

Driver Usage Risk Profiling by Analyzing Vehicle Driving Behavior using Machine Learning Model Based on Vehicular Cloud Telematics Data

MSc Research Project
Cloud Computing

Anuj Kumar
Student ID: X17157641

School of Computing
National College of Ireland

Supervisor: Victor Del Rosal

National College of Ireland
Project Submission Sheet
School of Computing



Student Name:	Anuj Kumar
Student ID:	X17157641
Programme:	Cloud Computing
Year:	2018
Module:	MSc Research Project
Supervisor:	Victor Del Rosal
Submission Due Date:	20/12/2018
Project Title:	Driver Usage Risk Profiling by Analyzing Vehicle Driving Behavior using Machine Learning Model Based on Vehicular Cloud Telematics Data
Word Count:	5921
Page Count:	21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	19th December 2018

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Driver Usage Risk Profiling by Analyzing Vehicle Driving Behavior using Machine Learning Model Based on Vehicular Cloud Telematics Data

Anuj Kumar
X17157641

Abstract

This paper is a novel approach to assist the vehicle insurance companies for calculating the annual insurance premium cost for the vehicle owner by developing Vehicle Prediction System (VPS) using Vehicular Cloud and Machine learning Models. By the development of new technologies in recent years, the potential to handle the Vehicle sensing data is increased. This Proposed system is using real-time operational information from a vehicle including engine Revolutions per minute (RPM), vehicle speed, throttle position, Mass Air Flow (MAF), engine load, time advance through On-Board Diagnostic (OBD) interface. By utilising Cloud infrastructure and Machine learning algorithms, the proposed system will make use of vehicle data to create a drive behaviour prediction model to calculate the risk profile of insurance policyholder. Using Cloud Computing technology, a massive amount of Telematics data can be handle generated by these vehicular sensors and Machine learning algorithms can be used to create a risk profile of the vehicle owner by measuring the driving behaviour. This proposed model is using Unsupervised and Supervised machine learning methods to measure the driving behaviour. This platform can be beneficial for the policy subscriber as this prediction model will be used by the insurance companies to improve the User-based insurance (UBI) and Pay-as-you-drive (PAYD) pricing model by offering lucrative price to their insurance subscribers.

1 Introduction

Handling a Massive amount of data generated by the Internet of Things (IoT) devices requires a efficient platform. We require an adequate platform to analyse and process this amount of data. Cloud Infrastructure is an efficient platform to handle this significant volume of data by utilising its availability and scalability featureSimmhan et al. (2013).

Traditional insurance premium calculation models, the insurance companies are using information related to the drivers instead of vehicle condition and driving behaviour to charge their customersTroncoso et al. (2011). In this proposed paper, we are discussing numerous sensors installed in the vehicles to get the real-time data of vehicle health to analyse and use it for driving behaviour prediction model.

This paper aims to develop a driving prediction model for the insurance companies to charge their customers by their driving behaviour instead of being charged with a fixed amount. Currently, there are very few companies which are charging their customers using

User Based Insurance (UBI) and Pay-How-you-Drive (PHYD) pricing model. Insurance companies should conspire with information technology (IT) to apply digital marketing approach through which they can attract more consumers by providing benefits like lowering the premium cost to their existing and future customers along with earning profits.

All vehicle drivers have a different style of driving such as how to accelerate, decelerate, braking and different ways to use vehicle control devices. We are developing the prediction model to use these vehicle sensors data for driving behaviour analysis using cloud technology as its platform Amsalu et al. (2015) for the insurance companies. The Proposed Machine learning model consists of three major components 1) On-Board-diagnostic (OBD) device: To collect the vehicle sensor telematics data from the vehicle. OBD port is available in all the vehicles to fetch all the sensors data Takefuji (2018). 2) Cloud infrastructure: Using cloud infrastructure to analyse and process massive amount because of its high scalability and availability. 3) Vehicle prediction server (VPS): The proposed machine learning model is running on a server build on cloud infrastructure. For Driving behaviour analysis, the Proposed model is using six different attributes to measure the driving behaviour like Engine Throttle position, Engine load, MAF, Speed, RPM, and Time Advancing. These values are selected after calculating the correlation between RPM and all other attributes.

There are some driving prediction models, where the author is using Supervised machine learning techniques with the combination of mobile and vehicle sensor data Zhang et al. (2016) Meseguer et al. (2013). Additionally, there are some models, which are using Unsupervised machine learning techniques Van Ly et al. (2013). This proposed model,

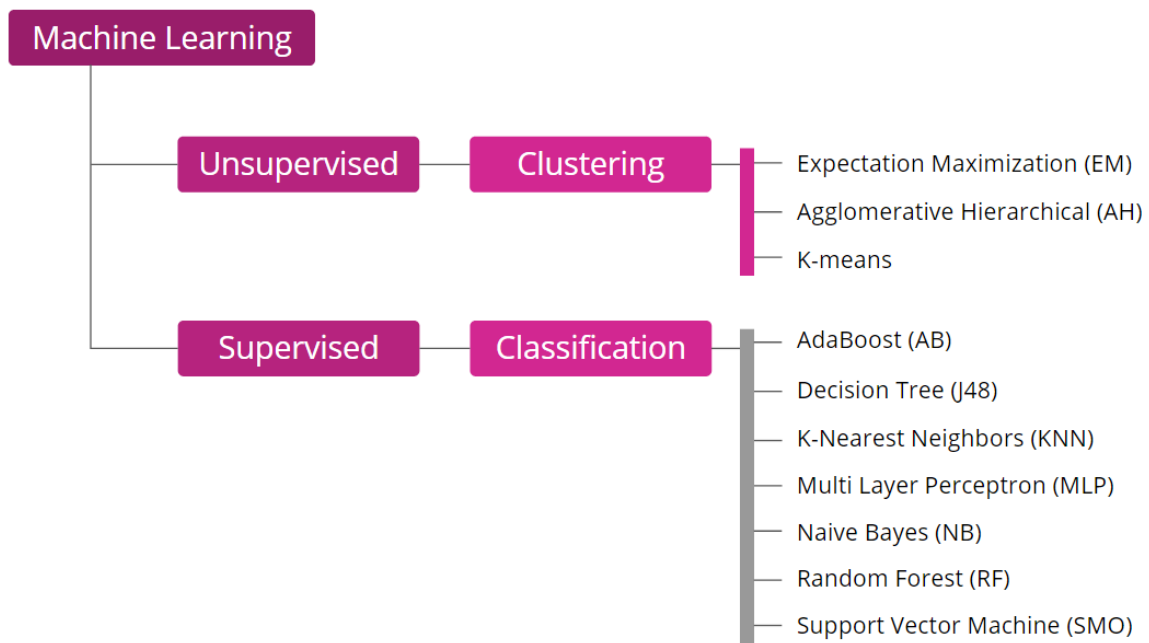


Figure 1: Machine Learning methods used

is using both supervised and unsupervised machine learning techniques mentioned in Figure 1 to analyse the driving behaviour for the insurance companies.

Unsupervised Machine learning: EM: This Technique is used to find the maximum similarity of function, AH: This technique forms a treelike structure by creating partition sequence where a single group is sub-divided into multiple groups, K-means: This technique uses centroid, which is the average value of an instance in a group. It uses Euclidean distance to measure the similarity between the instancesHastie et al. (2009).

Supervised Machine learning: AdaBoost (AB): It works on the base algorithm, and it iteratively improves the classification in training set, Decision tree (J48).A decision tree is a technique which forms a tree-like structure to sort new instances, K-Nearest Neighbors (KNN); this technique provides a standard class in training set for classification, Naive Bayes (NB). This technique is based on Bayes theorem and doesn't provide the same result for the same base, Multi-Layer Perceptron (MLP). It works on the basis of the nervous system, having layers of neurons with a strong connection, Random Forest (RF): This technique forms a large number of decision trees to find the best possible way for result, Support Vector Machine (SVM): This Technique is base on Statical methods Kumari and Godara (2011);Rätsch et al. (2001).

2 Related Work

This Section outlines previous work happened in the area of Telematics, the Insurance industry and Machine Learning models. The main focus of this purposed prediction system is to improve the Insurance premium pricing model for the insurance business by combining different technologies. Additionally, to reduce the flaws in the existing inefficient and unfair traditional insurance premium criteria such as charging the individual user with a fixed or lump sum amount by covering total kilometres driven by the vehicle owner irrespective of Driving behaviorButler et al. (1988) Parry (2004). In below mentioned subsection 2.1 we are discussing a few previous insurance risk profiling models using telematics data.

2.1 Telematics Data usage for insurance Risk profiling.

The Major focus of Insurance market is to modify the Pay-As-You-Drive (PAYD), Pay-How-You-Drive (PHYD) and Usage-Based Insurance (UBI) pricing model to utilise the driving behaviour pattern information of an individual policy owner to decide their premium cost Boquete et al. (2010). There are multiple mobile and tablets applications available in the market like Aviva RatemyDrive ¹, State Farm driver feedback ² to score an individual driving behaviour by using different technologies like GPS, Magnetometers accelerometers. These applications are using the driving score to provide 20% of discount for the insurance premiumCastignani et al. (2015).

Eren et al. (2012) developed an algorithm to distinguish between safe and risky drivers having an accuracy of 93.3% by evaluating data from 15 iPhone User with the fixed point of the journey. The Author is using empirical thresholds and moving algorithm to detect

¹Aviva PLC.(2013) Aviva RateMyDrive:<http://www.aviva.co.uk/drive/>

²Statefarm DriverfeedbackState Farm Mutual Automobile Insurance Company.(2013): <http://www.statefarm.ca/about/mobile/>

driving events like aggressive steering, acceleration, braking etc via gyroscope, magnetometer and smooth acceleration data from smartphones. This paper is using Bayesian classification and Dynamic Time Warping (DTW) to differentiate between safe and risky drivers by using template data (Similar patterns collected for safe and risky events).

The future work is to Paefgen et al. (2012) utilise the Smartphone data mainly for insurance companies to predict the driving behaviour by manually using user setting and the direction of the vehicle after the calibration process and start collecting data such as braking, Steering events and acceleration. By defining, a threshold value for sensing data (like 0.2g for steering and 0.1g for acceleration), the events are trigger and compare the event detection data by a smartphone with the values of telematics boxed fixed in the vehicle on the Inertial Measurement Unit (IMU). According to the observation of the author due to the variation in smartphone position in the vehicle, the event count is match by distinct analytical distribution. However, the paper, the author provided some error sources and correlation between that IMU and smartphone-based events

Later on, many insurance premium calculation models are introduced like Baecke and Bocca (2017) propose PAYD pricing model using the Internet of Things (IoT) Device to create risk profile of insurance policyholder by analysing the driving behaviour. Tselentis et al. (2016) propose combination of PAYD and PHYD .In 2016 Nai et al. (2016) propose UBI pricing model for risk analysis of insured vehicle by Fuzzy Risk Mode and FRAME method (i.e. effect Analysis) where the telematics data is provided to the Expert team to analyse and provide a grade to the vehicle according to the basic risk analysis Bian et al. (2018).

2.2 Existing Cloud Implementation to process OBD telematics Data.

In this subsection, discussion is about the existing approach to process OBD telematics data to measure the driving behavior for the insurance policy subscriber using cloud technology as their platform. Initially OBD device were locally install in vehicles to measure the health of vehicle. By the advent of technology ,current OBD system is improve by the on-line feature to fetch the real-time vehicular data for analysis. Iqbal and Lim (2006) propose a Global Positioning System (GPS) based insurance calculation system ;this proposed insurance model was base on mobility of vehicle that includes total kilometers driven by vehicle, Zone of traveling, time of traveling and average speed to measure the risk profile of vehicle owner.

Different attributes used by Jhou et al. (2013) to reduce the time to detect the fault in vehicle in-case of any breakdown. In this proposed, cloud-based detection model, all the dynamic real-time information like Engine RPM, Coolant temperature, fault codes, vehicle speed were used. The information is received on cloud server called Vehicle Diagnostic Server (VDS) over a 3.5 wireless network for the analysis. The cloud server is an online expert system providing a solution in-case of any breakdown by analysing all the parameters using statical algorithms.

The early fault detection system was lately improved by Amarasinghe et al. (2015) the author is using the Android mobile application to connect the OBD device with a

server running on Cloud infrastructure. This application was utilising mobile data for its communication.

2.3 Different Technique used for measuring Driving Behavior

Araújo et al. (2012) developed a smart-phone application to analyse the Driving behaviour and help the drivers by suggesting them the better way to drive. The main aim of this application was to reduce fuel consumption by improving the driving style. This application was using different vehicle sensors like vehicle Speed, Engine RPM, Throttle signal etc to measure the driving behaviour.

Similarly, Meseguer et al. (2013) also used Speed, RPM, Acceleration data for analysing the driving behaviour. The author of this paper used Multilayer perceptron (MLP) algorithm for the classification of data, and the approach of this paper is considering adding the road conditions as well to analyse the driving behaviour. Zhang et al. (2016) proposed a platform using Support Vector Machine (SVM) algorithm that uses both smart-phone and vehicle sensors to analyse the driving behaviour by Support-Vector-Machine (SVM) classifier.

2.4 Discussion

In the above literature review section 2, we have discussed different studies using only Mobile sensor and Only vehicle sensors to analyse the driving behaviour also we have discussed some papers, the author is using both mobile and vehicle sensor to analyse the driving behaviour. Some of the studies are utilising cloud infrastructure as their platform, and the outcomes of most of the studies are using vehicle sensors provides better accuracy.

We have discuss some of the studies using Unsupervised and supervised machine learning techniques to analyse the data. The purpose of those studies was different for this proposed model. Most of them analysed the driving behaviour to reduce the CO2 emission, to analyse risky and safe driver, Reducing the fault analysis time in case of breakdown and reducing the fuel consumption by proving better suggestion based on the vehicle driving behaviour.

This proposed paper is using both machine learning techniques, i.e. Unsupervised and Supervised. Unsupervised techniques are use for clustering the data without a class label and supervised techniques for the classification of data Michalski et al. (2013).

3 Methodology

The chief objective of this research paper is to develop a prediction system/model for analysing the driving behavior to calculate the cost of insurance subscription. In this section, we are discussing about all the steps involved to develop the proposed vehicle driving prediction model. Below are the steps mentioned involved:

1. **Data Procurement**
2. **Data Preprocessing**

3. Apply Machine Learning Methods and Post Processing

- (a) Apply Clustering Methods over Dataset
- (b) Evaluate Partitions
- (c) Apply Classification Methods over Best Partitions
- (d) Evaluate Results from Classification Methods
- (e) Relabel Best Partition

4. Build Model According to Step 3

5. Develop cloud application

These steps are base on methodology defined by Barreto (2018) for build machine learning model with clustering and classification methods and some steps for validation at final of using of ML methods. The Figure 2 shows the methodology used and steps defined to it.

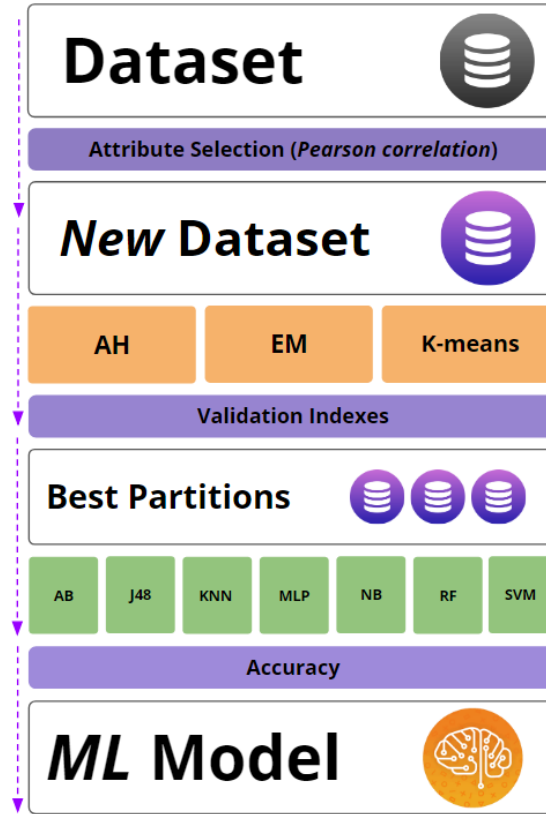


Figure 2: Methodology used - based on Barreto et al. (2018)

3.1 Data Procurement

The proposed model is using Dataset from Kaggle.com³ with the permission of dataset owner. This dataset is holding vehicle telematics raw data from 19 different drivers on a single vehicle on a single route. Dataset is having total 28 different column and 8261 rows.

³<https://www.kaggle.com/cephasax/obdii-ds3>

3.2 Data Preprocessing

This subsection is carried all the steps involved to fetch out the required information in a specific format from the raw information. The Dataset includes different 28 vehicle sensor data like Longitude, Latitude, Altitude, Fuel Level, Vehicle Id, Engine Coolant Temperature, Barometric Pressure etc. Using all the attributes is not required for the proposed model, selection of few attributes for better accuracy is necessary. below are the steps involved in clearing the data and selection of different attributes.

3.2.1 Fixing Values.

After we have Data for vehicle sensors, we need to check and verify all the values if there are any missing, noisy or outliers values in the dataset. In the proposed model is using Mean strategy to fill all the missing values for all the attributes used.

1. Using MS Excel, we are calculating the mean values by using Minimum and maximum values for all the columns.
2. These Mean values are used to replace the missing values.

3.2.2 Normalization of Data.

After fixing, the missing values for all the attributes used, we need to normalise the data for better accuracy and avoiding the redundancy.

$$Normalize \quad value = \frac{(DataToNormalize - Minimum)}{(Maximum - Minimum)} \quad (1)$$

3.2.3 Calculate the Correlation

The proposed model is using the Pearson method for fetching the correlation between RPM and other different attributes to check the relationship between them; how the values are changing concerning the RPM.

3.2.4 Attributes Selection

Based on the correlation values, six different attributes to be processed by a proposed model including RPM values using MS Excel Filter option.

3.2.5 Building a new Dataset

After performing all the steps mentioned above, new data is formed to be used by the prediction model. That dataset is name as “**anuj_norm_6**” for this research proposals..

3.3 Apply Machine Learning Methods and Post Processing

This sub-section includes all the steps involved while building the machine learning model like clustering of data, classification of clusters and re-labeling the data.

3.3.1 Apply Clustering Methods over Dataset

No class label available in the dataset, Unsupervised Machine learning technique is used for clustering the data. An open-source data mining tool Weka⁴ is used for this process. The proposed model is using three different clustering techniques: Expectation Maximization (EM), Agglomerative Hierarchical (AH) and K-Means. All of these clustering techniques are necessary to perform for clustering the data with different configurations such as a number of clusters varies from 2 to 11 ($K=2,3,11$). Beyond this configuration EM, and K-means are probabilistic techniques. It is necessary to use different numerical seeds (proposed model is using five different seeds). As a result of this part of work, we had 110 partitions: EM (50), AH(10) and K-means(50).

3.4 Evaluate Partitions

After using clustering techniques, we are using Silhouette and Davies Boldin (DB) indexes to evaluate the quality of the cluster. To measure, the better quality of the cluster silhouette index value should be near to 1 but inverse for Davies Boudin, the index value should be near to 0 Wiwie et al. (2015). It is observable in Figure 3, the best partitions index results (for every clustering methods) are for cluster value equals to 2. With this value for K, the Silhouette index reaches the value 0,468 for EM method (average), and the DB reached 0,379 for the same method (average).

Based on the results supported by the depreciation of the index values that with the increase of k, EM (with $k = 2$ and seed = 11) is chosen as the best partition. Similarly, the other two partitions are select as input for the next part of the research are based on the same index result values. These other two partitions are K-means (with $k = 2$ and seed = 11) and AH (with $k = 2$). These three partitions are used based on the result in Figure 3 for the next phase of work.

3.4.1 Apply Classification Methods over Best Partitions

Based on Silhouette and Davies Boudin Index Values, the best three partitions are chose for further classification. Seven different methods made the classification: Ada Boost (AB), Decision Tree (J48), K-Nearest Neighbors (KNN), Naive Bayes (NB), Multi-Layer Perceptron (MLP), Random Forest (RF) and Support Vector Machine (SVM).

For all classification methods used, five different approaches are applied for subdividing the dataset into two partitions (training Set and Test Set). Approach for subdividing the dataset are : 90%/10%, 75%/25%, 50%/50%, 66%/34% and 10 Cross-fold Validation. These percentages are the division of dataset into training and test set for example that 90%/10% is a divide in 90% of the dataset are use for training and 10% for the testing. The approach 10 Cross fold means that the dataset is divided in 10 parts and 1/10 is used for test and 9/10 for training.

3.5 Evaluate Results from Classification Methods

As shown in Figure 4, the average results for the accuracy of all methods is above 94,3%. The results obtained by MLP are better over the K-means partitions, with more than

⁴<https://www.cs.waikato.ac.nz/ml/weka/>

Method	Number of Clusters	Silhouette	Davies Boudin
AH	k2	0,343	0,665
	k3	0,174	1,275
	k4	0,158	0,775
	k5	0,145	0,773
	k6	0,193	0,846
	k7	0,137	1,248
	k8	0,147	1,198
	k9	0,228	0,993
	k10	0,207	0,856
	k11	0,203	0,831
EM(average)	k2	0,468	0,379
	k3	0,349	0,581
	k4	0,300	0,654
	k5	0,229	0,893
	k6	0,164	0,721
	k7	0,216	0,661
	k8	0,192	0,736
	k9	0,178	0,703
	k10	0,155	0,681
	k11	0,140	0,653
K-means(average)	k2	0,422	0,428
	k3	0,356	0,611
	k4	0,280	0,600
	k5	0,251	0,552
	k6	0,230	0,561
	k7	0,233	0,490
	k8	0,219	0,494
	k9	0,204	0,504
	k10	0,192	0,500
	k11	0,181	0,475

Figure 3: Silhouette and Davies Boudin Index Values

99,8% of accuracy. These results are taken as the best result from classification methods and are supported by the best average for all the values (for MLP against all methods).

Method	Approach	AB	J48	KNN	MLP	NB	RF	SMO
EM	90-10	99,274	98,814	98,232	99,274	99,564	99,153	98,208
	75-25	99,359	98,896	98,327	99,181	99,751	99,074	98,327
	50-50	99,177	99,128	98,838	99,274	99,709	99,274	97,966
	66-34	99,274	98,789	98,668	99,395	100,000	99,274	98,305
	cross-validation	99,153	98,620	98,475	99,371	99,831	99,177	98,317
H A	90-10	99,032	98,547	99,419	98,087	94,770	98,935	97,361
	75-25	98,896	98,647	99,466	98,398	94,802	98,861	97,900
	50-50	99,080	98,596	99,467	98,499	94,383	98,983	97,918
	66-34	99,274	99,032	99,637	98,910	94,310	98,910	98,305
	cross-validation	99,250	98,923	99,552	98,668	94,746	98,935	97,773
K-Means	90-10	98,402	97,337	98,087	99,806	94,673	97,821	99,371
	75-25	98,434	97,508	98,149	99,786	95,087	98,754	99,502
	50-50	98,644	97,627	98,450	99,419	96,174	98,354	99,613
	66-34	98,789	97,700	99,274	100,000	95,763	98,184	99,879
	cross-validation	98,826	97,712	98,487	99,891	95,461	98,257	99,576
Mean and Standard Dev.		98,990±0,301	98,391±0,601	98,835±0,551	99,197±0,559	96,601±2,291	98,796±0,425	98,554±0,776

Figure 4: Classification values

3.5.1 Friedman Chi-square

In order to give robustness to the decision for the best partition, we used Friedman statistical test for check if there is any relevant difference (under statistical terms) between results from classification methods. The Friedman test is performed over the data presented in Figure 4 returned **p-value** value equals to 0.001259, which means that there is the significant difference between data distribution for this data. As MLP is selected as the best method for this model, the post-hoc test Nemenyi is perform over the data that returned values less than 0,05 when MLP is compare with other methods (see Figure 5) J48 and NB as the accuracy of this method is above 99%.

Method	AB	J48	KNN	MLP	NB	RF
J48	0,0524					
KNN	0,8960	0,5917				
MLP	1,0000	0,0469	0,8808			
NB	0,0419	1,0000	0,5391	0,0373		
RF	0,8280	0,6942	1,0000	0,8081	0,6437	
SMO	0,0585	1,0000	0,6179	0,0524	1,0000	0,7186

Figure 5: Safe Unsafe

3.5.2 Relabel Best Partition

With the best clustering and Classification methods and configurations selected, it is necessary to make changes in the name of cluster for a better understanding. To do this change, we used the average values of each attribute, shown in the Figure 6. It is observed that all values for the “**cluster1**” is are more than the values for the “**cluster2**” (average values) for all attributes. It means that data clustered as “**cluster1**” is more “risky” then values clustered in “**cluster2**”. For this reason, we decided to relabel the “**cluster1**” as “**unsafe**” and “**cluster2**” as “**safe**”.

3.6 Build Model According to Step 3

Using the weka data mining tool with User Application Interface, it is possible to make the entire process selected as best methods for clustering and classification. As mentioned, K-means with k=2 and seed=11 was chosen as the best clustering method for the dataset. After opening the dataset “anuj_kmeans_k2.arff” was applied the best classification method, Multi-Layer Perceptron with ten cross-fold validation configuration. Then, the file with extension “.model” was made and was used inside the application as the core of it.

3.7 Developing Cloud Application

Development of Cloud platform was made as a prototype and is made with Eclipse IDE and Spring Boot framework to provide RESTfull endpoints to use the business model proposed here. In the next section, platform details and its use are used.

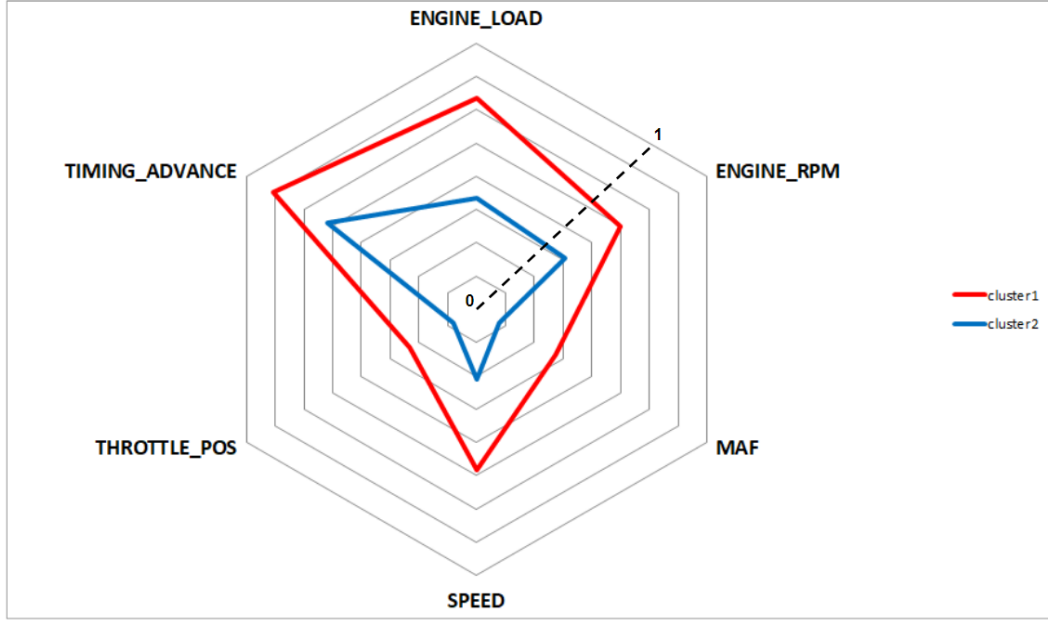


Figure 6: Friedman Chi-square

4 Design Specification

The designed platform is use as a cloud-based service for insurance companies. Multiple insurance companies can utilise the proposed model to utilise Machine learning model for the identifying of the driving behaviour (safe/unsafe) of insurance policyholder and modify their pricing approach. Beyond this, companies can use their pricing model and use the information about driver behaviour as a factor to do its price more flexible and robust.

The platform receives data from insurance companies containing telematics information of its insured vehicles and starts the internal process to predict driving behaviour using the data. The overview or architecture designed to achieve this goal is shown in Figure 7. For this work, we will name the Platform as **Driver Behavior Cloud Application (DBCA)**.

4.1 Basic Functional Architecture of proposed model

The platform works using simple steps covered by Figure 7 mentioned below:

1. Insurance Companies send data to Drive Behavior Cloud Application;
2. Driver Behavior Cloud Application (DBCA) receives the data;
3. DBCA process data;
4. DBCA classifies data according trained ML model;
5. DBCA stores the data classified;
6. DBCA sends the data to Insurance company, when it is requested.

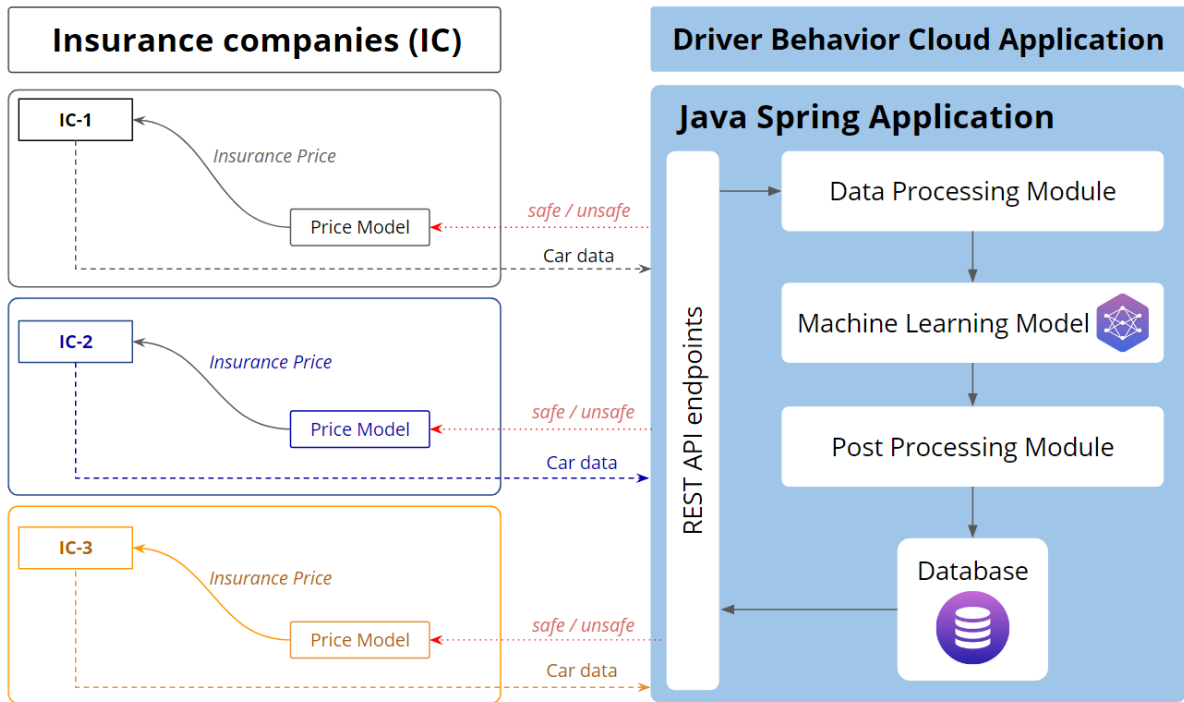


Figure 7: Driver Behavior Application General Architecture

4.1.1 DBCA receives the data

The model receives the telematics data (with vehicle telematics inside) from insurance companies through the REST API endpoint

4.1.2 DBCA process the Data

The received data is pre-processed to be analysed by the machine learning model. The data processing module will fix the missing values and normalise the data to turning into the specific pattern (ranges and value types) and allowing to use it by machine learning model.

4.1.3 DBCA classifies data according trained ML model

Data from the previous step is accepted as the input of ML model, that classifies it under “safe” or “unsafe”, and sends it to the next step.

4.1.4 DBCA stores the data classified

Data already classified is stored respecting the driver that produces it and the insurance company sends it to the DBCA service.

4.1.5 DBCA sends the data to Insurance company

When the insurance company needs some price decision or information, it sends a request over REST API to the DBCA service with driver identification and receives back all the data already classified for the driver specified. Insurance companies can use the data sent by DBCA for their pricing model which includes their company policy and possibly the

policyholder related data (Previous claims and other information) or to allow or facilitate any decision they require.

These all information is used by insurance company system for offering their customer with a beneficial policy price i.e the price can be more related to the real condition or the driving behaviour of a specific driver not with any average values for some person (average profile).

4.2 Data Pattern

Data needed by DBCA is required a specific pattern delimited for proper function of the system. Description of all the attributes is defined as described in Figure 8. This types and names are defined arbitrarily and use in the same pattern as the input of the ML model.

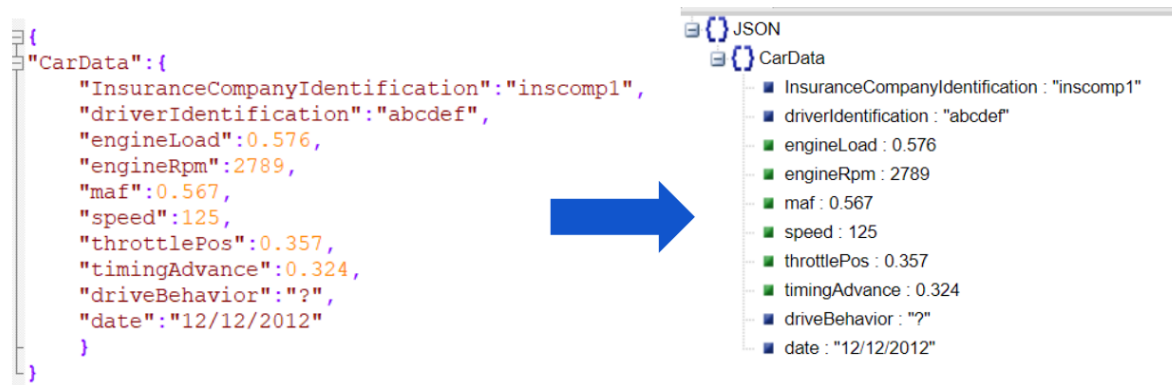


Figure 8: Data pattern defined

- **Insurance Company Identification:** *name:* "insuranceCompanyIdentification" and *type:* string
- **Driver Identification:** *name:* "driverIdentification" and *type:* string
- **Engine Load:** *name:* "engineLoad" and *type:* float point number;
- **Engine RPM:** *name:* "engineRpm" and *type:* float point number;
- **MAF:** *name:* "maf" and *type:* float point number;
- **Speed:** *name:* "speed" and *type:* integer;
- **Throttle Position:** *name:* "throttlePosition" and *type:* float point number;
- **Timing Advance:** *name:* "timingAdvance" and *type:* float point number;
- **Drive Behavior:** *name:* "driveBehavior" and *type:* string;
- **Date:** *name:* "date" and *type:* string.

4.3 Other Possible Clients

Beyond the Insurance Companies, other types of business can also have some benefits with the use of DBCA. From the best of our knowledge, it is possible that business like fleet managers and assemblers of vehicles that offer services such as scheduled revisions with pre-fixed price can use DBCA. They can offer their services to their clients with attractive/discounted prices based on their driving behaviour.

5 Implementation

The development of DBCA is made with Java Language using Eclipse IDE. The model is using a straightforward Rest API approach added by class for Machine learning model (exported from Weka -already mentioned in section 3.6). This model is loaded as a singleton at the start of the application and working on making predictions for any data while the application is running. Some details about Platform components as described in the next subsections.

5.1 Materials used

Spring Framework Johnson et al. (2004) is a Java Framework that allows Server applications to run like regular Java applications. Beyond this fact, this framework has a significant number of features (submodules) for handling essential requirements from Server Side systems. Questions like availability, security and others features are provided by Spring submodules and can be used.

Eclipse IDE, Photon-version is used as a development environment for the DBCA. Using a simple new Spring Boot starter application, it is possible to find the web module responsible for Rest API endpoints and the module H2, to store data over memory (simplest database).

5.2 Classes and Packages

Packages for the prototype platform are made under certain concepts, some of these concepts are: separate things according to the type of “job” the class do; Make sure that classes are made using best Software Engineering practices and patterns, without losing the goal of the platform; use methods and attribute names with defaults pointed by Java patterns.

Packages made for this project are:

- **dbca**: main package of application. It has only the main class of application;
- **dbca.business**: package that contains all classes for drive business rules;
- **dbca.mockdata**: package that contains classes used to create mock data inside platform;
- **dbca.repositories**: it contains the classes for manage store and recover data from database;
- **dbca.resources**: set of classes that holds and defines the REST api endpoints and also do some data manipulation;

- **dbca.tests**: classes used to test some features and changes inside application during its development;
- **dbca.utils**: classes to handle and transform data, mainly between Weka API Instances and Java object.

Figure 9 shows the organization of packages, underlining “DBCA project” (with blue line), “DBCA Main class” (with orange line), “Car Data class” (with sky blue line), “ML model class” (with purple line) and “ML .model file” (with red line).

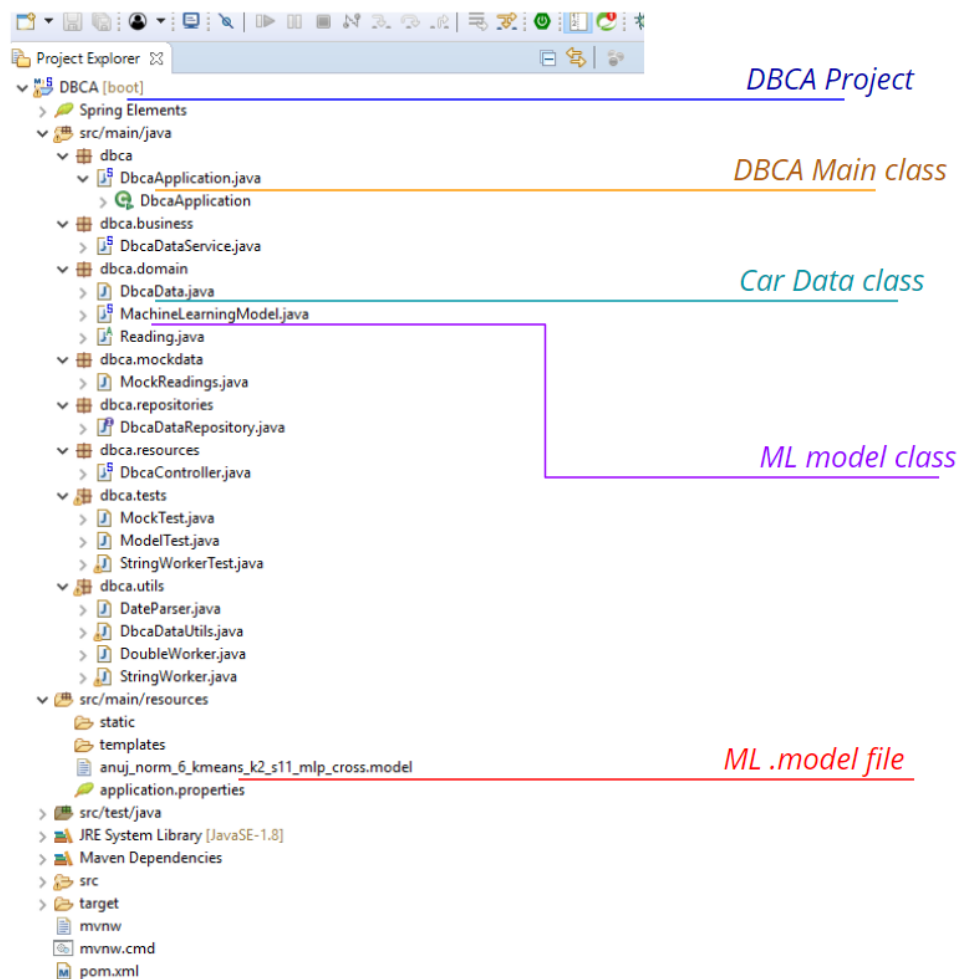


Figure 9: Packages and classes developed

5.3 Principal classes

The two principal classes of this platform are showed in Figures 10 and 11. The class named “DbcaData” defines the attributes already mentioned for Java class, allowing them the use of the object to send and receive information through the platform. The second class named “MachineLearningModel” is a class that loads and file with .model extension and keeps the object ready to classify new data. These classes are the core of the application and are responsible for define data and classify any new data from insurance companies.

The entire code of DBCA Application can be view at the link <https://github.com/anujroxx/DBCA>.

```

15
16 @Entity
17 public class DbcaData extends Reading implements Serializable {
18
19     private static final long serialVersionUID = -2977483707963112157L;
20
21     @Id
22     @GeneratedValue(strategy = GenerationType.AUTO)
23     private Integer idDbcaData;
24
25     private String insuranceCompanyIdentification;
26
27     @NotNull
28     private String driverIdentification;
29
30     private double engineLoad;
31     private double engineRpm;
32     private double maf;
33     private double speed;
34     private double throttlePos;
35     private double timingAdvance;
36     private String driverBehavior;
37
38     @Temporal(TemporalType.DATE)
39     private Date date;
40
41     public DbcaData() {
42     }
43
44     public DbcaData(String driverIdentification, String date) {
45         this.driverIdentification = driverIdentification;
46         this.driverBehavior = new String("cluster1");
47         try {
48             this.setDate(DateParser.parseString(date));
49         } catch (Exception e) {
50             this.setDate(null);
51             e.printStackTrace();
52         }
53     }
54
55     public Integer getIdDbcaData() {
56         return idDbcaData;
57     }

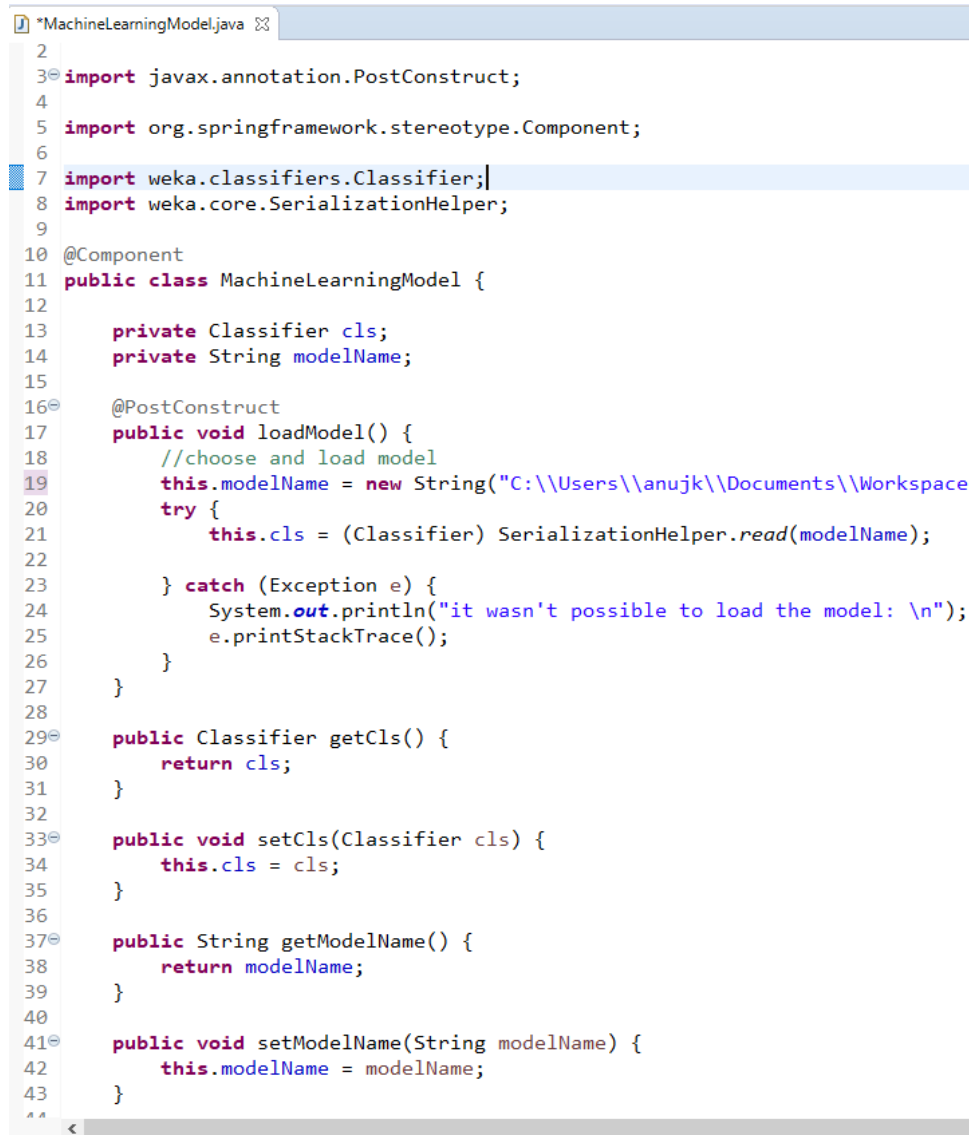
```

Figure 10: Java class that define data for application

6 Evaluation

The expected output for the proposed model is get the driving behavior of a insurance policy holder. The model selected present better results for indexes and accuracy and the main purpose of platform is possible which proves the initial hypothesis that present the possible feasibility of using of Machine learning in cloud applications for identifying driver behavior.

After some use of data mocked inside the application, one test was made to prove that any insurance company or business can use the proposed platform to make predictions of driver behaviour and apply results to its pricing model. The next section will briefly explain develop in the Prototype testing.



```

1  *MachineLearningModel.java
2
3  import javax.annotation.PostConstruct;
4
5  import org.springframework.stereotype.Component;
6
7  import weka.classifiers.Classifier;
8  import weka.core.SerializationHelper;
9
10 @Component
11 public class MachineLearningModel {
12
13     private Classifier cls;
14     private String modelName;
15
16     @PostConstruct
17     public void loadModel() {
18         //choose and load model
19         this.modelName = new String("C:\\Users\\anujk\\Documents\\Workspace\\
20         try {
21             this.cls = (Classifier) SerializationHelper.read(modelName);
22
23         } catch (Exception e) {
24             System.out.println("it wasn't possible to load the model: \n");
25             e.printStackTrace();
26         }
27     }
28
29     public Classifier getCls() {
30         return cls;
31     }
32
33     public void setCls(Classifier cls) {
34         this.cls = cls;
35     }
36
37     public String getModelName() {
38         return modelName;
39     }
40
41     public void setModelName(String modelName) {
42         this.modelName = modelName;
43     }
44 }

```

Figure 11: Java class for using ML model built in Weka

6.1 Prototype testing - Research Experiment

After development, one test performed with the client-side application (Postman API Development Environment). In this experiment, data was sent by the postman and classified by DBCA application on the server side. The Figure 12 shows the JSON data selected to be sent to DBCA application, classified by it and sent back to Postman. Note that the attribute “driver behaviour” was sent with value “?” and returns with class predicted by ML model inside DBCA application, that was “unsafe”.

6.2 Discussion

Results present by Subsection 6.1 shows that it is possible to run and deploy a Machine Learning model over the cloud to classify driver behaviour using data from cars. It is possible to see the business model and the pricing of insurance companies more just. Alternatively, people that drives car with safe behaviour can have insurance service with less expensive than people that drive using car under the unsafe behaviour.

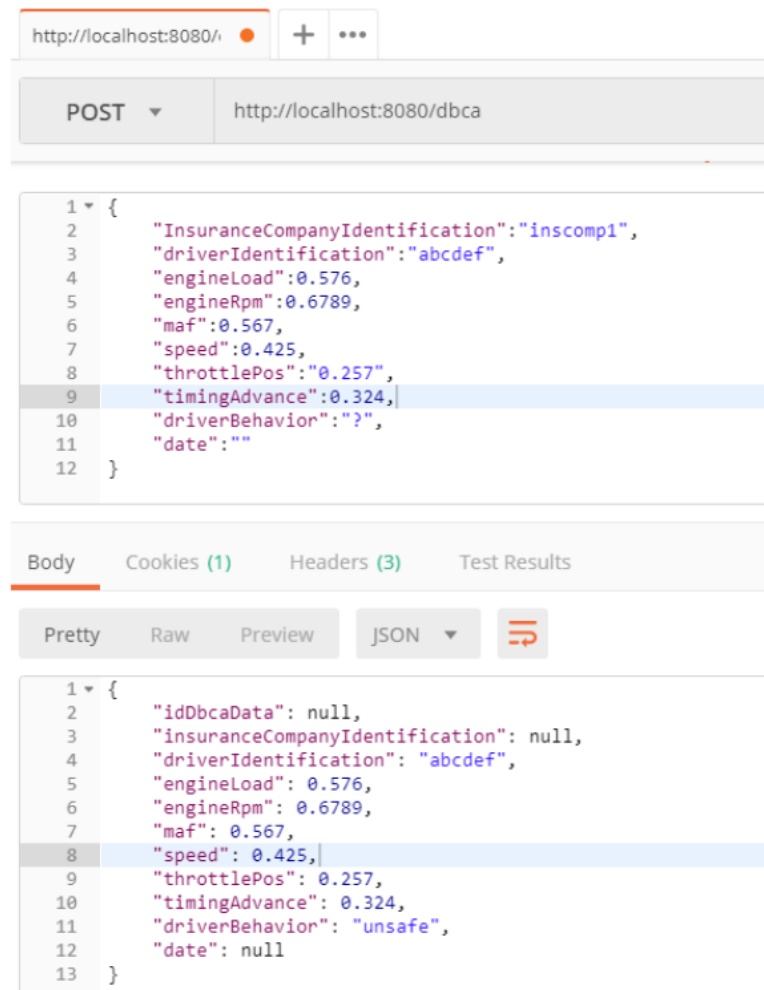


Figure 12: Data sent and received with postman

Another possible benefit coming from this model of business is that, due to its business model, possibly used by various insurance companies and business, government entities can use the data for allowing people with safe behaviour get discount when they need to buy fuel or thins related to vehicles. It is true, people with less risky behaviour uses less fuel and then produces less pollution wish can means economy in areas such as health, infrastructure and recovery of the natural environment.

7 Conclusion and Future Work

The main focus of this research is to utilise the cloud technology integrated with machine learning model to analyse driving behavior where the cloud technology is used as vehicular telematics platform to analyse and process the massive amount data generated by vehicle sensors and predict the driving behavior of driver.

This work is supported by insurance companies using User-Based-Insurance or Pay-How-You-Drive pricing model and other business related to driver behaviour. The proposed prediction model was made by using clustering and classifying methods and achieves good validation index values (Silhouette and DB indexes) and also good value for accuracy at the classifying step. Using this model, we are sure that many insurance companies

and other business can get a more personalized pricing model and with it, get benefits also for its clients.

Some limitations of this work are that the application depends of receiving data from platform evolving hardware(car, ECUS, sensors, OBD and others) and software inside some smartphone (the most comfortable way to get data from the car). The other thing is that a right approach is needed to re-train the ML method according to new data from new drivers (making it more robust and wise) but it can be a problem according to insurance companies policies.

At final, the overall result is robust enough with some variations of this work can be done while a business model of this research is under evaluation. Mentioned variation is the use of OBD device with the Internet connection and an approach using streaming analysis for data coming from cars.

References

- Amarasinghe, M., Kottegoda, S., Arachchi, A. L., Muramudalige, S., Bandara, H. D. and Azeez, A. (2015). Cloud-based driver monitoring and vehicle diagnostic with obd2 telematics, *Electro/Information Technology (EIT), 2015 IEEE International Conference on*, IEEE, pp. 505–510.
- Amsalu, S. B., Homaifar, A., Afghah, F., Ramyar, S. and Kurt, A. (2015). Driver behavior modeling near intersections using support vector machines based on statistical feature extraction, *Intelligent Vehicles Symposium (IV), 2015 IEEE*, IEEE, pp. 1270–1275.
- Araújo, R., Igreja, Â., de Castro, R. and Araujo, R. E. (2012). Driving coach: A smart-phone application to evaluate driving efficient patterns, *Intelligent Vehicles Symposium (IV), 2012 IEEE*, IEEE, pp. 1005–1010.
- Baecke, P. and Bocca, L. (2017). The value of vehicle telematics data in insurance risk selection processes, *Decision Support Systems* **98**: 69–79.
- Barreto, C. A. d. S. (2018). *Uso de técnicas de aprendizado de máquina para identificação de perfis de uso de automóveis baseado em dados automotivos*, Master’s thesis, Brasil.
- Barreto, C., Xavier-Jnior, J. C., Canuto, A. M. P. and da Silva, I. M. D. (2018). A machine learning approach based on automotive engine data clustering for driver usage profiling classification, *Anais do Encontro Nacional de Inteligncia Artificial e Computacional (ENIAC)* pp. 174–185.
URL: <http://portaldeconteudo.sbc.org.br/index.php/eniac/article/view/4414>
- Bian, Y., Yang, C., Zhao, J. L. and Liang, L. (2018). Good drivers pay less: A study of usage-based vehicle insurance models, *Transportation research part A: policy and practice* **107**: 20–34.
- Boquete, L., Rodríguez-Ascariz, J. M., Barea, R., Cantos, J., Miguel-Jiménez, J. M. and Ortega, S. (2010). Data acquisition, analysis and transmission platform for a pay-as-you-drive system, *Sensors* **10**(6): 5395–5408.
- Butler, P., Butler, T. and Williams, L. L. (1988). Sex-divided mileage, accident, and insurance cost data show that auto insurers overcharge most women.

- Castignani, G., Derrmann, T., Frank, R. and Engel, T. (2015). Driver behavior profiling using smartphones: A low-cost platform for driver monitoring, *IEEE Intelligent Transportation Systems Magazine* **7**(1): 91–102.
- Eren, H., Makinist, S., Akin, E. and Yilmaz, A. (2012). Estimating driving behavior by a smartphone, *Intelligent Vehicles Symposium (IV), 2012 IEEE*, IEEE, pp. 234–239.
- Hastie, T., Tibshirani, R. and Friedman, J. (2009). Unsupervised learning, *The elements of statistical learning*, Springer, pp. 485–585.
- Iqbal, M. U. and Lim, S. (2006). A privacy preserving gps-based pay-as-you-drive insurance scheme, *Symposium on GPS/GNSS (IGNSS2006)*, pp. 17–21.
- Jhou, J.-S., Chen, S.-H., Tsay, W.-D. and Lai, M.-C. (2013). The implementation of obd-ii vehicle diagnosis system integrated with cloud computation technology, *Robot, Vision and Signal Processing (RVSP), 2013 Second International Conference on*, IEEE, pp. 9–12.
- Johnson, R., Hoeller, J., Donald, K., Sampaleanu, C., Harrop, R., Risberg, T., Arendsen, A., Davison, D., Kopylenko, D., Pollack, M. et al. (2004). The spring framework—reference documentation, *Interface* **21**: 27.
- Kumari, M. and Godara, S. (2011). Comparative study of data mining classification methods in cardiovascular disease prediction 1.
- Meseguer, J. E., Calafate, C. T., Cano, J. C. and Manzoni, P. (2013). Drivingstyles: A smartphone application to assess driver behavior, *Computers and Communications (ISCC), 2013 IEEE Symposium on*, IEEE, pp. 000535–000540.
- Michalski, R. S., Carbonell, J. G. and Mitchell, T. M. (2013). *Machine learning: An artificial intelligence approach*, Springer Science & Business Media.
- Nai, W., Chen, Y., Yu, Y., Zhang, F., Dong, D. and Zheng, W. (2016). Fuzzy risk mode and effect analysis based on raw driving data for pay-how-you-drive vehicle insurance, *Big Data Analysis (ICBDA), 2016 IEEE International Conference on*, IEEE, pp. 1–5.
- Paefgen, J., Kehr, F., Zhai, Y. and Michahelles, F. (2012). Driving behavior analysis with smartphones: insights from a controlled field study, *Proceedings of the 11th International Conference on mobile and ubiquitous multimedia*, ACM, p. 36.
- Parry, I. W. (2004). Comparing alternative policies to reduce traffic accidents, *Journal of Urban Economics* **56**(2): 346–368.
- Rätsch, G., Onoda, T. and Müller, K.-R. (2001). Soft margins for adaboost, *Machine learning* **42**(3): 287–320.
- Simmhan, Y., Aman, S., Kumbhare, A., Liu, R., Stevens, S., Zhou, Q. and Prasanna, V. (2013). Cloud-based software platform for big data analytics in smart grids, *Computing in Science & Engineering* **15**(4): 38–47.
- Takefuji, Y. (2018). Connected vehicle security vulnerabilities [commentary], *IEEE Technology and Society Magazine* **37**(1): 15–18.

- Troncoso, C., Danezis, G., Kosta, E., Balasch, J. and Preneel, B. (2011). Pripayd: Privacy-friendly pay-as-you-drive insurance, *IEEE Transactions on Dependable and Secure Computing* **8**(5): 742–755.
- Tselentis, D. I., Yannis, G. and Vlahogianni, E. I. (2016). Innovative insurance schemes: pay as/how you drive, *Transportation Research Procedia* **14**: 362–371.
- Van Ly, M., Martin, S. and Trivedi, M. M. (2013). Driver classification and driving style recognition using inertial sensors, *Intelligent Vehicles Symposium (IV), 2013 IEEE*, IEEE, pp. 1040–1045.
- Wiwie, C., Baumbach, J. and Röttger, R. (2015). Comparing the performance of biomedical clustering methods, *Nature methods* **12**(11): 1033.
- Zhang, C., Patel, M., Buthpitiya, S., Lyons, K., Harrison, B. and Abowd, G. D. (2016). Driver classification based on driving behaviors, *Proceedings of the 21st International Conference on Intelligent User Interfaces*, ACM, pp. 80–84.

Configuration Manual

MSc Research Project
Cloud Computing

Anuj Kumar
Student ID: X17157641

School of Computing
National College of Ireland

Supervisor: Victor Del Rosal

National College of Ireland
Project Submission Sheet
School of Computing



Student Name:	Anuj Kumar
Student ID:	X17157641
Programme:	Cloud Computing
Year:	2018
Module:	MSc Research Project
Supervisor:	Victor Del Rosal
Submission Due Date:	20/12/2018
Project Title:	Configuration Manual
Word Count:	426
Page Count:	4

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	19th December 2018

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Anuj Kumar
X17157641

1 Pre-requisite

1. Java is required to run application.
2. Java Eclipse IDE (Integrated development Environment) to run the Java application Murphy et al. (2006).
3. Postman (API development environment) for testing the data flow Schreier and Hayn (2018).

2 Run Application

1. Need to Download/clone in eclipse workspace from the git hub link <https://github.com/anujroxx/DBCA>.
2. After installing Eclipse, use it to create IDE for Java application and import DBCA project in Eclipse IDE. we need to browse the root directory to select the location of DBCA.
3. After importing DBCA, Run DbcaApplication.java present in dbca package under the path ""src/main/java"" as shown in Figure 2 to initiate the proposed model.
4. After running the application, Sprint boot is initiated and that can seen in figureFigure 3the application is running on localhost:8080 port.
5. Internal processing of the application is already explained in main report (Sec:5)how all classed are packages are used to process the data.

3 DBCA is used to Analyze the input data

1. Open the Postman API development environment to send a post request with input data on <http://localhost:8080/dbca>. Input data should be in Json format to be analyzed by machine learning model as shown in figure Figure 4.
2. It is observable in figure Figure 5 for the input values of "drivingBehaviour" in Figure 4 is marked as "?" and the return value for the the query driving behaviour is "unsafe".
3. Similarly in figure it is observable, for different input values of telematics data DBCA is proving results based on analysis as safe and unsafe.

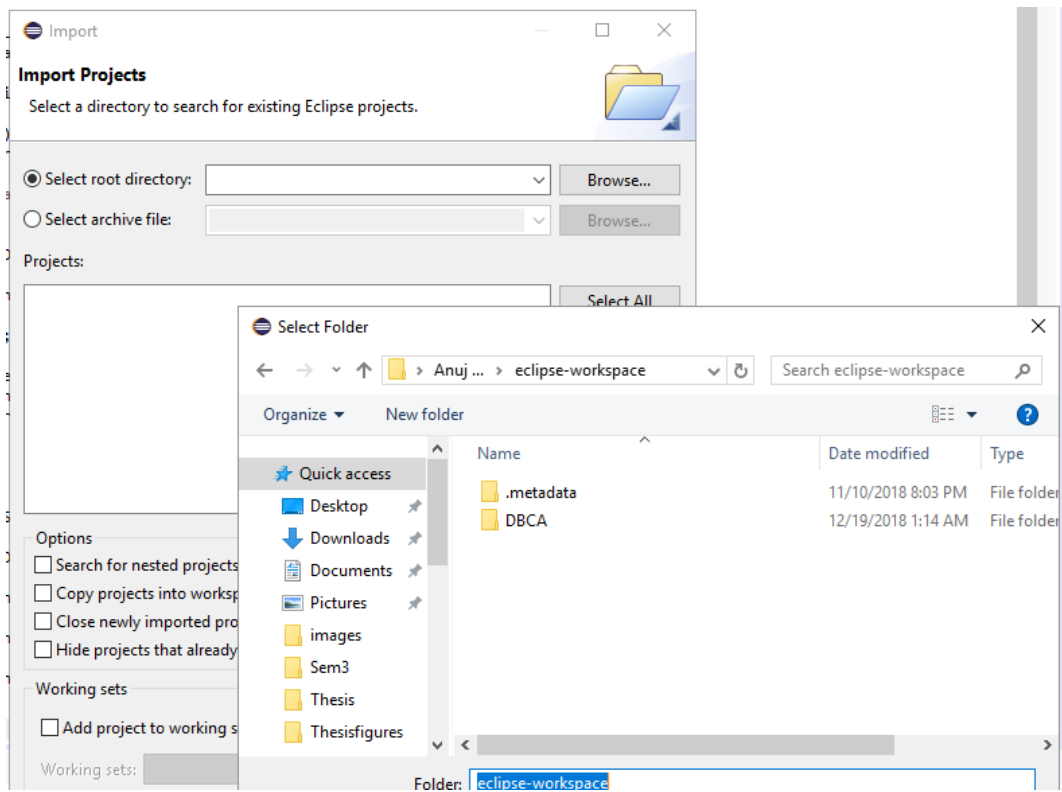


Figure 1: Import DBCA in Eclipse IDE

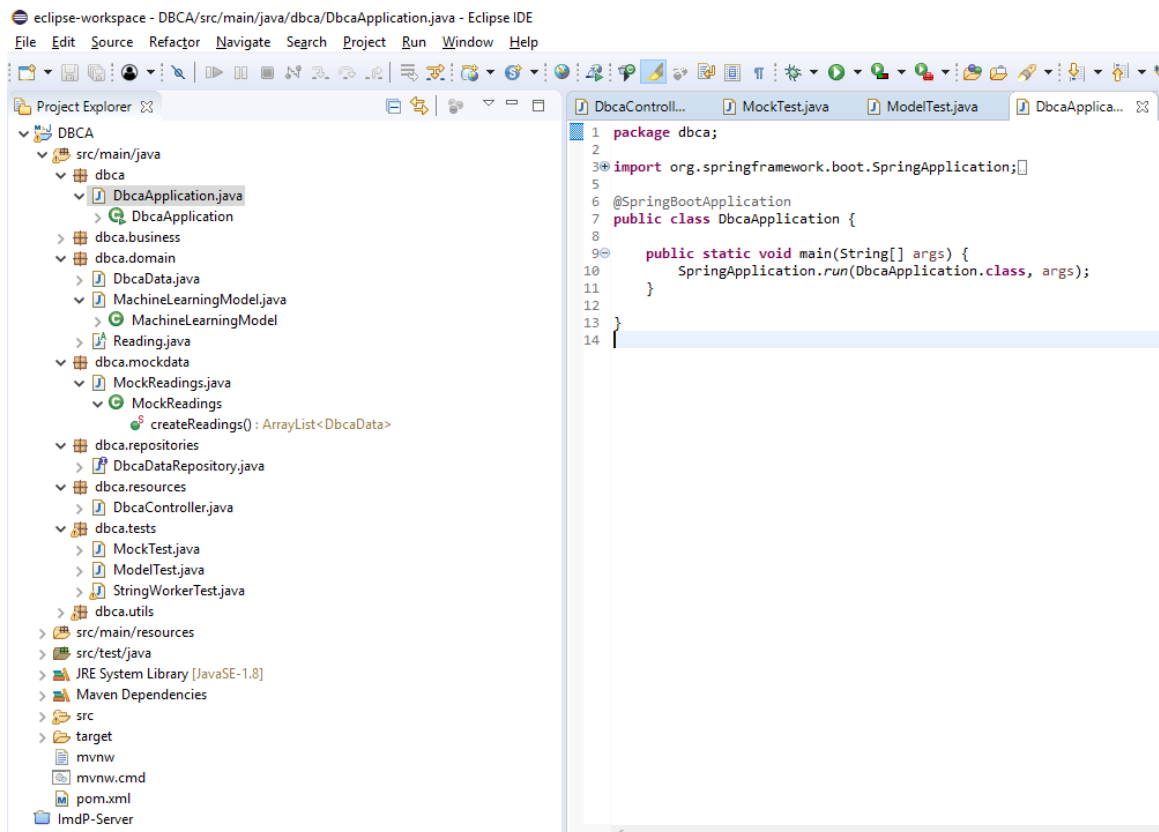


Figure 2: Run DbcaApplication.java

```

2018-12-19 16:24:50.102 INFO 1508 --- [main] org.hibernate.Version : HHH000412: Hibernate Core (5.2.17.Final)
2018-12-19 16:24:50.103 INFO 1508 --- [main] org.hibernate.cfg.Environment : HHH000206: hibernate.properties not found
2018-12-19 16:24:50.156 INFO 1508 --- [main] o.hibernate.annotations.common.Version : HCANNN000001: Hibernate Commons Annotations (5.0.1.Final)
2018-12-19 16:24:50.345 INFO 1508 --- [main] org.hibernate.dialect.Dialect : HHH000408: Using dialect: org.hibernate.dialect.H2Dialect
2018-12-19 16:24:50.806 INFO 1508 --- [main] o.h.t.schema.internal.SchemaCreatorImpl : HHH000476: Executing import script 'org.hibernate.tool.schema.internal.exec.ScriptSourceInputNonExiste
2018-12-19 16:24:50.889 INFO 1508 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2018-12-19 16:24:51.373 INFO 1508 --- [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**/favicon.ico] onto handler of type [class org.springframework.web.servlet.resource
2018-12-19 16:24:51.655 INFO 1508 --- [main] s.w.s.m.m.a.RequestMappingHandlerAdapter : Looking for @ControllerAdvice: org.springframework.boot.web.servlet.context.AnnotationConfigServletWet
2018-12-19 16:24:51.698 WARN 1508 --- [main] aWebConfiguration$JpaWebMvcConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during vie
2018-12-19 16:24:51.733 INFO 1508 --- [main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/dbca],methods=[POST],consumes=[application/json]}" onto public dbca.domain.DbcaData dbca.re
2018-12-19 16:24:51.735 INFO 1508 --- [main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error]}" onto public org.springframework.http.ResponseEntity<java.util.Map<java.lang.String
2018-12-19 16:24:51.736 INFO 1508 --- [main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error],produces=[text/html]}" onto public org.springframework.web.servlet.ModelAndView org.
2018-12-19 16:24:51.768 INFO 1508 --- [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/webjars/**] onto handler of type [class org.springframework.web.servlet.resource.Res
2018-12-19 16:24:51.768 INFO 1508 --- [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**] onto handler of type [class org.springframework.web.servlet.resource.ResourceHtt
2018-12-19 16:24:52.203 INFO 1508 --- [main] o.s.j.e.a.AnnotationMBeanExporter : Registering beans for JMX exposure on startup
2018-12-19 16:24:52.206 INFO 1508 --- [main] o.s.j.e.a.AnnotationMBeanExporter : Bean with name 'dataSource' has been autodetected for JMX exposure
2018-12-19 16:24:52.211 INFO 1508 --- [main] o.s.j.e.a.AnnotationMBeanExporter : Located MBean 'dataSource': registering with JMX server as MBean [com.zaxxer.hikari:name=dataSource,t
2018-12-19 16:24:52.279 INFO 1508 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2018-12-19 16:24:52.284 INFO 1508 --- [main] dbca.DbcaApplication : Started DbcaApplication in 6.191 seconds (JVM running for 6.76)
2018-12-19 16:27:42.790 INFO 1508 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring FrameworkServlet 'dispatcherServlet'
2018-12-19 16:27:42.791 INFO 1508 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initialization started

```

Figure 3: Application listening on port localhost:8080

The screenshot shows the Postman interface for a POST request to `http://localhost:8080/dbca`. The request body is a JSON object with the following structure:

```

1 {
2   "insuranceCompanyIdentification": "inscomp1",
3   "driverIdentification": "abcdef",
4   "engineLoad": 0.424,
5   "engineRpm": 0.410,
6   "maf": 0.425,
7   "speed": 0.300,
8   "throttlePos": "0.257",
9   "timingAdvance": 0.324,
10  "driverBehavior": "?",
11  "data": ""
12 }

```

Figure 4: Value sent using Postman

The screenshot shows the response body in Postman, which is a JSON object with the following structure:

```

1 {
2   "idDbcaData": null,
3   "insuranceCompanyIdentification": "inscomp1",
4   "driverIdentification": "abcdef",
5   "engineLoad": 0.424,
6   "engineRpm": 0.41,
7   "maf": 0.425,
8   "speed": 0.3,
9   "throttlePos": 0.257,
10  "timingAdvance": 0.324,
11  "driverBehavior": "safe",
12  "date": null
13 }

```

Figure 5: Return value for the Postman input

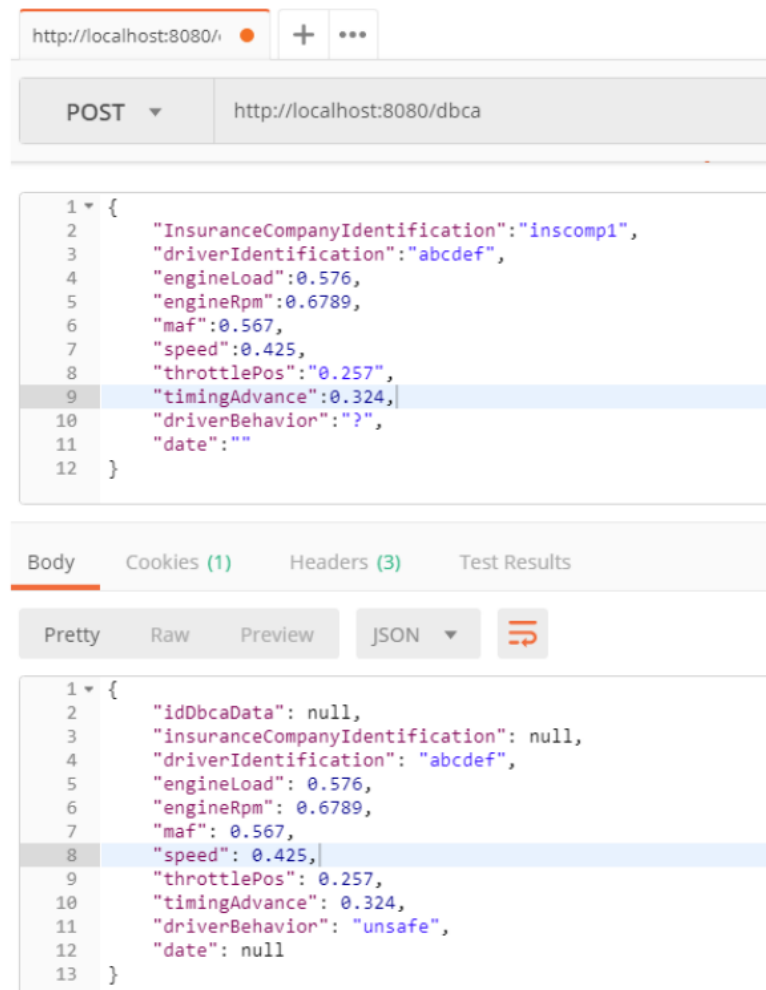


Figure 6: Return value for the Postman input

References

- Murphy, G. C., Kersten, M. and Findlater, L. (2006). How are java software developers using the eclipse ide?, *IEEE software* **23**(4): 76–83.
- Schreier, G. and Hayn, D. (2018). Achieving interoperability between arden-syntax-based clinical decision support and openehr-based data systems, *Health Informatics Meets EHealth: Biomedical Meets EHealth—From Sensors to Decisions. Proceedings of the 12th EHealth Conference*, Vol. 248, IOS Press, p. 338.