Machine Learning Models applied to Car Industry

Juan Pablo Palma B.
MSc in Data Analytics
National College of Ireland
Dublin, Ireland
x20225776@student.ncirl.ie

***Abstract*—This report aims to apply five Machine Learning algorithms to accomplish prediction and classification tasks using the KDD methodology. The three data set used Car price prediction, Car Insurance Claim Classification, and Car accident severity Classification. Separate data exploration, clean and transform each dataset before applying various machine learning models to them. Evaluating models were based on multiple parameters. For the regression task, linear regression model and regression tree, achieving an R^2 of 0.90, and for Binary Classification (Car Insurance Claim) K-nearest neighbour (KNN) classifier, Random Forest (RF) achieving an accuracy of 0.82 and 0.84, respectively, and Naive Bayes classifier (Car accident Severity) with an accuracy of 0.8. The conclusions were then drawn and reported comprehensively.**

***Index Terms*—machine learning, car insurance claim, car price, car accident severity, linear regression, regresion tree, KNN, random forest, naive bayes.**

I. Introduction

This paper will review three of the classic appliances of Machine Learning (ML) algorithms in real-world problems. The primary motivation was to carefully find a group of data sets that allow us to apply different uses of ML in other contexts such as numerical prediction, binary classification, and multi-class classification. Also, we looked for three data sets related to the same industry. And, finally, apply new knowledge acquired in the course in a wide range of real-world problems for data analysis and where ML can be a helpful tool to solve it.

A. Machine Learning

Machine Learning can be defined as a subset of the Artificial Intelligence that use the computer resources for learn about examples, data and experience and use that knowledge in future appliances and in a scalable dimension [1].

Machine Learning can use Supervised Learning (when we know the target variable) or Unsupervised Learning (when we are unsure of the target variable). This paper will focus on Supervised Learning in the two main branch problems: Regression and Classification models [2] [3].

B. Car industry

The car industry is often used in many ML courses. However, it is usually for appliances of one of the possible target predictions. In this paper the goal is cover different target predictions applied to the same industry. Therefore, we uses the Car industry to apply Regression and Classification problems, such as:

- **Regression:** Car price predictions using Linear regression models and Regression Tree model. The objective is predict a numerical value using ML algorithm.
- **Binary Classification:** Car insurance Claim using KNN and Random Forest Models. The objective is predict a categorical variable (factor of two levels) using ML algorithm.
- **Multi-class Classification:** Car accident severity using Naive Bayes Models. The objective is predict a categorical variable (factor of more than two levels) using ML algorithm.

Finally, the research question along all the documents is which ML algorithm had a better performance for the different questions that each data set had. Thus, this paper is an academic effort to apply other ML methods in three different real-world challenges where ML can be a helpful answer.

The paper is structured as follows, presenting the Related Work for each Data set after explaining the Methodology used in this project. Next, Exploratory and Data cleaning is the most time-consuming process after the Result of the appliances of the five ML models in the different data sets. And Finally, an evaluation of these models' performance and the main conclusion and future work challenges.

We wrote this paper to present every step across the three problems that each data set presents in a comparative perspective, meeting the main objective of applying different ML uses in other contexts.

II. Related Work

A. Car price prediction - Regression

- Anamika Das Mou and others [4]. They developed a model for orienting customers' purchases, using different

models, such as Support Vector Machine (SVM), Naive Bayes, Random Forest tree, KNN and others. The result of the paper shows how SVM obtain the best result.

- Nitis Monburinon and others [5]. Perform a comparative between gradient boosted regression tree, random forest and multiple linear regression. In the paper, the worst performance came from the linear regression, which could be a future contribution and challenge the linear regression model built in this document.

- Enis Gegic and others [6]. They applied Artificial Neural Network, Support Vector Machine and Random Forest as an ensemble, which achieves better performance but uses more computational resources. And also can be considered as a future perspective for our work in this paper.

B. Car insurance claim - Classification

- Mohamed Hanafy and Ruixing Ming [7] have worked on the importance of using ML in increasing trends in the "number" and "severity" of the car insurance claim. They used models such as logistic regression, XGBoost, random forest, decision trees, Naive Bayes and K-NN. The central insight is that Random Forest performs consistently better.

- Hui Dong Wang [8] tested models such as Random Forest, Gradient lifting tree (GBDT) and Lifting machine algorithm (LightGBM) as part of the increasing demand for this kind of analysis. The central feature of this paper is the influence of some variables, such as new car purchase price, car age, or car insurance business channel. These results are related to the analysis presented in this paper because of the leading role of the year of car release (car age), which significantly increased the Regression Model's performance.

- Ranjodh Singh and others [9]. They developed an automatic process able to estimate the repair cost using deep learning. Even though the paper is a high level of machine learning is an important inspiration source for motivating future work in this topic and opening the range of possible applications of ML models.

C. Car accident severity - Multi-class Classification

- A. Atwah and A. Al-Mousa [10]. Develop a machine learning model that achieves an accuracy of 83.9% with the Random Forest. Also, the focus was related to future uses of medical resources needed for the attention in this kind of accident.

- B. Geyik and M. Kara [11]. Perform four ML Classification models, and it is noticed the excellent performance of the Naive Bayes in this paper with an accuracy of 83.40%. Still, in our model it accuracy is considered not enough because of the original distribution of our target variable.

- R. Al Mamlook and others [12]. The main insight in this paper was the role of three variables: traffic volume, driver's age and car's age. Unfortunately, in our data set, these variables were not present, and it can be considered another good improvement in future works.

III. Methodology

A. KDD - Knowledge Discovery from Data

In this project we will use the KDD (Knowledge Discovery in Databases) methodology for building the machine learning models [13]. That involves the following steps:

- **Selection of the target data set:** in this step we focused to find a data set with the required properties such as at least 10,000 rows and 10 columns, ideally related.

- **Data Understanding:** at this step we focused to find and define the first set of possible variables to be included in the final data set.

- **Data preparation and Exploratory Analysis:** in this step we checked the quality of the data, checking quantity of missing values or spelling mistakes and fixing when possible. Also we prepared preliminary visualizations of univariate and multivariate analysis as part of the Exploratory analysis.

- **Defining the feature matrix:** at this step we were able to define all the pull of independent variable that will be tested on the ML model process. Also, at this step we made the finally changes such as normalize the data when required or fixing the levels of the factor variables for a better interpretation.

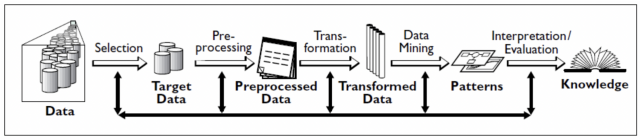
- **Modelling:** at this step we applied the ML models using the feature matrix and the target variable, also in this step we split the data in training data (usually 80% of the data) and testing data (usually the 20% of the data). Often this process is iterative checking the specification of the models and the assumptions at the background.

- **Evaluation:** at this step we compared and checked the finals adjustments that can be possible to apply in models, but mainly this step is for checking the performance of the models.

- **Presentation:** at this final step we select and improve the past visualization and decide the better way for presenting the new knowledge in a clear and understandable story.

An structured summary of all above steps is depicted in the figure 1:

Fig. 1: Knowledge DIscovery from Databases (KDD)



IV. Exploratory Analysis and Data cleaning

A. Car price prediction

In a general viewing, the dataset has 16 variables and 11,915 observations.

Fig. 2: Data understanding - General Viewing

```
'data.frame':   11914 obs. of  16 variables:
 $ make             : Factor w/ 48 levels "Acura","Alfa Romeo",..: 6 6 6 6 6 6 6 6 6 6 ...
 $ model            : Factor w/ 915 levels "1 Series","1 Series M",..: 2 1 1 1 1 1 1 1 1 1 ...
 $ year             : int  2011 2011 2011 2011 2011 2012 2012 2012 2012 2013 ...
 $ engine_fuel_type : Factor w/ 11 levels "","diesel","electric",..: 10 10 10 10 10 10 10 10 10 10 ...
 $ engine_hp        : int  335 300 300 230 230 230 300 300 230 230 ...
 $ engine_cylinders : int  6 6 6 6 6 6 6 6 6 6 ...
 $ transmission_type: Factor w/ 5 levels "AUTOMATED_MANUAL",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ driven_wheels    : Factor w/ 4 levels "all wheel drive",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ number_of_doors  : int  2 2 2 2 2 2 2 2 2 2 ...
 $ market_category  : Factor w/ 72 levels "Crossover","Crossover,Diesel",..: 39 68 65 68 64 68 68 65 64 6
4 ...
 $ vehicle_size     : Factor w/ 3 levels "Compact","Large",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ vehicle_style    : Factor w/ 16 levels "2dr Hatchback",..: 9 7 9 9 7 9 7 9 7 7 ...
 $ highway_mpg      : int  26 28 28 28 28 28 26 28 28 27 ...
 $ city_mpg         : int  19 19 20 18 18 18 17 20 18 18 ...
 $ popularity       : int  3916 3916 3916 3916 3916 3916 3916 3916 3916 3916 ...
 $ msrp             : int  46135 40650 36350 29450 34500 31200 44100 39300 36900 37200 ...
```

The cars are the ones between the years 1990 and 2017, and the average price of the sample is $40,595.

TABLE I: Statistical Descriptive

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
year	11,914	2,010	7.58	1,990	2,007	2,016	2,017
engine_hp	11,914	249.19	109.21	13	170	300	1,001
engine_cylinders	11,914	5.62	1.79	0	4	6	16
number_of_doors	11,914	3.44	0.88	2	2	4	4
highway_mpg	11,914	26.64	8.86	12	22	30	354
city_mpg	11,914	19.73	8.99	7	16	22	137
popularity	11,914	1,554.91	1,441.86	2	549	2,009	5,657
msrp	11,914	40,594.74	60,109.10	2,000	21,000	42,231.2	2,065,902

Focusing on the dependent variable, which is ”MSRP” (manufacturer’s suggested retail price), is clear the positive skewness (see figure 3).

Also, we tried to fix it with a logarithmic transformation (see Figure 4, and even when it does not fix it to a completely normal distribution, it is good enough for work with a Linear Regression model.

1) Data understanding: One of the most relevant variables is the brand of the car (variable ”make” in the data set). For instance, of course, a Lamborghini as an exclusive brand will be more expensive than a regular city car. We have 48 different brands in the data set. Therefore, It is possible to see the average price by each of the brands in the Table II. The central insight is that we have a top 11 brand with an average of over $100,000 dollars.

2) Data cleaning: The data cleaning process was challenging because it required considerable research effort. For example, the data set had 69 Non-value cases in the variable ”engine_hp” (engine horsepower), engine_cylinder

Fig. 3: Histogram of Car Price

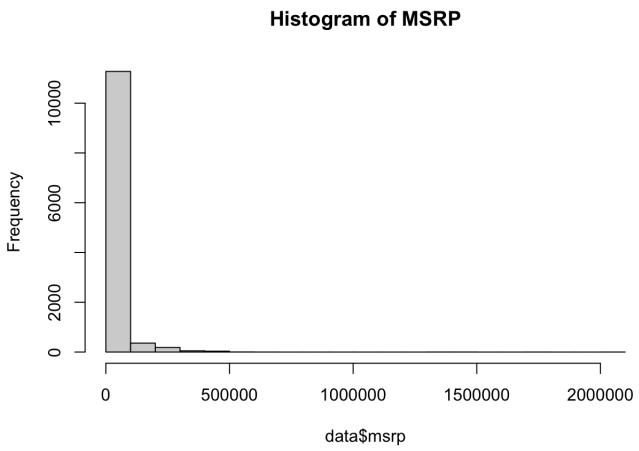
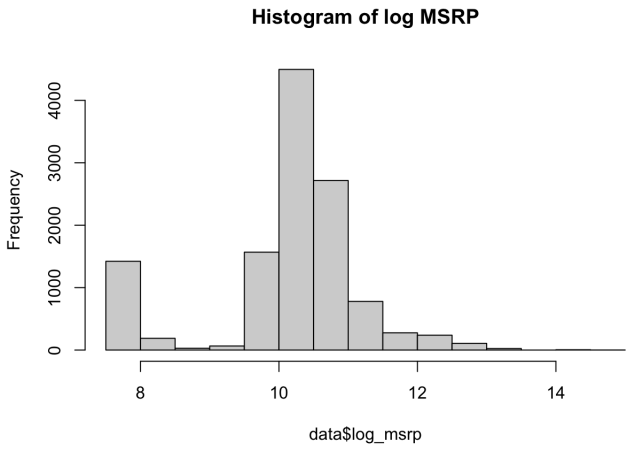


Fig. 4: Histogram of log Car price



with 30 observations and ”number of doors” with six observations.

We input the values of ”engine_hp” searching on the internet for the specific brand, model, engine, engine cylinders, and all the valuable characteristics to give us a good proxy to the actual value. The first step for the other two variables was to discover which type of car has the missing values and input them as a real value.

As a result of this process, we could recover 100% of the original data set, which means we did not lose any cases in the cleaning data process.

3) Feature Matrix: After the cleaning process, we selected a set of 12 variables potentially used in the modelling process and a data frame of 11,914 observations (see Figure 5).

TABLE II: Average Car price by Brand

N	Make	Average	N	Make	Average
1	Bugatti	1,757,224	25	Acura	34,888
2	Maybach	546,222	26	GMC	30,493
3	Rolls-Royce	351,131	27	Toyota	29,030
4	Lamborghini	331,567	28	Nissan	28,583
5	Bentley	247,169	29	Volvo	28,541
6	McLaren	239,805	30	Chevrolet	28,350
7	Ferrari	238,219	31	Buick	28,207
8	Spyker	213,323	32	Volkswagen	28,102
9	Aston Martin	197,910	33	Saab	27,414
10	Maserati	114,208	34	Ford	27,399
11	Porsche	101,622	35	Chrysler	26,723
12	Tesla	85,256	36	Honda	26,674
13	Mercedes-Benz	71,476	37	Kia	25,310
14	Lotus	69,188	38	Subaru	24,828
15	Land Rover	67,823	39	Hyundai	24,597
16	Alfa Romeo	61,600	40	FIAT	22,670
17	BMW	61,546	41	Dodge	22,390
18	Cadillac	56,231	42	Mitsubishi	21,241
19	Audi	53,452	43	Mazda	20,039
20	Lexus	47,549	44	Scion	19,932
21	Genesis	46,616	45	Pontiac	19,322
22	Lincoln	42,839	46	Suzuki	17,907
23	Infiniti	42,394	47	Oldsmobile	11,543
24	HUMMER	36,464	48	Plymouth	3,123

TABLE III: Variables with NA values

	x
engine_hp	69
engine_cylinders	30
number_of_doors	6

The decision to select this subset of variables was considering the criteria of remaining variables giving information, but not raising excessively the number of factors in the model and correcting the class of every variable to respect the original type of the data.

Fig. 5: Feature Matrix Car price

```
'data.frame':   11914 obs. of  12 variables:
 $ make             : Factor w/ 48 levels "Acura","Alfa Romeo",..: 6 6 6 6 6 6 6 6 6 6 ...
 $ year             : int  2011 2011 2011 2011 2011 2012 2012 2012 2012 2013 ...
 $ engine_fuel_type : Factor w/ 11 levels "","diesel","electric",..: 10 10 10 10 10 10 10 10 10 1
 $ engine_hp        : num  335 300 300 230 230 230 300 300 230 230 ...
 $ engine_cylinders : num  6 6 6 6 6 6 6 6 6 6 ...
 $ transmission_type: Factor w/ 4 levels "MANUAL","AUTOMATED_MANUAL",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ driven_wheels    : Factor w/ 3 levels "front wheel drive",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ number_of_doors  : Factor w/ 3 levels "Two","Three",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ vehicle_size     : Factor w/ 3 levels "Compact","Midsize",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ highway_mpg      : int  26 28 28 28 28 28 26 28 28 27 ...
 $ city_mpg         : int  19 19 20 18 18 18 17 20 18 18 ...
 $ log_msrp         : num  10.7 10.6 10.5 10.3 10.4 ...
```

It is also possible to see in Figure 6, the partial correlations between numerical values in the data set. It is particularly relevant because this information will help decide which variables are generation multicollinearity problems on the Multiple Linear Regression model, such as the relation of 0.77 between ”engine_hp” and ”engine_cylinders”.

B. Car insurance claim

1) Data understanding: The car insurance claim data set has 19 variables and 10,000 observations from a general perspective (see Figure 7).

Fig. 6: Partial Correlation

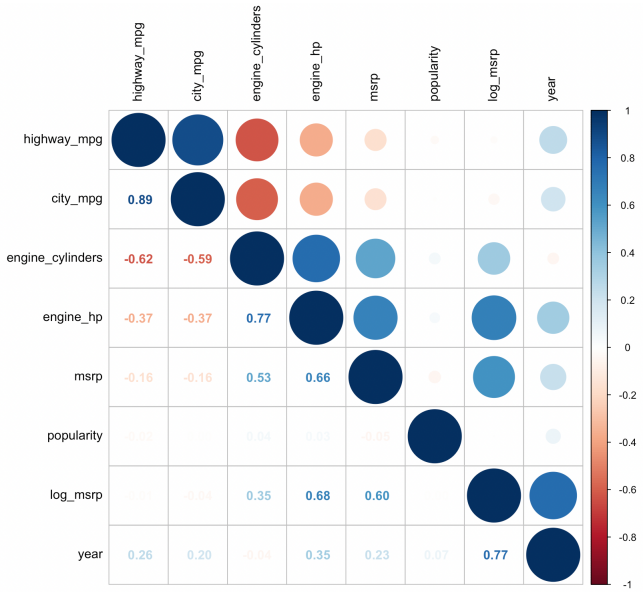


Fig. 7: Data understanding - General Viewing

```
'data.frame':   10000 obs. of  19 variables:
 $ id                 : int  569520 750365 199901 478866 731664 877557 930134 461006 68366 445911 ...
 $ age                : Factor w/ 4 levels "16-25","26-39",..: 4 1 1 1 2 3 4 2 3 3 ...
 $ gender             : Factor w/ 2 levels "female","male": 1 2 1 2 2 1 2 1 1 1 ...
 $ race               : Factor w/ 2 levels "majority","minority": 1 1 1 1 1 1 1 1 1 1 ...
 $ driving_experience : Factor w/ 4 levels "0-9y","10-19y",..: 1 1 1 1 2 3 4 1 3 1 ...
 $ education          : Factor w/ 3 levels "high school",..: 1 2 1 3 2 1 1 3 3 1 ...
 $ income             : Factor w/ 4 levels "middle class",..: 3 2 4 4 4 3 3 4 4 3 ...
 $ credit_score       : num  0.629 0.358 0.493 0.206 0.388 ...
 $ vehicle_ownership  : num  1 0 1 1 1 1 0 0 0 1 ...
 $ vehicle_year       : Factor w/ 2 levels "after 2015","before 2015": 1 2 2 2 2 1 1 1 2 2 ...
 $ married            : num  0 0 0 0 0 0 0 1 0 1 0 ...
 $ children           : num  1 0 0 1 0 1 1 1 0 1 ...
 $ postal_code        : int  10238 10238 10238 32765 32765 10238 10238 10238 10238 32765 ...
 $ annual_mileage     : num  12000 16000 11000 11000 12000 13000 13000 13000 14000 13000 11000 ...
 $ vehicle_type       : Factor w/ 2 levels "sedan","sports car": 1 1 1 1 1 1 1 1 1 1 ...
 $ speeding_violations: int  0 0 0 0 2 3 7 0 0 0 ...
 $ duis               : int  0 0 0 0 0 0 0 0 0 0 ...
 $ past_accidents     : int  0 0 0 0 1 3 3 0 0 0 ...
 $ outcome            : num  0 1 0 0 1 0 0 1 0 1 ...
```

2) Data cleaning: The non-values in this data set were in two variables: ”credit_score” and ”annual_mileage” (see Figure IV).

For ”credit_score”, we input the median of the ”credit_score” for each income group. On the other hand, the non-values in ”annual_mileage” we imputed the mean of the ”annual_mileage”. Doing these actions, we preserved the original 10,000 data set cases.

TABLE IV: Variables with NA values

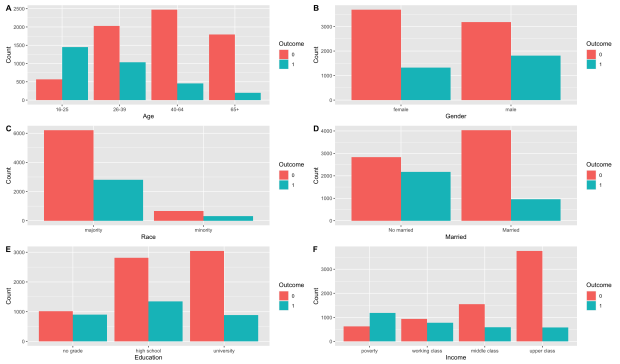
Variable	NA’s
credit_score	982
annual_mileage	957

During the preliminary bi-variate analysis (see Figure 8), it was possible to see some relevant insights, such as the influence of the age in an insurance claim, being the young people under 40 years old more probably to claim the

insurance.

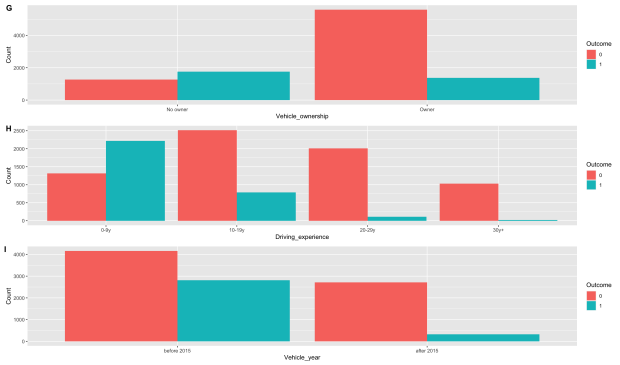
Also, being single is related and has bigger chances to claim insurance. Also, the majority of the people who claim insurance belong to a non-minority racial group. But a possible explanation of this is that the majority of the sample people belong to a non-minority race group.

Fig. 8: Bivariate analysis



Another important insight of this analysis (see Figure 9) is that less experienced drivers have more chances to claim car insurance. Therefore, the new cars (after 2015) have significantly fewer cases to be involved in an insurance claim.

Fig. 9: Bivariate analysis II



3) Feature Matrix: Finally, after recoding and fixing problems in the variable's type, we advanced to the final step and removed the residual variables such as "id" and "postal code". And also, all the variables were normalized by the formula for the right appliance of the models:

$$normalize = f(x) = \frac{x - min(x)}{max(x) - min(x)} \quad (1)$$

The final feature matrix has 16 variables and 10,000 cases (See Figure 10).

C. Car accident severity

1) Data understanding: In a general perspective, the data set has 47 variables and 1,516,064 cases. However, as we will see in the next step, the non-values was impossible to solve appropriately (see Figure 11).

Fig. 10: Feature Matrix

```
'data.frame':   10000 obs. of  16 variables:
 $ age                : num  1 0 0 0 0.333 ...
 $ gender             : num  0 1 0 1 1 0 1 0 0 0 ...
 $ race               : num  0 0 0 0 0 0 0 0 0 0 ...
 $ driving_experience : num  0 0 0 0 0.333 ...
 $ education          : num  0.5 0 0.5 1 0 0.5 0.5 1 1 0.5 ...
 $ income             : num  1 0 0.333 0.333 0.333 ...
 $ credit_score       : num  0.634 0.335 0.485 0.168 0.369 ...
 $ vehicle_ownership  : num  1 0 1 1 1 1 0 0 0 1 ...
 $ vehicle_year       : num  1 0 0 0 0 1 1 1 0 0 ...
 $ married            : num  0 0 0 0 0 0 1 0 1 0 ...
 $ children           : num  1 0 0 1 0 1 1 1 0 1 ...
 $ annual_mileage     : num  0.5 0.7 0.45 0.45 0.5 0.55 0.55 0.6 0.55 0.45 ...
 $ vehicle_type       : num  0 0 0 0 0 0 0 0 0 0 ...
 $ speeding_violations: num  0 0 0 0 0 0.0909 ...
 $ duis               : num  0 0 0 0 0 0 0 0 0 0 ...
 $ past_accidents     : num  0 0 0 0 0 0.0667 ...
```

Fig. 11: General Viewing

```
'data.frame': 1516064 obs. of  47 variables:
 $ id                    : Factor w/ 1516064 levels "A-2716600","A-2716601",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ severity              : int  3 2 2 2 2 3 2 2 2 2 ...
 $ start_time            : Factor w/ 1037092 levels "2016-02-08 00:37:08",..: 1 2 3 3 4 5 7 7 6 8 ...
 $ end_time              : Factor w/ 1161415 levels "2016-02-08 06:37:08",..: 1 2 3 3 4 5 7 7 6 8 ...
 $ start_lat             : num  40.1 39.9 39.1 39.1 41.1 ...
 $ start_lng             : num  -83.1 -84.1 -84.5 -84.5 -81.5 ...
 $ end_lat               : num  40.1 39.9 39.1 39.1 41.1 ...
 $ end_lng               : num  -83 -84 -84.5 -84.5 -81.5 ...
 $ distance_mi           : num  3.23 0.747 0.055 0.219 0.123 ...
 $ description           : Factor w/ 527655 levels " 1039 GOLDEN BEAR - BOT",..: 190539 98499 69664 69
88954 44564 44564 98490 18769 ...
 $ number                : num  NA NA NA NA NA ...
 $ street                : Factor w/ 93048 levels " 1 Mile Rd"," 1/2 Ave",..: 81271 73304 73321 89425
87102 50005 73321 73309 ...
 $ side                  : Factor w/ 2 levels "L","R": 2 2 2 2 2 2 2 2 2 2 ...
 $ city                  : Factor w/ 10658 levels "","Aaronsburg",..: 2556 2298 1725 1725 68 1725 1039
1801 ...
 $ county                : Factor w/ 1671 levels "Abbeville","Acadia",..: 539 1015 643 643 1452 643 32
392 ...
 $ state                 : Factor w/ 49 levels "AL","AR","AZ",..: 34 34 34 34 34 34 34 34 34 34 ...
 $ zipcode               : Factor w/ 177197 levels "","01001","01005-9392",..: 84662 88600 87269 87219
87212 87113 88156 86336 ...
 $ country               : Factor w/ 1 level "US": 1 1 1 1 1 1 1 1 1 1 ...
 $ timezone              : Factor w/ 5 levels "","US/Central",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ airport_code          : Factor w/ 1986 levels "","K01M","K04V",..: 1445 695 1142 1142 145 1142 953
...
 $ weather_timestamp     : Factor w/ 331749 levels "","2016-02-08 00:53:00",..: 2 4 3 3 5 6 7 7 8 9 ...
 $ temperature_f         : num  42.1 36.9 36 36 39 37 35.6 35.6 33.8 33.1 ...
 $ wind_chill_f          : num  36.1 NA NA NA NA 29.8 29.2 29.2 NA 30 ...
 $ humidity              : num  58 91 97 97 55 93 100 100 100 92 ...
 $ pressure_in           : num  29.8 29.7 29.7 29.7 29.6 ...
 $ visibility_mi         : num  10 10 10 10 10 10 10 3 0.5 ...
 $ wind_direction        : Factor w/ 25 levels "","Calm","CALM",..: 19 2 2 2 2 25 25 25 19 19 ...
 $ wind_speed_mph        : num  10.4 NA NA NA NA 10.4 8.1 8.1 2.3 3.5 ...
 $ precipitation_in      : num  0 0.02 0.02 0.02 NA 0.01 NA NA NA 0.08 ...
 $ weather_condition     : Factor w/ 117 levels "","Blowing Dust",..: 52 52 75 75 75 52 75 75 72 96 ...
 $ amenity               : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ bump                  : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ crossing              : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ give_way              : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ junction              : Factor w/ 2 levels "False","True": 1 1 2 2 1 1 1 1 1 2 ...
 $ no_exit               : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ railway               : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ roundabout            : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ station               : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ stop                  : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ traffic_calming       : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ traffic_signal        : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 2 1 1 1 ...
 $ turning_loop          : Factor w/ 1 level "False": 1 1 1 1 1 1 1 1 1 1 ...
 $ sunrise_sunset        : Factor w/ 3 levels "","Day","Night": 3 3 3 3 3 2 2 2 2 2 ...
 $ civil_twilight        : Factor w/ 3 levels "","Day","Night": 3 3 3 3 3 2 2 2 2 2 ...
 $ nautical_twilight     : Factor w/ 3 levels "","Day","Night": 3 3 3 3 2 2 2 2 2 2 ...
 $ astronomical_twilight: Factor w/ 3 levels "","Day","Night": 3 3 2 2 2 2 2 2 2 2 ...
```

2) Data cleaning: Because of the type of variables where the missing cases were presented, it was impossible to recode somehow. But, fortunately, the data set was big enough to eliminate these cases and train the model only with valid observations.

However, to avoid losing too much information, we first deleted the variables that did not meet the criteria for the analysis. After that, we use only valid cases. In that way, we had 572,485 missing values, and we remained a total of 943,579 cases (See Figure 12).

3) Feature Matrix: For defining the final feature matrix, we checked possible high correlations that may affect the

TABLE V: Variables with NA values

Variable	Number of NA's
number	1,046,095
precipitation_in	510,549
wind_chill_f	449,316
wind_speed_mph	128,862
humidity	45,509
visibility_mi	44,211
temperature_f	43,033
pressure_in	36,274

TABLE VI: Variables with NA values II

Variable	Number of NA's
precipitation_in	510,549
wind_chill_f	449,316
wind_speed_mph	128,862
humidity	45,509
visibility_mi	44,211
temperature_f	43,033
pressure_in	36,274

correct performance of the Naive Bayes model. As shown in Figure 13, the correlations between all the variables are reasonable lower and appropriate for the analysis, except for the relation between "wind_chill_f" and "temperature_f".

On the other hand, after the iterative process to consolidate the best subset for the feature matrix. It revealed that many variables affected the performance negatively. It means that the final Feature Matrix has a total of 12 variables and 943,532 observations (see Figure 14).

V. Results

A. Car price prediction

1) Multiple Regression: In Table VII, it is possible to see the contrast between the Null model and the Full model. The full model is where all the feature matrix has been used as explanatory variables.

Fig. 12: General Viewing

```
'data.frame': 943532 obs. of  34 variables:
 $ severity            : int  3 3 2 2 2 2 2 3 2 3 ...
 $ start_time          : Factor w/ 1037092 levels "2016-02-08 00:37:08",..: 1 5 8 10 11 13 14 17 19 20 ...
 $ end_time            : Factor w/ 1161415 levels "2016-02-08 06:37:08",..: 1 5 8 10 11 13 14 17 19 20 ...
 $ distance_mi         : num  3.23 0.5 0.521 0.826 0.307 0.07 2.59 0.999 0.477 0.471 ...
 $ side                : Factor w/ 2 levels "L","R": 2 2 2 2 2 2 2 2 2 2 ...
 $ city                : Factor w/ 10658 levels "","Aaronsburg",..: 2556 1725 1801 10278 1725 1801 4615 1927
8078 3433 ...
 $ county              : Factor w/ 1671 levels "Abbeville","Acadia",..: 539 643 392 539 643 392 618 539 27 783
...
 $ state               : Factor w/ 49 levels "AL","AR","AZ",..: 34 34 34 34 34 34 34 34 14 16 ...
 $ temperature_f       : num  42.1 37 33.1 32 33.8 33.1 33.8 33.8 28 26.6 ...
 $ wind_chill_f        : num  36.1 29.8 30 28.7 29.6 24.4 28.6 27 16.1 15.2 ...
 $ humidity            : num  58 93 92 100 100 96 93 100 88 80 ...
 $ pressure_in         : num  29.8 29.7 29.6 29.6 29.7 ...
 $ visibility_mi       : num  10 10 0.5 0.5 3 1.8 1 1 8 3 ...
 $ wind_direction      : Factor w/ 25 levels "","Calm","CALM",..: 19 25 19 23 11 23 23 23 23 23 ...
 $ wind_speed_mph      : num  10.4 10.4 3.5 3.5 4.6 11.5 5.8 8.1 16.1 13.8 ...
 $ precipitation_in    : num  0 0.01 0.08 0.05 0.03 0 0.01 0 0 0 ...
 $ weather_condition   : Factor w/ 117 levels "","Blowing Dust",..: 52 52 96 96 60 60 60 60 60 60 ...
 $ amenity             : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ bump                : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ crossing            : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ give_way            : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ junction            : Factor w/ 2 levels "False","True": 1 1 2 1 1 1 1 1 1 1 ...
 $ no_exit             : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ railway             : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ roundabout          : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ station             : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ stop                : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ traffic_calming     : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ traffic_signal      : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ turning_loop        : Factor w/ 1 level "False": 1 1 1 1 1 1 1 1 1 1 ...
 $ sunrise_sunset      : Factor w/ 2 levels "Day","Night": 2 1 1 1 1 1 1 2 2 2 ...
 $ civil_twilight      : Factor w/ 2 levels "Day","Night": 2 1 1 1 1 1 1 2 2 2 ...
 $ nautical_twilight   : Factor w/ 2 levels "Day","Night": 2 1 1 1 1 1 1 2 2 2 ...
 $ astronomical_twilight: Factor w/ 2 levels "Day","Night": 2 1 1 1 1 1 1 2 2 2 ...
 - attr(*, "na.action")= 'omit' Named int [1:572485] 2 3 4 5 7 8 9 11 14 17 ...
  ..- attr(*, "names")= chr [1:572485] "2" "3" "4" "5" ...
```

Fig. 13: Partial correlation

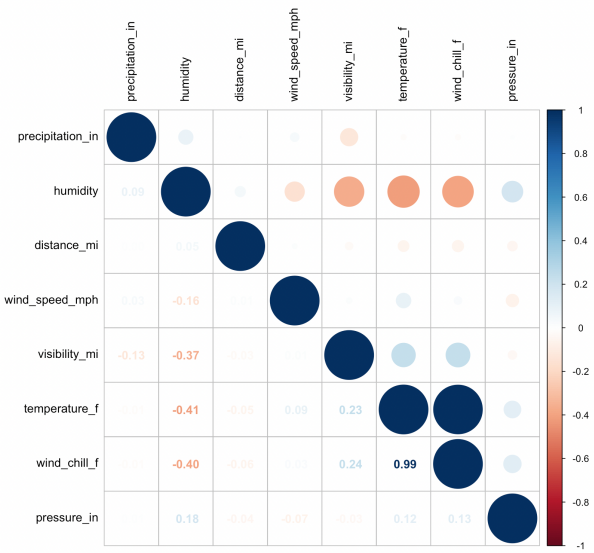


Fig. 14: Feature Matrix

```
'data.frame': 943532 obs. of  12 variables:
 $ severity       : Factor w/ 4 levels "1","2","3","4": 3 3 2 2 2 2 2 3 2 3 ...
 $ distance_mi    : num  3.23 0.5 0.521 0.826 0.307 0.07 2.59 0.999 0.477 0.471 ...
 $ state          : Factor w/ 49 levels "AL","AR","AZ",..: 34 34 34 34 34 34 34 34 14 16 ...
 $ temperature_f  : num  42.1 37 33.1 32 33.8 33.1 33.8 33.8 28 26.6 ...
 $ wind_chill_f   : num  36.1 29.8 30 28.7 29.6 24.4 28.6 27 16.1 15.2 ...
 $ humidity       : num  58 93 92 100 100 96 93 100 88 80 ...
 $ pressure_in    : num  29.8 29.7 29.6 29.6 29.7 ...
 $ visibility_mi  : num  10 10 0.5 0.5 3 1.8 1 1 8 3 ...
 $ wind_speed_mph : num  10.4 10.4 3.5 3.5 4.6 11.5 5.8 8.1 16.1 13.8 ...
 $ precipitation_in: num  0 0.01 0.08 0.05 0.03 0 0.01 0 0 0 ...
 $ amenity        : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ sunrise_sunset : Factor w/ 2 levels "Day","Night": 2 1 1 1 1 1 1 2 2 2 ...
```

However, often more variables do not mean a better model, but having the contrast between these extremes is a good reference for the modelling process. Every model which fewer variables a better-explained performance will be considered a better model.

The process of modelling performed using the "Backforward" method means that considering the full model, we removed the variables with a lower contribution.

In Table VIII, we tried five new models to reduce the number of explanatory variables. The central insight is that even when model 6 uses fewer variables, the explanatory power measure (R^2) is almost the same.

Finally, considering model 6 as a good model, we tried to improve it by looking for a slight change, finding hiding effects in the data. There we got an improvement adding a quadratic effect of the year. As shown in Table IX, the R^2 is 0.90, which is considered an excellent model performance.

2) Regression Tree: On the other hand, we used the Regression Tree as an alternative model for predicting the car price.

TABLE VII: Comparison between Null model vs. Full model

	Model 0	Model 1
(Intercept)	10.11^{***}	-196.83^{***}
	(0.01)	(2.08)
make		-
year		0.10^{***}
		(0.00)
engine_fuel_typeelectric		1.09^{***}
		(0.32)
engine_fuel_typeflex-fuel (premium unleaded recommended/E85)		-0.30^{**}
		(0.11)
engine_fuel_typeflex-fuel (premium unleaded required/E85)		-0.39^{***}
		(0.09)
engine_fuel_typeflex-fuel (unleaded/E85)		-0.57^{***}
		(0.05)
engine_fuel_typeflex-fuel (unleaded/natural gas)		-0.64^{**}
		(0.19)
engine_fuel_typenatural gas		-0.27
		(0.30)
engine_fuel_typepremium unleaded (recommended)		-0.36^{***}
		(0.04)
engine_fuel_typepremium unleaded (required)		-0.02
		(0.04)
engine_fuel_typeregular unleaded		-0.51^{***}
		(0.04)
engine_hp		0.00^{***}
		(0.00)
engine_cylinders		0.06^{***}
		(0.01)
transmission_typeAUTOMATED_MANUAL		0.12^{***}
		(0.02)
transmission_typeAUTOMATIC		0.20^{***}
		(0.01)
transmission_typeDIRECT_DRIVE		0.13
		(0.30)
driven_wheelsrear wheel drive		-0.13^{***}
		(0.02)
driven_wheelsfour wheel drive		-0.09^{***}
		(0.01)
number_of_doorsThree		-0.30^{***}
		(0.03)
number_of_doorsFour		-0.01
		(0.01)
vehicle_sizeMidsize		-0.06^{***}
		(0.01)
vehicle_sizeLarge		-0.11^{***}
		(0.02)
highway_mpg		-0.04^{***}
		(0.00)
city_mpg		0.02^{***}
		(0.00)
R^2	0.00	0.86
Adj. R^2	0.00	0.86
Num. obs.	9531	9531

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Note: we hide the effect for the 48 brands ("make" variable) to visualise the table.

TABLE VIII: Backforward Modelling Analysis

	Model 2	Model 3	Model 4	Model 5	Model 6
(Intercept)	-186.18^{***}	-185.02^{***}	-186.10^{***}	-186.19^{***}	-180.56^{***}
	(2.00)	(2.00)	(1.99)	(1.96)	(1.48)
make	***	***	***	***	***
	–	–	–	–	–
year	0.10^{***}	0.10^{***}	0.10^{***}	0.10^{***}	0.09^{***}
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
engine_fuel_typeelectric	1.24^{***}				
	(0.32)				
engine_fuel_typeflex-fuel (premium unleaded recommended/E85)	-0.14				
	(0.11)				
engine_fuel_typeflex-fuel (premium unleaded required/E85)	-0.32^{***}				
	(0.09)				
engine_fuel_typeflex-fuel (unleaded/E85)	-0.44^{***}				
	(0.05)				
engine_fuel_typeflex-fuel (unleaded/natural gas)	-0.48^{*}				
	(0.20)				
engine_fuel_typenatural gas	-0.28				
	(0.31)				
engine_fuel_typepremium unleaded (recommended)	-0.24^{***}				
	(0.04)				
engine_fuel_typepremium unleaded (required)	0.09^{*}				
	(0.04)				
engine_fuel_typeregular unleaded	-0.40^{***}				
	(0.04)				
engine_hp	0.00^{***}	0.00^{***}	0.00^{***}	0.00^{***}	0.00^{***}
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
engine_cylinders	0.07^{***}	0.05^{***}	0.05^{***}	0.05^{***}	
	(0.01)	(0.01)	(0.01)	(0.01)	
transmission_typeAUTOMATED_MANUAL	0.14^{***}	0.16^{***}	0.16^{***}	0.15^{***}	
	(0.02)	(0.03)	(0.03)	(0.03)	
transmission_typeAUTOMATIC	0.23^{***}	0.20^{***}	0.20^{***}	0.19^{***}	
	(0.01)	(0.01)	(0.01)	(0.01)	
transmission_typeDIRECT_DRIVE	0.43	1.64^{***}	1.66^{***}	1.52^{***}	
	(0.30)	(0.13)	(0.13)	(0.13)	
driven_wheelsrear wheel drive	-0.07^{***}	-0.03			
	(0.02)	(0.02)			
driven_wheelsfour wheel drive	0.00	0.02			
	(0.01)	(0.01)			
number_of_doorsThree	-0.23^{***}	-0.22^{***}	-0.23^{***}		
	(0.03)	(0.03)	(0.03)		
number_of_doorsFour	0.00	-0.01	0.00		
	(0.01)	(0.01)	(0.01)		
vehicle_sizeMidsize	-0.08^{***}	-0.08^{***}	-0.08^{***}	-0.08^{***}	
	(0.01)	(0.01)	(0.01)	(0.01)	
vehicle_sizeLarge	-0.12^{***}	-0.15^{***}	-0.15^{***}	-0.15^{***}	
	(0.02)	(0.02)	(0.02)	(0.02)	
city_mpg	-0.01^{***}	-0.01^{***}	-0.01^{***}	-0.01^{***}	
	(0.00)	(0.00)	(0.00)	(0.00)	
R^2	0.85	0.84	0.84	0.84	0.83
Adj. R^2	0.85	0.84	0.84	0.84	0.83
Num. obs.	9531	9531	9531	9531	9531

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Note: we hide the effect for the 48 brands ("make" variable) to visualise the table.

TABLE IX: Best Model

	Best Model - Model 7
(Intercept)	-23770.46^{***}
	(305.73)
make	***
	–
year	23.62^{***}
	(0.30)
engine_hp	0.00^{***}
	(0.00)
year^2	-0.01^{***}
	(0.00)
R^2	0.90
Adj. R^2	0.90
Num. obs.	9531

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Note: we hide the effect for the 48 brands ("make" variable) to visualise the table.

This model is different from the Linear Regression Model because it uses the mechanism of splitting the data into different subgroups until getting a good prediction performance. As shown in Figure XX, it is possible to see the Regression Tree summary process.

Fig. 15: Regression Tree summary I

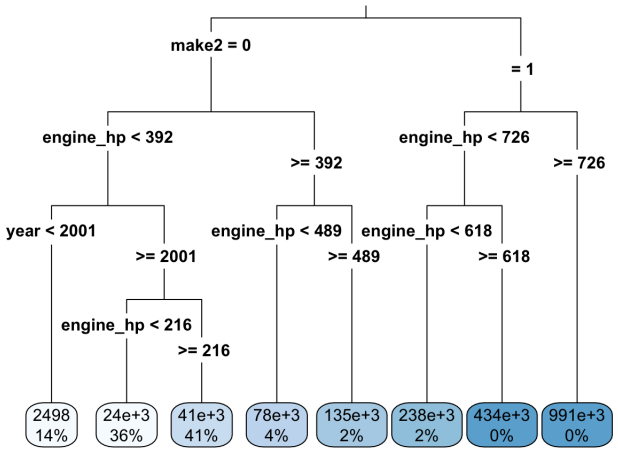
```
Call:
rpart(formula = msrp ~ make2 + year + engine_hp, data = data_train,
    method = "anova")
  n= 9531

          CP nsplit rel error    xerror       xstd
1 0.47205532      0 1.0000000 1.0000968 0.17404055
2 0.10007093      1 0.5279447 0.5303118 0.12667778
3 0.07901029      2 0.4278738 0.5518515 0.12777674
4 0.03656162      3 0.3488635 0.3890401 0.08645855
5 0.03040450      4 0.3123018 0.3862250 0.08644367
6 0.01482713      5 0.2818973 0.3458873 0.08121208
7 0.01280626      6 0.2670702 0.3250702 0.07776165
8 0.01000000      7 0.2542639 0.3165507 0.07767699

Variable importance
    make2 engine_hp      year
       61        35         4

Node number 1: 9531 observations,    complexity param=0.4720553
  mean=40626.49, MSE=3.741193e+09
  left son=2 (9259 obs) right son=3 (272 obs)
  Primary splits:
      make2     < 0.5    to the left,  improve=0.47205530, (0 missing)
      engine_hp < 449.5  to the left,  improve=0.38141740, (0 missing)
      year      < 2000.5 to the left,  improve=0.06318446, (0 missing)
  Surrogate splits:
      engine_hp < 650.5  to the left,  agree=0.973, adj=0.048, (0 split)

Node number 2: 9259 observations,    complexity param=0.07901029
  mean=33423.65, MSE=7.373054e+08
  left son=4 (8635 obs) right son=5 (624 obs)
  Primary splits:
      engine_hp < 392    to the left,  improve=0.4126870, (0 missing)
      year      < 2000.5 to the left,  improve=0.2181549, (0 missing)

Node number 3: 272 observations,    complexity param=0.1000709
  mean=285814.4, MSE=4.411169e+10
  left son=6 (265 obs) right son=7 (7 obs)
  Primary splits:
      engine_hp < 725.5  to the left,  improve=0.29739520, (0 missing)
      year      < 2011.5 to the right, improve=0.03472012, (0 missing)

Node number 4: 8635 observations,    complexity param=0.0304045
  mean=28734.49, MSE=2.777784e+08
  left son=8 (1332 obs) right son=9 (7303 obs)
  Primary splits:
      year      < 2000.5 to the left,  improve=0.4519869, (0 missing)
      engine_hp < 236.5  to the left,  improve=0.4090777, (0 missing)
  Surrogate splits:
      engine_hp < 97.5   to the left,  agree=0.86, adj=0.093, (0 split)
```

One of the main advantages of this kind of model is there are easy to explain. For example, Figure 16 shows that the ML model split into different subgroups where the tree applied a predictor factor for the final prediction. It is noticed

```
Node number 5: 624 observations,    complexity param=0.01280626
  mean=98312.93, MSE=2.58141e+09
  left son=10 (404 obs) right son=11 (220 obs)
  Primary splits:
      engine_hp < 489    to the left,  improve=0.28348460, (0 missing)
      year      < 2016.5 to the right, improve=0.01687625, (0 missing)

Node number 6: 265 observations,    complexity param=0.03656162
  mean=267199.1, MSE=1.939972e+10
  left son=12 (225 obs) right son=13 (40 obs)
  Primary splits:
      engine_hp < 618    to the left,  improve=0.2535904, (0 missing)
      year      < 2011.5 to the right, improve=0.0235594, (0 missing)

Node number 7: 7 observations
  mean=990536.3, MSE=4.698847e+11

Node number 8: 1332 observations
  mean=2497.7, MSE=3377278

Node number 9: 7303 observations,    complexity param=0.01482713
  mean=33519.84, MSE=1.793749e+08
  left son=18 (3391 obs) right son=19 (3912 obs)
  Primary splits:
      engine_hp < 216    to the left,  improve=0.40359230, (0 missing)
      year      < 2010.5 to the left,  improve=0.05687725, (0 missing)
  Surrogate splits:
      year < 2006.5 to the left,  agree=0.597, adj=0.131, (0 split)

Node number 10: 404 observations
  mean=78350.48, MSE=7.311284e+08

Node number 11: 220 observations
  mean=134971.2, MSE=3.903577e+09

Node number 12: 225 observations
  mean=237625.6, MSE=5.910312e+09

Node number 13: 40 observations
  mean=433550.1, MSE=6.268539e+10

Node number 18: 3391 observations
  mean=24381.07, MSE=3.983822e+07

Node number 19: 3912 observations
  mean=41441.5, MSE=1.651809e+08
```

that the variable make was recoded in two groups, only to visualise the plot easily.

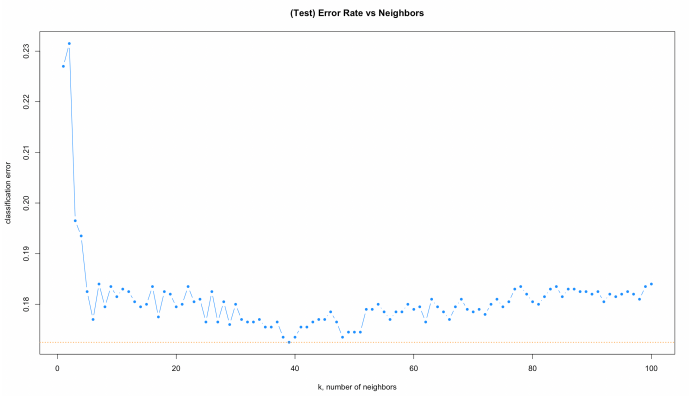
Fig. 16: Regression Tree plot



B. Car insurance claim

1) KNN models: The k-nearest neighbour is a non-parametric classification ML method, which use Euclidean distance to determine the nearest neighbour. As shown in Figure 17, the best value for K to minimise the class error is 39.

Fig. 17: KNN Performance



As shown in Table X, the model's performance is being tested. As a result, 1,271 cases were classified correctly as No Claim and 385 as Claim. However, 193 cases were classified as No claim, and 151 as Claim wrongly, respectively.

TABLE X: Confusion Matrix KNN model

	KNN Prediction	
	No Claim	Claim
Test - No Claim	1271	151
Test - Claim	193	385

2) Random Forest: As a result of the Random Forest Analysis, 1,256 cases were predicted as No Claim, and 423 cases as Claim correctly, respectively.

However, 155 cases were predicted as No Claim, and 166 cases as Claim wrongly, respectively.

Fig. 18: Random Forest Analysis

```
Total Observations in Table:  2000


                 | predicted
data_test_labels |  No Claim |     Claim | Row Total |
-----------------|-----------|-----------|-----------|
        No Claim |      1256 |       166 |      1422 |
                 |     0.883 |     0.117 |     0.711 |
                 |     0.891 |     0.281 |           |
                 |     0.628 |     0.083 |           |
-----------------|-----------|-----------|-----------|
           Claim |       153 |       425 |       578 |
                 |     0.265 |     0.735 |     0.289 |
                 |     0.109 |     0.719 |           |
                 |     0.076 |     0.212 |           |
-----------------|-----------|-----------|-----------|
    Column Total |      1409 |       591 |      2000 |
                 |     0.705 |     0.295 |           |
-----------------|-----------|-----------|-----------|
```

C. Car accident severity

As a result of Naive Bayes ML Classification, the predictor had overestimated the values of 1 in 6,153 cases and 2 in 5,975. However, the model underestimated the value classification of 3 in 7,459 and 4 in 4,669, respectively (see Figure 19).

Fig. 19: Confusion Matrix

Total Observations in Table: 188707

data_test$severity	predicted_nb 1	2	3	4	Row Total
1	1792	2986	193	75	5046
	0.355	0.592	0.038	0.015	0.027
	0.160	0.018	0.031	0.017	
	0.009	0.016	0.001	0.000	
2	8198	146704	2945	3051	160898
	0.051	0.912	0.018	0.019	0.853
	0.732	0.879	0.472	0.695	
	0.043	0.777	0.016	0.016	
3	792	10018	2561	334	13705
	0.058	0.731	0.187	0.024	0.073
	0.071	0.060	0.410	0.076	
	0.004	0.053	0.014	0.002	
4	417	7165	547	929	9058
	0.046	0.791	0.060	0.103	0.048
	0.037	0.043	0.088	0.212	
	0.002	0.038	0.003	0.005	
Column Total	11199	166873	6246	4389	188707
	0.059	0.884	0.033	0.023	

VI. Evaluation

A. Car price prediction

The R^2 of the best Regression Model is 0.90, which mean that 90% of the variance in the car price can be explained as a linear combination of the feature predictors. Therefore, we solved many problems during the modelling process, precisely multicollinearity problems between the feature matrix variables.

However, we preserve some problems in the homogeneity of the variance. Even though Figure XX shows the evaluation of all assumptions in Regression Models, most of them look at an acceptable level.

Moreover, we showed a scatterplot evaluating the relationship between the target variable in the testing data set and the predicted values, and the prediction looks very good in the performance, the correlation coeficient is 0.95 (see Figure 21).

To sum up, the performance of the Regression Model and Tree Regression Model can be compared using the Mean Absolute Error (MAE).

In this comparison, the Regression Model has an MAE of $9,006, and the Tree Regression Model has an MAE of $10,540. And for this reason, the best performance as a predictor of the car price is the Linear Regression Model.

Fig. 20: Regression Diagnostic

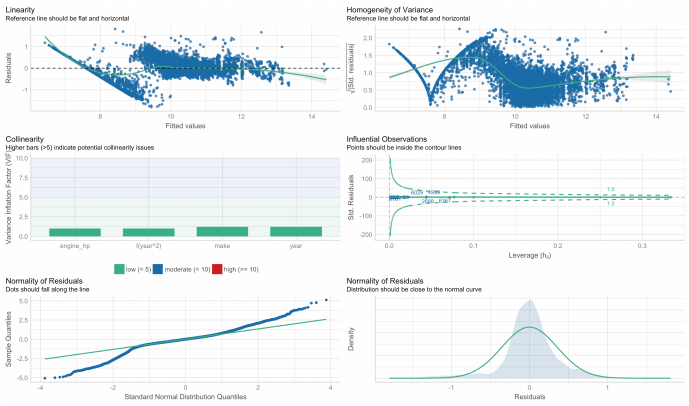
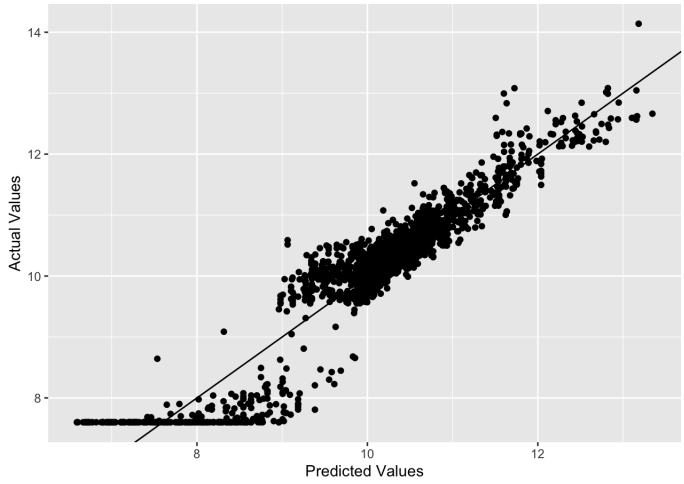


Fig. 21: Comparison between Test values and Prediction values



B. Car insurance claim

After comparing both Confusion Matrix such as KNN prediction and Random Forest prediction, the result showed that the KNN model's accuracy is 0.828, and for the Random Forest model is 0.841. Therefore, it means the Random Forest model has a slightly better performance for the classification of the Car Insurance Claim.

C. Car accident severity

Finally, we only applied the Naive Bayes model to the Car accident severity data set, and it is impossible to compare the performance.

However, the model has many challenges that we should consider in future work, such as looking for an ML model that can assess the variable's ordinal nature and a model that can perform a multinomial classification.

As a result of the confusion matrix, the model's accuracy is 0.8. This result is negative because if we used a predicted vector with only values of two, we would expect an accuracy of 0.85. That is something that an improvement of the model should consider in future work.

VII. Conclusions and future work

The Evaluation section for the Car price prediction seems like the regression model has a robust performance with an R^2 of 0.90 and a lower MAE (compared to the MSE and RMSE). However, the model can improve more if we apply more effort to solve the problems in the homogeneity of the variance, especially in the lower values.

Secondly, after many improvements in the modelling background to get better the knn model, the random forest performs better with an accuracy of 0.841 over 0.828.

On the other hand, it was impossible to compare the Naive Bayes model. However, here is the most evident field for improvement. It can be using different algorithms as the related work is shown or defining better adjustment on the feature matrix considering relevant variables in the literature such as traffic volume, driver's age and car's age.

Moreover, in future work, it is clear that we have to perform a more significant set of ML models in the first two data set to decide which one performs better.

For instance, in the related work, the support vector machine demonstrated have good performance in the predictions task or use models such as gradient boosted regression tree, which was not even considered in this project.

Also, many of the related work had used more sophisticated methods like Deep Learning about the classification task. Even though the data set initially does not have more complex information, it can inspire future work and answer more complex research questions and improve ML field skills.

Finally, we must use other models in future work to have a reference for comparing the model performance. Also, it is essential to find models that respect the variable's ordinal nature and do not affect the high concentration in one of the levels of the factor.

References

[1] The Royal Society. Machine Learning: the power and promise of computers that learn by example. Summary Report. 2017.

[2] James, G., Witten, D., Hastie, T., and Tibshirani, R. An Introduction to Statistical Learning with applications in R. 2014.

[3] Lantz, B. Machine Learning with R (2nd ed), Packt Publishing ltd. 2015.

[4] A. Das Mou, P. K. Saha, S. A. Nisher and A. Saha, "A Comprehensive Study of Machine Learning algorithms for Predicting Car Purchase Based on Customers Demands," 2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), 2021, pp. 180-184, doi: 10.1109/ICICT4SD50815.2021.9396868.

[5] N. Monburinon, P. Chertchom, T. Kaewkiriya, S. Rungpheung, S. Buya and P. Boonpou, "Prediction of prices for used car by using regression models," 2018 5th International Conference on Business and Industrial Research (ICBIR), 2018, pp. 115-119, doi: 10.1109/ICBIR.2018.8391177..

[6] Enis Gegic, Becir Isakovic, Dino Keco, Zerina Masetic, Jasmin Kevric, Car Price Prediction using Machine Learning Techniques. TEM Journal. Volume 8, Issue 1, Pages 113-118, ISSN 2217-8309, DOI: 10.18421/TEM81-16, February 2019.

[7] Hanafy, Mohamed,and Ruixing Ming. 2021. Machine Learning Approaches for Auto InsuranceBigData.Risks9: 42. https://doi.org/10.3390/risks9020042.

[8] Hui Dong Wang. 2020. Research on the Features of Car Insurance Data Based on Machine Learning. Procedia Computer 166: 582-587.

[9] R. Singh, M. P. Ayyar, T. V. Sri Pavan, S. Gosain and R. R. Shah, "Automating Car Insurance Claims Using Deep Learning Techniques," 2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM), 2019, pp. 199-207, doi: 10.1109/BigMM.2019.00-25.

[10] Atwah, Abdulrahman and Al-Mousa, Amjed, Car Accident Severity Classification Using Machine Learning. IEEE. 2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT) Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), 2021 International Conference on. :186-192 Sep, 2021.

[11] Geyik, Buket and Kara, Medine, Severity Prediction with Machine Learning Methods. IEEE. 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA) Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2020 International Congress on. :1-7 Jun, 2020.

[12] Al Mamlook, Rabia Emhamed, and Abdulhameed, Tiba Zaki, and Hasan, Raed, and Al-Shaikhli, and Hasnaa Imad, and Mohammed, Ihab, and Tabatabai, Shadha. Utilizing Machine Learning Models to Predict the Car Crash Injury Severity among Elderly Drivers. IEEE. 2020 IEEE International Conference on Electro Information Technology (EIT) Electro Information Technology (EIT), 2020 IEEE International Conference on. :105-111 Jul, 2020.

[13] Brachman and Anand, 1996; Fayyad, 1996; Fayyad et al., 1996.