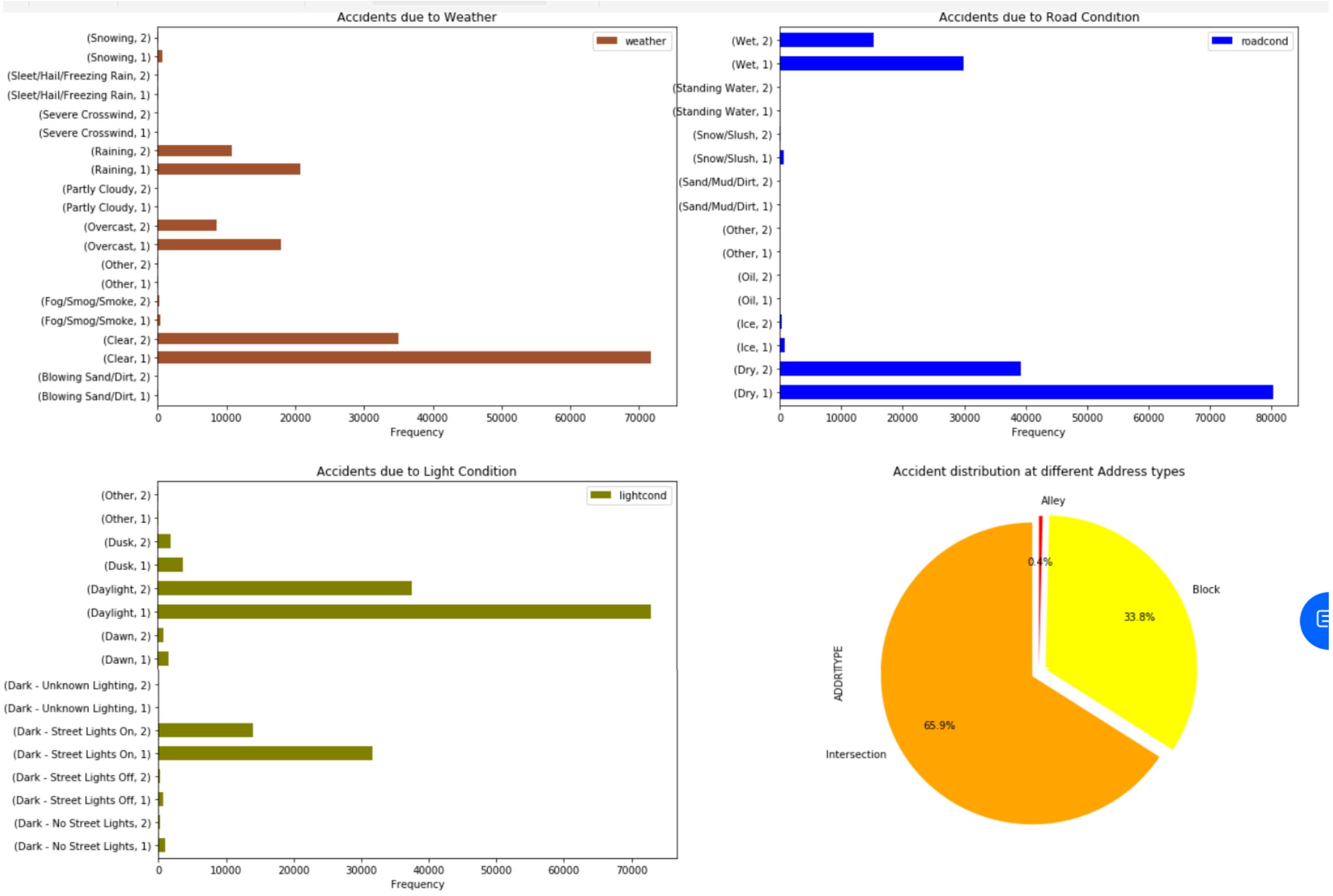
The different road conditions are: ‘Wet’, ‘Dry’, ‘Snow/Slush’, ‘Ice’, ‘Other’, ‘Sand/Mud/Dirt’, ‘Standing Water’ and ‘Oil’.

The different light conditions are: ‘Daylight’, ‘Dark - Street Lights On’, ‘Dark - No Street Lights’, ‘Dusk’, ‘Dawn’, ‘Dark - Street Lights Off’, ‘Other’ and ‘Dark - Unknown Lighting’.

The different weather conditions are: ‘Overcast’, ‘Raining’, ‘Clear’, ‘Snowing’, ‘Other’, ‘Fog/Smog/Smoke’, ‘Sleet/Hail/Freezing Rain’, ‘Blowing Sand/Dirt’, ‘Severe Crosswind’ and ‘Partly Cloudy’.

The different addresses are: ‘Intersection’, ‘Block’ and ‘Alley’.

- The pie chart depicts the proportions of accidents that take place at different types of addresses.

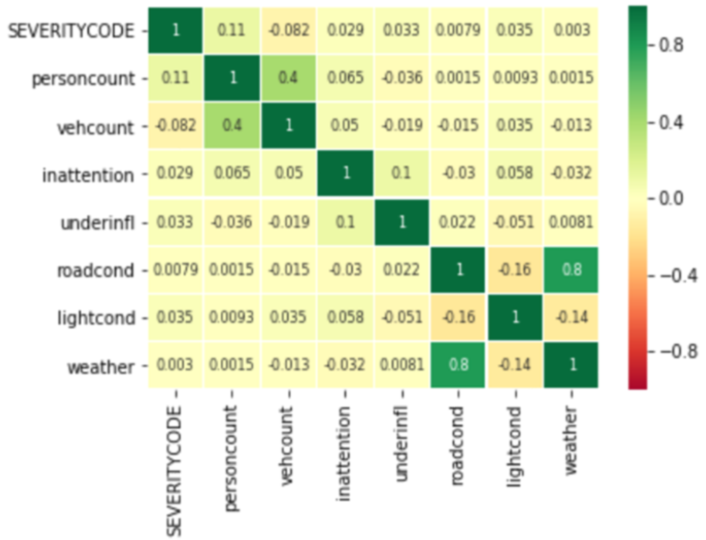


- A correlation matrix between the encoded categorical and numerical variables is constructed. It is then visualised using a Heatmap of colour scheme ‘cmap = ‘RdYlGn’’. This correlation from -1 to +1 is represented by varying intensity of the colours from Red to Yellow and then Green.

Heatmaps

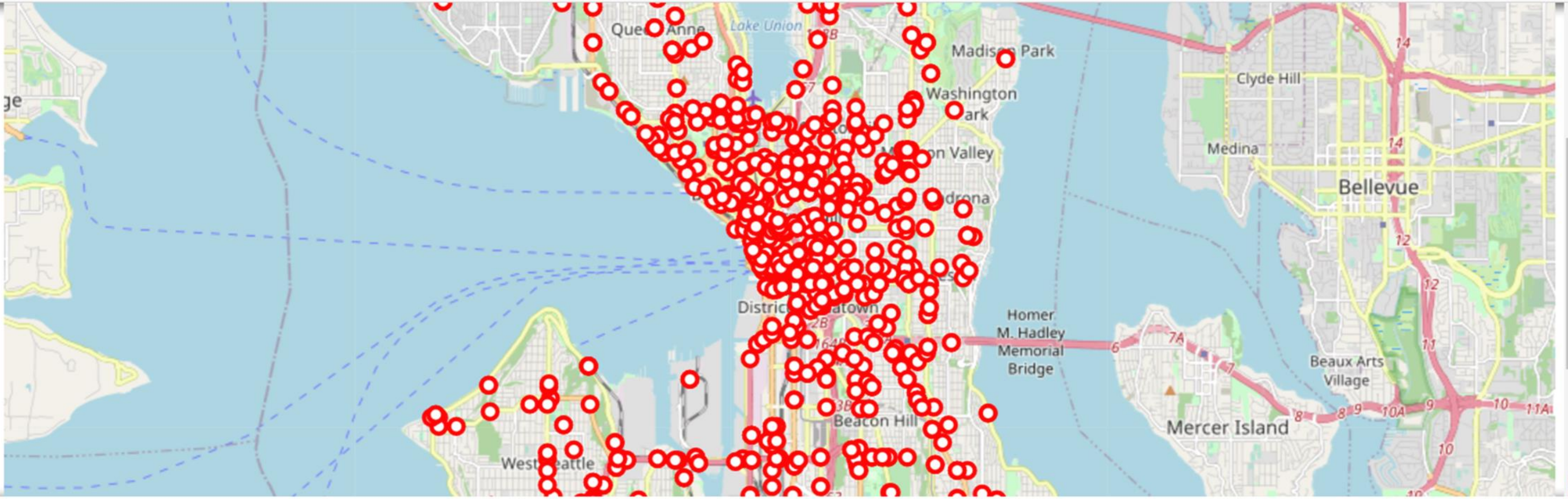
In [30]:
```
sns.heatmap(data.corr(), annot=True, vmin=-1, vmax=1, center=0, cmap='RdYlGn', linewidths=0.2, annot_kws={'size':8})
```

Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x7f5900444b38>



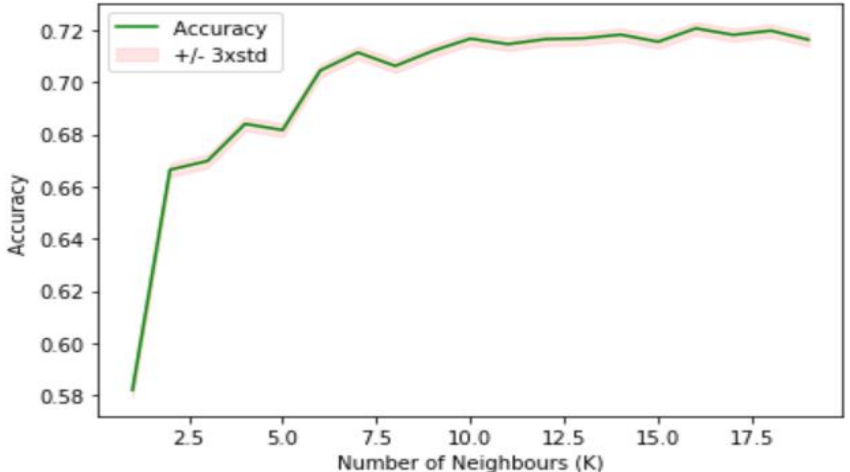
- An interactive map of Seattle is created with circle markers at the place the accidents have occurred. Markers have been plotted for the hindmost records in the cleaned dataset so that they show the more recent accident prone areas in the city.
  From the map, on zooming in, we can see that the number of accidents are higher around University Street, Westlake, Pioneer Square, Green Lake, etc. In general, they are higher in central Seattle.

```python
# add accidents to map
seattle_map.add_child(accidents)
```



Data Modelling

- In this stage, different Machine Learning models are applied to the dataset to predict the target variable i.e. accident severity.
- First, the ‘feature’ and ‘target’ sets are defined and then split into training and testing sets. 20% of the data is used for testing and 80% for training.
- The models used are Decision Trees, Support Vector Machine(SVM), K-Nearest Neighbours and Logistic Regression.
- The predicted values are displayed for the Decision Tree model.
- The best ‘K’ value is calculated and the graph of accuracy is plotted for the K-Nearest Neighbours model. The best ‘K’ was found to be 16.



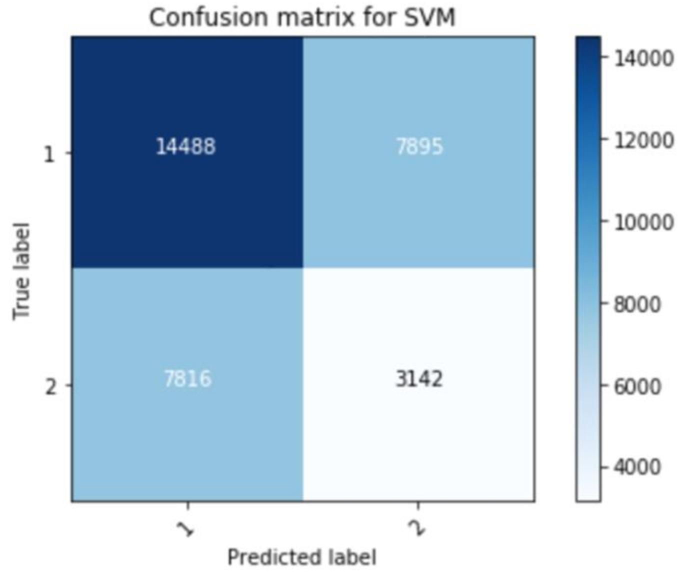
- Confusion matrices are constructed for SVM and Logistic Regression.

```
              precision    recall  f1-score   support

           1       0.65      0.65      0.65     22383
           2       0.28      0.29      0.29     10958

   micro avg       0.53      0.53      0.53     33341
   macro avg       0.47      0.47      0.47     33341
weighted avg       0.53      0.53      0.53     33341

Confusion matrix, without normalization
[[14488  7895]
 [ 7816  3142]]
```

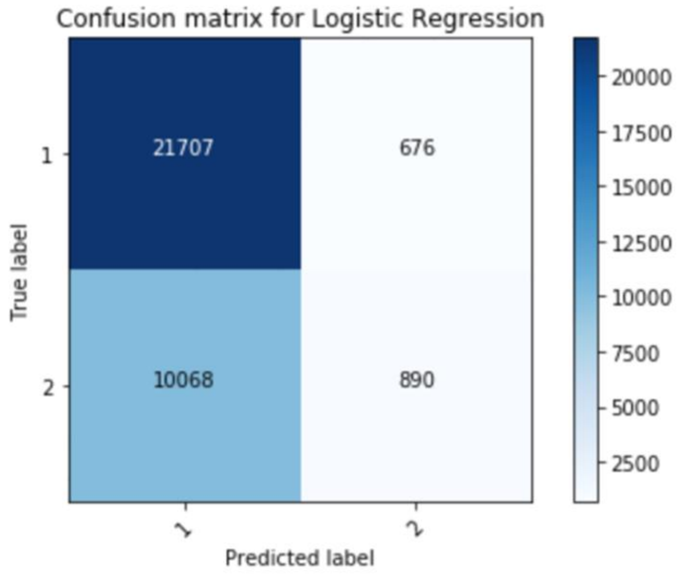



```
              precision    recall  f1-score   support

           1       0.68      0.97      0.80     22383
           2       0.57      0.08      0.14     10958

   micro avg       0.68      0.68      0.68     33341
   macro avg       0.63      0.53      0.47     33341
weighted avg       0.65      0.68      0.58     33341

Confusion matrix, without normalization
[[21707   676]
 [10068   890]]
```



Model Evaluation

- In this stage, the different Machine Learning models are compared based on their evaluation metric scores to decide which is the best for the data.
- Each model's accuracy is calculated using the Jaccard Similarity Score, F1-Score and Log Loss (only for Logistic Regression), each of which range between 0 to 1.

Algorithm	Jaccard Similarity Score	F1-score	Logloss
KNN	0.716325	0.679737	NA
Decision Tree	0.725503	0.684016	NA
SVM	0.528778	0.529208	NA
Logistic Regression	0.677754	0.584866	0.612872

Conclusion

- From the Model Evaluation stage we get the evaluation metrics for each model.
- All models can be seen to have a considerably good accuracy except the SVM model.
- The Jaccard Similarity Score and F1-Score for SVM are average as SVM is not very good with handling large datasets.
- The model with the highest Jaccard Similarity Score is Decision Trees, with a score of 0.725503
- The model with the highest F1-Score is also Decision Trees, with a score of 0.684016.
- From these scores, we can say that Decision Trees is the best model for predicting the severity of an accident based on the dataset that was made available.