Report - Hackathon

CS5590: Foundations of Machine Learning

Prof. Vineeth N Balasubramanian

Naitik Malav | CS19BTECH11026

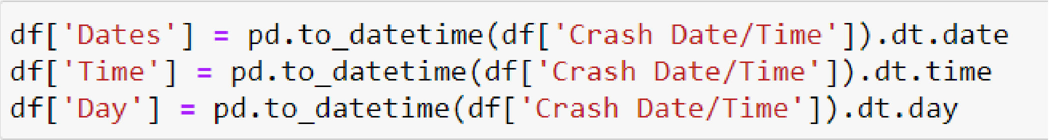
Jatin Kumar | CS19BTECH11036

Deliverables: We have submitted python notebook = Foml_Hackathon.ipynb and Report.pdf

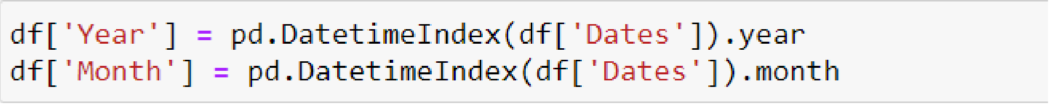
Data Pre-Processing:

1. Given csv files train.csv and test.csv are loaded into df and df_test dataframes.
2. We have parsed 'Crash Data/Time' column into Dates, Time, Year, Month, Day.
3. We have also dropped those columns which are having NaN values more than 30%
4. We have also dropped rows which are having Nan values.

Commands for parsing date and time.



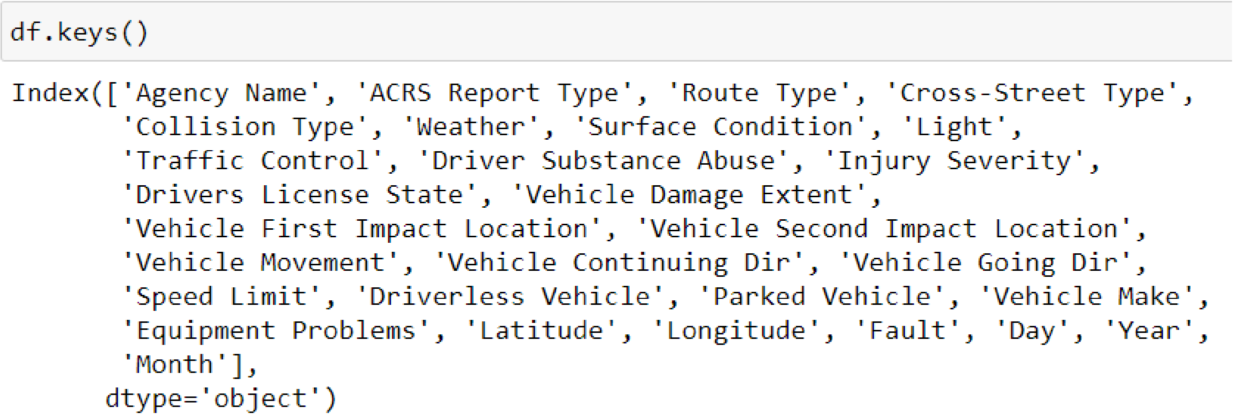
```
df['Dates'] = pd.to_datetime(df['Crash Date/Time']).dt.date
df['Time'] = pd.to_datetime(df['Crash Date/Time']).dt.time
df['Day'] = pd.to_datetime(df['Crash Date/Time']).dt.day
```



```
df['Year'] = pd.DatetimeIndex(df['Dates']).year
df['Month'] = pd.DatetimeIndex(df['Dates']).month
```

We have dropped = ['x', 'Local Case Number', 'Road Name', 'Cross-Street Name', 'Off-Road Description', 'Municipality', 'Related Non-Motorist', 'Non-Motorist Substance Abuse', 'Person ID', 'Circumstance', 'Drivers License State', 'Vehicle Year', 'Vehicle ID', 'Location']

Training dataset final columns:



```
df.keys()
```

```
Index(['Agency Name', 'ACRS Report Type', 'Route Type', 'Cross-Street Type',
       'Collision Type', 'Weather', 'Surface Condition', 'Light',
       'Traffic Control', 'Driver Substance Abuse', 'Injury Severity',
       'Drivers License State', 'Vehicle Damage Extent',
       'Vehicle First Impact Location', 'Vehicle Second Impact Location',
       'Vehicle Movement', 'Vehicle Continuing Dir', 'Vehicle Going Dir',
       'Speed Limit', 'Driverless Vehicle', 'Parked Vehicle', 'Vehicle Make',
       'Equipment Problems', 'Latitude', 'Longitude', 'Fault', 'Day', 'Year',
       'Month'],
      dtype='object')
```

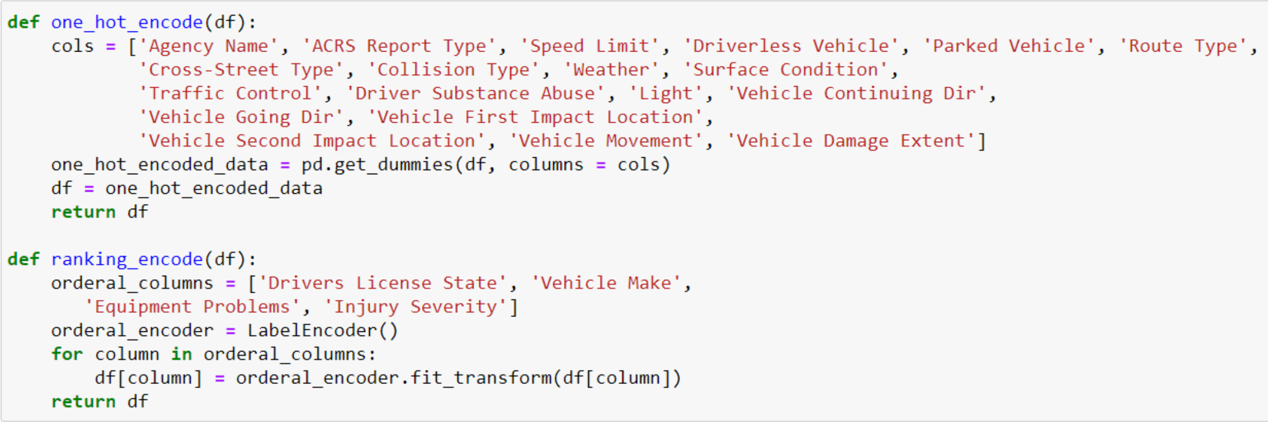
Our idea of the dropping these columns is based on the criteria of percentage nan values. We kept of the threshold 50%. As Filling those values using the remaining data would have increased the uncertainty in data. As distribution of data change drastically from the missing data.

One Hot Encoding:

[Agency Name, ACRS Report Type, Speed Limit, Driverless Vehicle, Parked Vehicle, Route Type, Cross-Street Type, Collision Type, Weather, Surface Condition, Traffic Control, Driver Substance Abuse, Light, Vehicle Continuing Dir, Vehicle Going Dir, Vehicle First Impact Location, Vehicle Second Impact Location, Vehicle Movement, Vehicle Damage Extent]

As One Hot Encoding is used for Nominal Categorial Data Type which all these columns satisfy. For example: Agency Name has name of the places where the accident was reported. it has more than 4 kinds of types. Which can be converted into columns (1,0,0,0) for a kind.

Functions used for One Hot Encoding and Label Encoding



```
def one_hot_encode(df):
    cols = ['Agency Name', 'ACRS Report Type', 'Speed Limit', 'Driverless Vehicle', 'Parked Vehicle', 'Route Type',
            'Cross-Street Type', 'Collision Type', 'Weather', 'Surface Condition',
            'Traffic Control', 'Driver Substance Abuse', 'Light', 'Vehicle Continuing Dir',
            'Vehicle Going Dir', 'Vehicle First Impact Location',
            'Vehicle Second Impact Location', 'Vehicle Movement', 'Vehicle Damage Extent']
    one_hot_encoded_data = pd.get_dummies(df, columns = cols)
    df = one_hot_encoded_data
    return df

def ranking_encode(df):
    orderal_columns = ['Drivers License State', 'Vehicle Make',
       'Equipment Problems', 'Injury Severity']
    orderal_encoder = LabelEncoder()
    for column in orderal_columns:
        df[column] = orderal_encoder.fit_transform(df[column])
    return df
```

Label Encoding:

Label Encoding is for ordeal categorial data type. These are those data types in which order of the category matters. For example: Grade of an exam. In in the given dataset columns Injury Severity, Speed Limit etc.

Model Selection:

As we are predicting binary classification. First comes in our mind was Logistic Regression which is one of the most basic model binary classification.

Accuracy using Logistic Regression – 0.73760

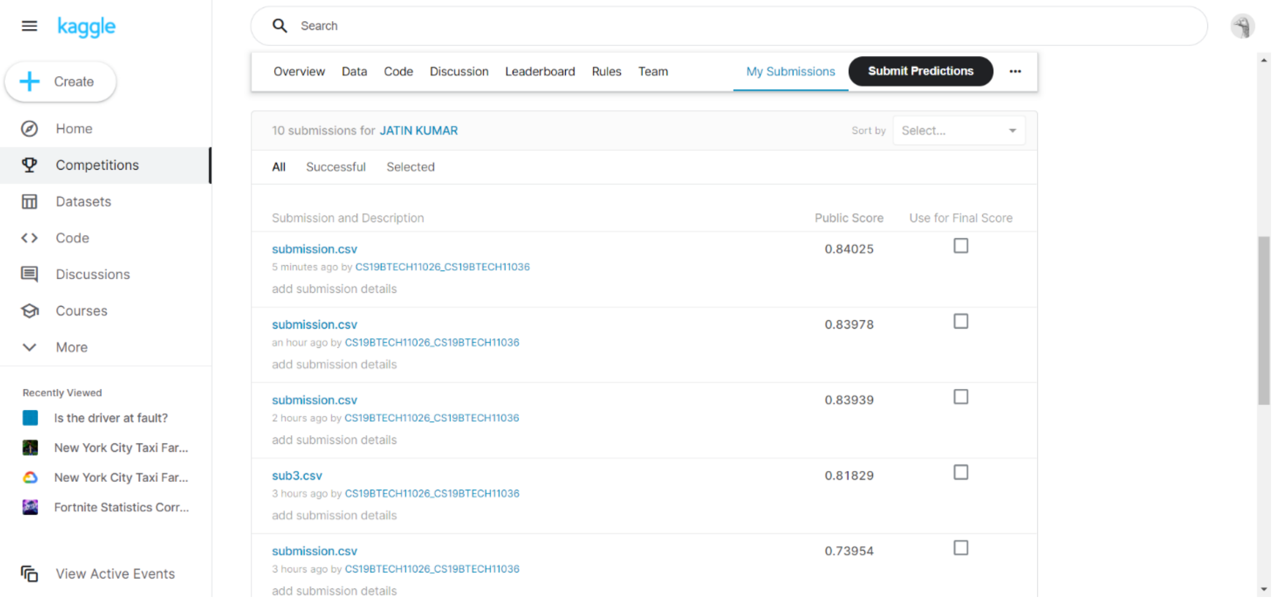
After that assuming the input features to be independent next Naïve Bayes Classifier is used, we didn't give the results we expected.

Accuracy using Naïve Bayes Classifier – 0.73954

Then several models such as Decision Tree Classifier, Ada Boost, XGBoost, the accuracy are given below:
Accuracy using Decision Tree Classifier – 0.81829
Accuracy using Ada Boost – 0.83939
Accuracy using XGBoost – 0.83934

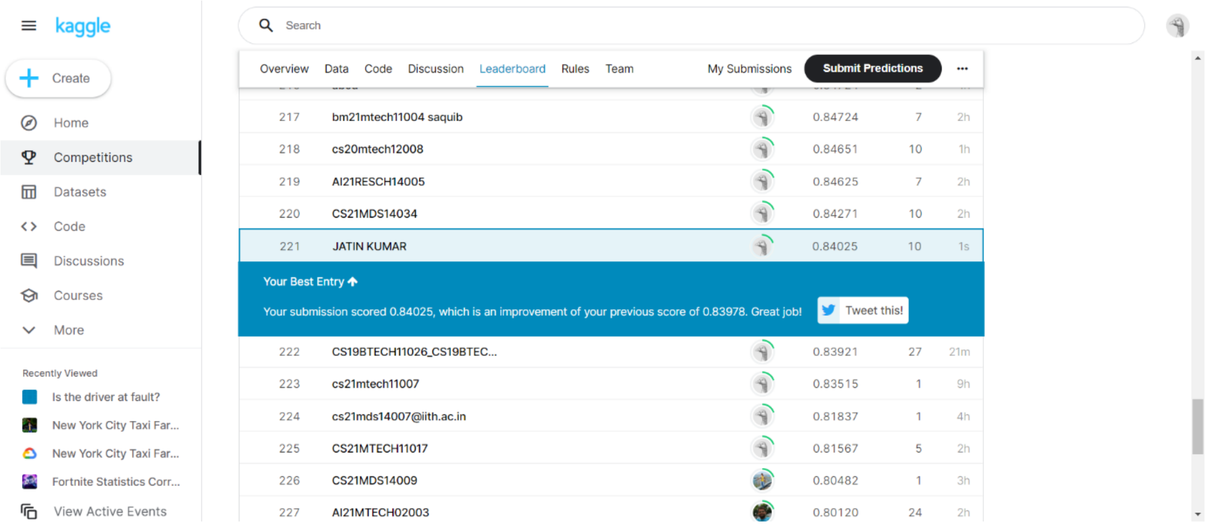
At last I have used Random Forest Classifier –
So when using no. of estimators=1000, accuracy=0.83978
And when using no. of estimators=5000, accuracy=0.84025

Below attached is the screenshot of our submitted accuracy. My friend tried to rename his username but it's not updating. So, I have attached screenshot below for proof.



10 submissions for JATIN KUMAR

Submission and Description	Public Score	Use for Final Score
submission.csv 5 minutes ago by CS19BTECH11026_CS19BTECH11036 add submission details	0.84025	□
submission.csv an hour ago by CS19BTECH11026_CS19BTECH11036 add submission details	0.83978	□
submission.csv 2 hours ago by CS19BTECH11026_CS19BTECH11036 add submission details	0.83939	□
sub3.csv 3 hours ago by CS19BTECH11026_CS19BTECH11036 add submission details	0.81829	□
submission.csv 3 hours ago by CS19BTECH11026_CS19BTECH11036 add submission details	0.73954	□



217	bm21mtech11004 saquib	0.84724	7	2h
218	cs20mtech12008	0.84651	10	1h
219	AI21RESCH14005	0.84625	7	2h
220	CS21MDS14034	0.84271	10	2h
221	JATIN KUMAR	0.84025	10	1s

Your Best Entry ↑

Your submission scored 0.84025, which is an improvement of your previous score of 0.83978. Great job!

222	CS19BTECH11026_CS19BTEC...	0.83921	27	21m
223	cs21mtech11007	0.83515	1	9h
224	cs21mds14007@iith.ac.in	0.81837	1	4h
225	CS21MTECH11017	0.81567	5	2h
226	CS21MDS14009	0.80482	1	3h
227	AI21MTECH02003	0.80120	24	2h