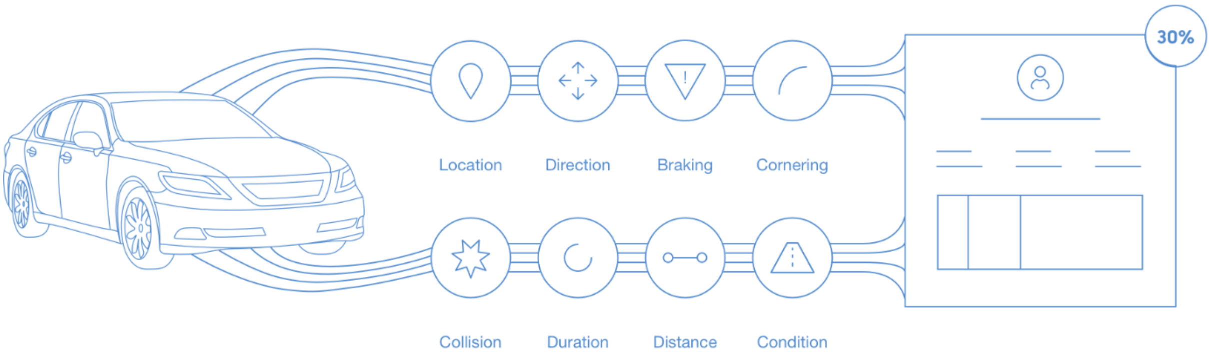
ISM6930.005F18 DSP

DATA SCIENCE PROJECT ON



DRIVER TELEMATICS

Group Project

Submitted by,
Prithvi Kocherla (#U50193006)
Ramreddy Duddela (#U62350397)

University Of South Florida

November 9th , 2018

1. Introduction:

In contemporary world, tracking information has become a wide spread phenomenon and it has transcended into every capacity revolving around our day to day activities. Here our focal point is vehicle tracking through GPS sensors and storing them to get insights into various aspects. Majorly our work dealt in observing the driving patterns of individuals and deduce various standards by comparing them.

2. Objective:

Quickly shifting gears into the aim of our project, we set to analyse the aspects derived from the driving patterns of various trips and categorize them based on different levels of risks possible.

In a nutshell, our idea is to analyse the data that we have and infer results by placing different trips into three different categories named High risk, Medium risk and Low risk.

The crux of our project is to give an idea to all the leading insurance firms about their pricing policies. Providing good customers with low premium rates can be a win-win situation for both the company and the good customers as well. Novelty of our project lies in the thought behind replacing the old premium pricing conventions which were based on vehicle characteristics and previous accidents or the number of tickets issued to an individual.

Presenting this as a concept to customers can benefit them too to self-retrospect and get better at driving and even more less prone to accidents.

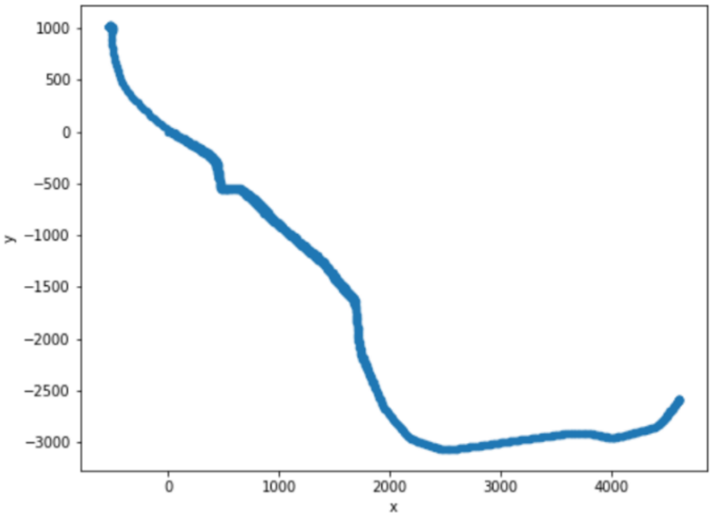
3. Data Exploration

➢ Data Source

There was a competition held by AXA (UK based insurance company) on Kaggle, which provided data for 30 individual drivers where each of them took 200 trips.

Each trip consists of x and y co-ordinate positions of vehicle noted for every 1 second. Due to identity protection of drivers, the co-ordinates have been shifted to graph plots with starting position being at origin (0,0).

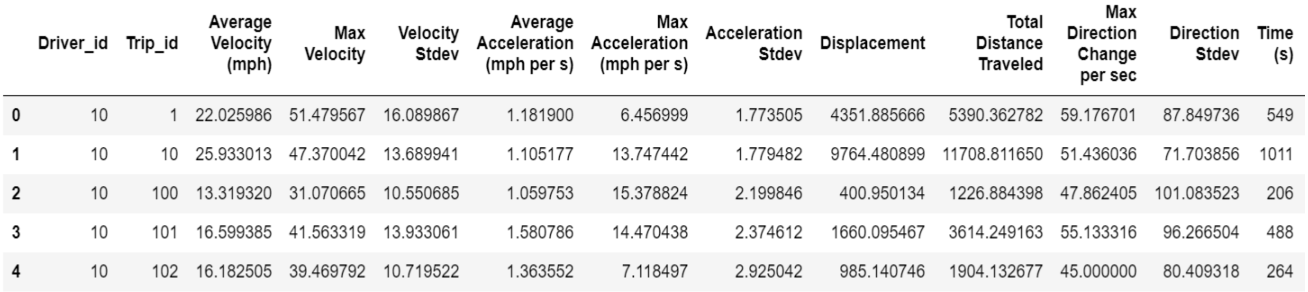
Out of the available sample trips, one trip is presented here, plotted the pattern on graph.



- **Dataset with Features**

There was also a dataset provided on Kaggle, which is the reference for this project with 17 features (*Average Velocity, Maximum Velocity, Velocity Stdev, Average Acceleration, Max Acceleration, Acceleration Stdev, Displacement, Direction Change, Direction Stdev, Time, Turns, Aggressive Turns, Stops, Deceleration, N/ of Decelerations, Max Deceleration, Total Distance*) that are formed from initial X, Y co-ordinate dataset.

It is grouped by Driver ID's and Trip ID's. Each row in this dataset represents the average value of the features calculated from each trip taken. Below shown is a sample of this dataset with features. It has data for 23 drivers with 200 trips, hence total of 4600 rows are available.



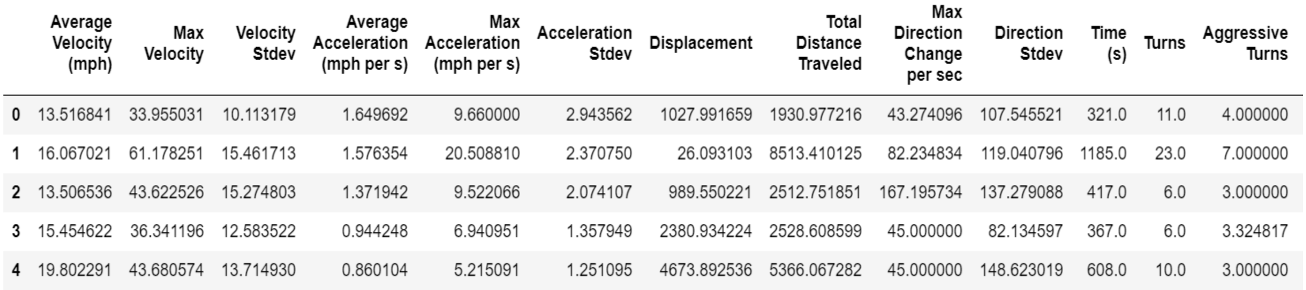
	Driver_id	Trip_id	Average Velocity (mph)	Max Velocity	Velocity Stdev	Average Acceleration (mph per s)	Max Acceleration (mph per s)	Acceleration Stdev	Displacement	Total Distance Traveled	Max Direction Change per sec	Direction Stdev	Time (s)
0	10	1	22.025986	51.479567	16.089867	1.181900	6.456999	1.773505	4351.885666	5390.362782	59.176701	87.849736	549
1	10	10	25.933013	47.370042	13.689941	1.105177	13.747442	1.779482	9764.480899	11708.811650	51.436036	71.703856	1011
2	10	100	13.319320	31.070665	10.550685	1.059753	15.378824	2.199846	400.950134	1226.884398	47.862405	101.083523	206
3	10	101	16.599385	41.563319	13.933061	1.580786	14.470438	2.374612	1660.095467	3614.249163	55.133316	96.266504	488
4	10	102	16.182505	39.469792	10.719522	1.363552	7.118497	2.925042	985.140746	1904.132677	45.000000	80.409318	264

4. Data Pre-processing

Since, above data contains Driver ID and Trip ID and as these columns do not play any important role in analysis, they have been removed. The dataset is then shuffled so that training and testing data would be randomised that helps for analysis.

Further, it is observed that this data has some null values in columns like Turns, Stops, Aggressive Turns etc., they have been replaced with mean values of that respective feature.

Below shown is the final modified dataset which is ready for our analysis.



	Average Velocity (mph)	Max Velocity	Velocity Stdev	Average Acceleration (mph per s)	Max Acceleration (mph per s)	Acceleration Stdev	Displacement	Total Distance Traveled	Max Direction Change per sec	Direction Stdev	Time (s)	Turns	Aggressive Turns
0	13.516841	33.955031	10.113179	1.649692	9.660000	2.943562	1027.991659	1930.977216	43.274096	107.545521	321.0	11.0	4.000000
1	16.067021	61.178251	15.461713	1.576354	20.508810	2.370750	26.093103	8513.410125	82.234834	119.040796	1185.0	23.0	7.000000
2	13.506536	43.622526	15.274803	1.371942	9.522066	2.074107	989.550221	2512.751851	167.195734	137.279088	417.0	6.0	3.000000
3	15.454622	36.341196	12.583522	0.944248	6.940951	1.357949	2380.934224	2528.608599	45.000000	82.134597	367.0	6.0	3.324817
4	19.802291	43.680574	13.714930	0.860104	5.215091	1.251095	4673.892536	5366.067282	45.000000	148.623019	608.0	10.0	3.000000

5. Data Analysis

The considered data here has no dedicated output column, because of this the data falls under unsupervised learning and requires obtaining a column which represents as output. This is planned to achieve by grouping the data into clusters and labelling it by a sequence number.

➢ K-Means Clustering

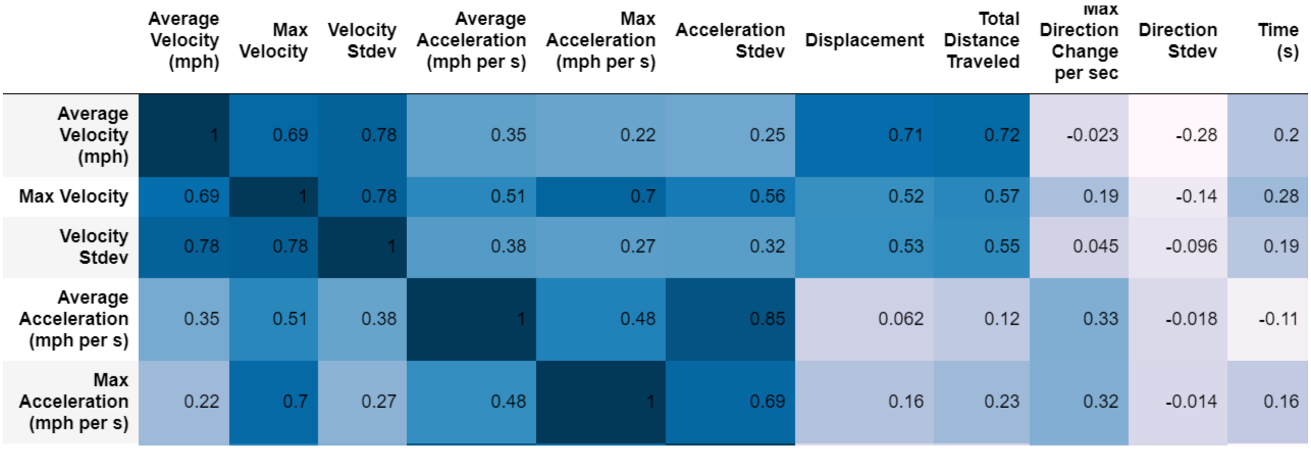
One of the best-known algorithm in clustering is K-means, in which it aims to divide the data into clusters by initially choosing a random centroid and then distributing the rows across the nearest centroids. The mean is then updated with the formed cluster data and the centroids are gradually moved resulting in a minimised sum of squared error value in between points and centroids.

In this algorithm, there is a provision to decide the number of clusters that can be formed and here number of clusters considered are 3, with an idea to distinguish the driving patterns as Low Risk/ Medium Risk/High Risk.

The outcome of this clustering results in an array having values of either 0/1/2 generated for each row. This ideally represents the output column, so it is now added to our cleaned dataset as last column.

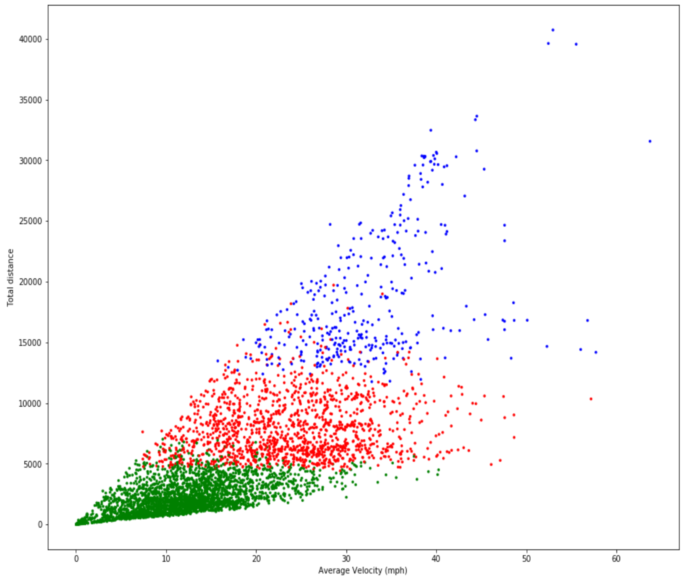
Correlation among features:

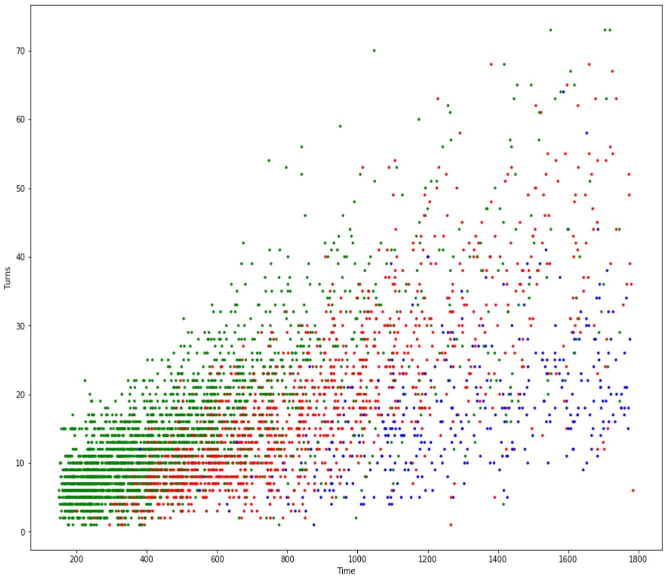
As there are many features, it is important to identify the correlation existing among themselves to formulate base decisions. By this, the least correlated features can be ignored.



	Average Velocity (mph)	Max Velocity	Velocity Stdev	Average Acceleration (mph per s)	Max Acceleration (mph per s)	Acceleration Stdev	Displacement	Total Distance Traveled	Max Direction Change per sec	Direction Stdev	Time (s)
Average Velocity (mph)	1	0.69	0.78	0.35	0.22	0.25	0.71	0.72	-0.023	-0.28	0.2
Max Velocity	0.69	1	0.78	0.51	0.7	0.56	0.52	0.57	0.19	-0.14	0.28
Velocity Stdev	0.78	0.78	1	0.38	0.27	0.32	0.53	0.55	0.045	-0.096	0.19
Average Acceleration (mph per s)	0.35	0.51	0.38	1	0.48	0.85	0.062	0.12	0.33	-0.018	-0.11
Max Acceleration (mph per s)	0.22	0.7	0.27	0.48	1	0.69	0.16	0.23	0.32	-0.014	0.16

From above correlation values, we can now visualize the clusters among highly correlated feature pairs.



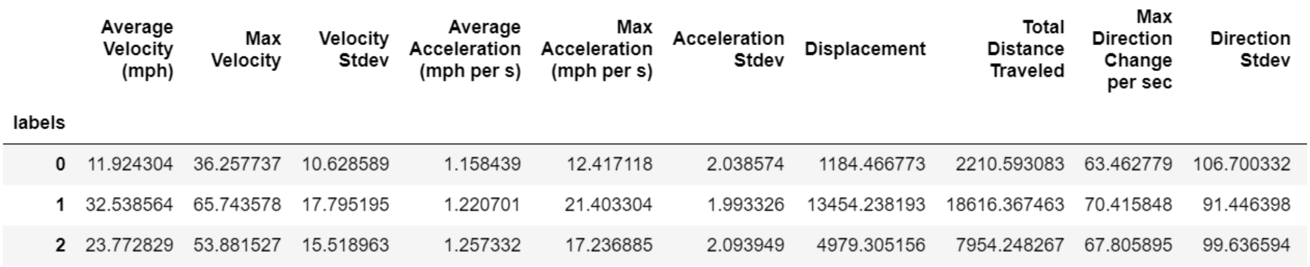


The first plot shows the cluster pattern formed by considering columns Average Velocity vs Total Distance with correlation value as 0.72.

Similarly, the second plot shows the cluster pattern formed by considering columns Time vs Turns with correlation value as 0.68.

In similar way, these patterns can be visualized based on considered features where in each case the rows fall into one of the groups differentiated by colours.

Now, as we have all the rows identified with outcome values, grouping by them by considering the average values would result as below, on broader sense a decision can be made by comparing new test values with the below obtained ones.



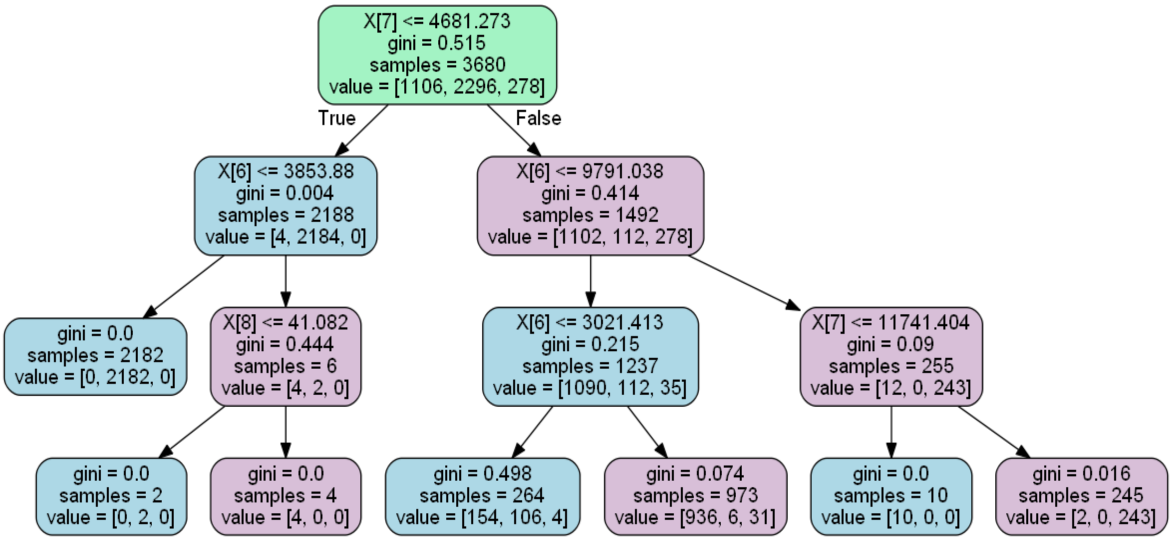
	Average Velocity (mph)	Max Velocity	Velocity Stdev	Average Acceleration (mph per s)	Max Acceleration (mph per s)	Acceleration Stdev	Displacement	Total Distance Traveled	Max Direction Change per sec	Direction Stdev
labels										
0	11.924304	36.257737	10.628589	1.158439	12.417118	2.038574	1184.466773	2210.593083	63.462779	106.700332
1	32.538564	65.743578	17.795195	1.220701	21.403304	1.993326	13454.238193	18616.367463	70.415848	91.446398
2	23.772829	53.881527	15.518963	1.257332	17.236885	2.093949	4979.305156	7954.248267	67.805895	99.636594

6. Data Modelling

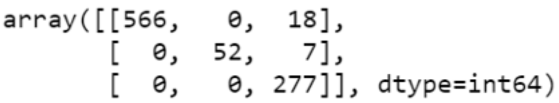
➢ Decision Tree Classifier

One of the algorithms, that helps to build a tree classification by breaking down the dataset into smaller subsets with Decision nodes and Leaf nodes is Decision Tree. It predominantly identifies a root node, that corresponds to the best prediction. Decision Trees can handle both categorical and numerical type of data.

The following tree is obtained when Decision Tree Classifier was ran on Driver data. Here, the root node corresponds to, X [7] -> Total Distance, X [6] –> Displacement & X [8] –> Max Direction Change/sec



Confusion Matrix:



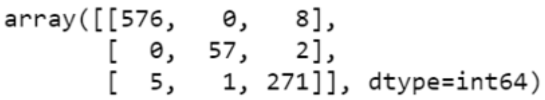
```
array([[566,   0,  18],
       [  0,  52,   7],
       [  0,   0, 277]], dtype=int64)
```

Accuracy: 0.9728

- **Random Forest Classifier**

In the case of Decision Tree, at any time there is one tree considered and decision is made, where as if considered Random Forest, it builds up some set of regression trees and out of which selects one tree that has the highest case predictive accuracy and controlling over-fitting.

Confusion Matrix:



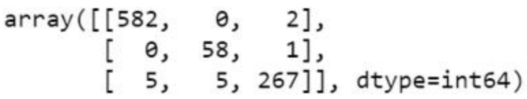
```
array([[576,   0,   8],
       [  0,  57,   2],
       [  5,   1, 271]], dtype=int64)
```

Accuracy: 0.9826

- **Support Vector Classifier**

As now, we have set up our data to be in the form of supervised learning, we can predict new test values with the help of SVC. In this project, SVC is used with polynomial kernel. The advantage with SVC is that we can incorporate polynomial features by setting up degree attribute and C value.

Confusion Matrix:



```
array([[582,   0,   2],
       [  0,  58,   1],
       [  5,   5, 267]], dtype=int64)
```

Accuracy: 0.9858

Inference:

From above results, we can see that obtained high accuracy values could be of reason that the data that was used for clustering is also used for training and testing. This may not be realistic when checking performance with real time observations. Hence, a sort of further measurement is required to test these model performances.

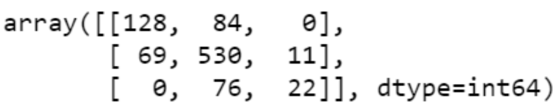
For this, a new column added which is ‘Percentage of Aggressive Turns’. It is obtained by dividing Aggressive Turns with Total Turns, hoping that this would play a role in detecting risk driving patterns.

Analysis on modified dataset:

After above mentioned data modifications, the same code is re-used again for pre-processing and model analysis. The measured values on our models are as follows,

- **Decision Tree Classifier**

Confusion Matrix:

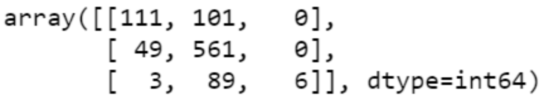


```
array([[128,  84,   0],
       [ 69, 530,  11],
       [  0,  76,  22]], dtype=int64)
```

Accuracy: 0.734

➢ Random Forest Classifier

Confusion Matrix:

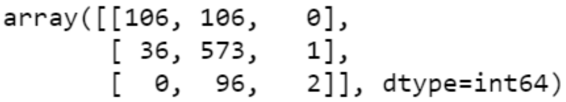


```
array([[111, 101,   0],
       [ 49, 561,   0],
       [  3,  89,   6]], dtype=int64)
```

Accuracy: 0.7369

➢ Support Vector Classifier

Confusion Matrix:



```
array([[106, 106,   0],
       [ 36, 573,   1],
       [  0,  96,   2]], dtype=int64)
```

Accuracy: 0.7401

Inference:

From these values, it is obvious that the accuracy values have dropped when compared to previous analysis. But the later data is much more realistic. When compared among models, decision tree accuracy is relatively less than to other two models. SVC and Random forest results are comparatively same. Hence, these models stand as a base in deciding our risk factors and can ideally be used as well.

Result:

We set out to ride this project to a finish line by converging our idea, objective and analysis, we categorized the data set into three different risk levels as already discussed. The primary intent behind taking up this project is to present the results and suggest the leading insurance firms to bring a change in their premium pricing conventions. These results can also be an eye opener for people too, to get better at driving, aware about safety and less prone to accidents. If our concept is considered on an execution level by the firms, it would be a game for both the vendor and the customer.

References:

1. Cover page Image
   https://www.octotelematics.com
2. K-means clustering in python
   http://benalexkeen.com/k-means-clustering-in-python/
3. Initial Dataset with features –
   https://github.com/georgetown-analytics/skidmarks/blob/master/bin/lin.csv
4. Kaggle competition work –
   http://nbviewer.jupyter.org/github/georgetown-analytics/skidmarks/tree/master/bin/
5. Decision Tree Example –
   https://pythonprogramminglanguage.com/decision-tree-visual-example/
6. https://www.tandfonline.com/doi/abs/10.1080/01431160412331269698
7. https://www.saedsayad.com/decision_tree_reg.htm