AN-NAJAH NATIONAL UNIVERSITY

FACULTY OF
ENGINEERING AND INFORMATION TECHNOLOGY
COMPUTER SCIENCE DEPARTMENT

GRADUATION PROJECT

Palestine Vehicle Price Prediction Using Different Machine learning Models and Deep Neural Networks - SOUQY APP

Author:
Naser Shanti
Akram Assi
Hamza Shakhshir

Supervisor:
Dr.Adnan Salman

Report Submitted in Partial Fulfillment of the Requirements for Bachelor degree of Computer Science.

June 1, 2021

Acknowledgements

In performing our project, we had to take the help and guideline from our Superviser, who deserve our greatest gratitude. The completion of this work gives us much powerful. We would like to show our gratitude to Dr. Adnan Salman for giving us a good guidance and consideration Time.

We would like also to thank The teaching staff for his effort during the academic period.

We also extend our deepest gratitude to our wonderful family who gave us support and love

Contents

List of Figures

Abstract

In the last five years, Palestine has seen an increase in the field of Selling and trading vehicles. This matter has pushed many people to buy a vehicle, but because lack of experience among people as the process of buying requires effort and knowledge from a technical expert and because there are several factors that affect the real price of the vehicle. We suggest in this project several machine learning techniques (Deep Neural Network, Support Vector Machine, KNeighborsRegressor, Random Forest, and Gradient Boosted) to obtain an approximate value for the vehicle price so that the person knows the amount of money that he must spend and why there's a difference between prices for the vehicle from the same class.

The data used for the prediction was collected from a Palestinian web Site شو بدك من فلسطين using Python web scraper(bs4) with a total of 200000 samples and 26 features. Then we Trained This Data in different Model to get the best Model we needed. Finally, we tested a new Data on Models set and The results were varied with Maximum Precision of (89.7%)

As a step of combine between Theories and product we integrate a prediction model with a Completed Flutter Mobile Application used to Selling vehicle and publishing ads.

1 Introduction

Due to a large number of (used, imported) cars the process of deciding whether a car is worth the posted price can be difficult. Several factors including distance(the number of kilometers it has run), make, model, Type(sedan, hatch, ..), fuel-type, history(The origin of the car it's private or taxi or commercial, etc), year, previous owners, payment-method, etc. can influence on the actual worth of a car. From the perspective of a shopper, it is a dilemma to Knowing the real price of the car appropriately. So, based on prepared data the aim is to use machine learning algorithms to develop models for predicting the price of the vehicle.

The challenging point of this project was to clean the data and prepared perfectly to be ready for training, It took a long time since there's a huge misspelling and different nickname for the same car, in addition, Mathematical problems in the value of distance that car run, even that a cleaning perform was done on some of the Target value(price).

This work aims to prepare a real and correct large data set of the cars in Palestine that can be used to generate more projects, this data can be used to solve and analyzed a lot of situation, for example, it can use by people to know the price change for a car over a time so they decide to buy it or not, it can also be used for car dealers for know which is the more stable car in price, it can modify this data to know which car consume more petrol via the distance that car run, analyze details that each car in data provide may generate a reason why this car is less/more average price, insurance company's, etc.., we aimed to use this data to predict the actual price for a car.

The importance of this work it gives a different model to get the best accuracy, in addition, we integrate this work with a real mobile application used to Selling cars and publishing ads hoping to get more data from ads to enhance model accuracy.

The next pages contain a brief description about the challenges that faces this project ,literature review which compare between the similar project on the global , methodology contain the details about how creating this project step by step , machine learning part , our mobile application and how to integrate with best model and at the end there is a conclusion learned from this experiment.

2 Constraints and Standards

2.1 Constraints

2.1.1 Data

The biggest problem that we faced its no enough resources to scrape the data from it. In addition, the website that we used to scrape data from it does not have any restriction in publishing the advertisement so this leads to a lack of credibility, For example, the user can input a Text entry in numeric filed and vice versa, so this led to a big problem later.

2.1.2 Training

We had some problems in training the Data because it takes a long time to get the result, we used Google-Colab to run some attempts but sometimes the server was Hang up or the Network failed.

2.1.3 Flutter & Dart For Mobile application

We don't have any previous experience in Flutter framework and Dart language, in addition, the resources for learning was too little because there's a new version continuously, so any problem we faced had a long Time and Deep search to solve.

2.2 Standards

2.2.1 TensorFlow

"TensorFlow is a free and open-source software library for machine learning.Its flexible architecture so its easy allow deployment across different platforms(CPUs, GPUs, TPUs) developed by researchers and engineers from the Google Brain team within Google's AI organization theres a high-level neural network APIs that can run on top of TensorFlow such as Keras"

2.2.2 Keras: The Python Deep Learning library

"Keras is a neural network library built in python so its more user-friendly than TensorFlow but it capable of running on top of TensorFlow its Provides only high-level APIs for building and training models."

- *We used Keras to build Model for prediction*

2.2.3 beautiful-soup

"Beautiful Soup is a Python package for parsing HTML and XML documents. It creates a parse tree for parsed pages that can be used to extract data from

HTML, which is useful for Scrapping."

- *We used BS4 to scrape the Data From Website*

2.2.4 Flutter & Dart For Mobile application

"Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, Mac, Windows,website.
Dart is a programming language designed by Google and it is the language used in Flutter framework"

- *We used Dart& Flutter to build mobile app*

2.2.5 Firebase

"Firebase is a platform developed by Google for helps you build, improve, and grow your app. it used to Store and sync data with (NoSQL) cloud real-time database."

- *We used Firebase for data,image storage and user authentication process*

3 Literature Review

Making a prediction Model for the price car isn't a new topic and there is a lot of researches done on it around the world but this project is the first project done in Palestine. In addition, it takes more care for the relationship between buyer and seller a lot of new features we added in our project and it wasn't used in the similar previous works, such as Day life that use as an age for each car, we used 13 different class for cars that increase the correct prediction, payment-method have a huge effect in price prediction was used. in addition, Neural Network was added to our project.

There are similar projects done by famous universities which have the same goals as this project but have a difference in the features and the Technique which been applied.

There is a lot of project dose the same as this one but no similar project has been implemented in our country, so the importance for This work that is simulate a real problem that people face every day. In addition, this project used different Techniques and it tries to get the best features that used on the other similar projects.

4 Methodology

In this chapter, we will describe everything we did while making this project. The first section will talk about the Scraping Data and how did?, the second one is about the Complex cleaning that is done on the Data, the third section describe the processes from End to End to build Machine learning Models, Keras Model and Run Successfully the fourth section is for comparison between all Result to decide which model is suitable to integrate with next section, Finally the last path integrates the prediction result with Mobile Application.

4.1 Scraping Data

As a simple definition and procedure way. Data Scraping or its also Called Web Scraping its a ”Technique in which a computer program extracts data from output generated from another program”, The process of web scraping implementation can be complex if you don’t understand the syntax of the page and it takes a long time without parallelizing the code.

The operation can be done in 3 steps:
1. Some pieces of written code sent an HTTP GET request to pull the information from a specific website.
2. Once the website responds, the scrapper has parsed an HTML page that contains the requested data.
3. in This step data now was downloaded in our machine and we reuse it as we want

Note:- Code was written with parallelism way —> GitHub

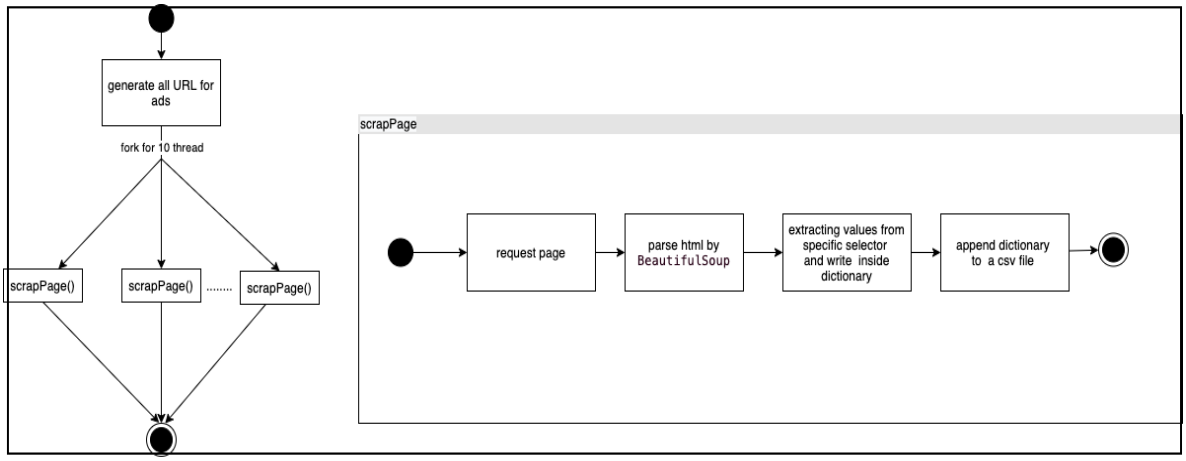


Figure 1: shows scraping process

اسم السيارة	سنة الإنتاج	السعر	اللون	نوع الوقود	أصل السيارة	رخصة السيارة	نوع الجير	قوة المانور	عداد السيارة	صحاب سابقو	تاريخ نشر الإعلان	إضافات	add_info	url	عدد الركاب	وسيلة الدفع	معروضة	الزجاج
فولكسفاجن باسات	1999	28500	كحلي	بنزين	خصوصي	فلسطينية	اوتوماتيك	1600			2016-09-20	تم تغيير قمرة عامود ع مكيف إغلاق مركزي,جهاز		https://shobiddak.c	4+1	نقدا فقط	للبيع أو التبديل	الكتروني
فولكسفاجن باسات	2001	43000	رمادي	بنزين	خصوصي	فلسطينية	اوتوماتيك	1600	100000	يد ثانية	2016-09-20	سيارة باسات للبيع مو مكيف إغلاق مركزي,جهاز		https://shobiddak.c	4+1	نقدا فقط	للبيع أو التبديل	الكتروني
بي ام دبليو 318	1998	10000	أبيض	بنزين	خصوصي	نمرة صفراء	اوتوماتيك	1800	شغال 000		2016-09-29	السيارة الف الف قل الة مكيف إغلاق مركزي,جهاز		https://shobiddak.c	5+1	نقدا فقط	للبيع أو التبديل	الكتروني
كيا بيكانتو	2008	49000	فضي	ديزل	خصوصي	فلسطينية	عادي	1500			2016-09-20	مكيف إغلاق مركزي,جهاز		https://shobiddak.c	4+1	نقدا فقط	للبيع أو التبديل	الكتروني
سانغ يونغ ركستون	2006	75000	أبيض عاجي	ديزل	خصوصي	فلسطينية	اوتوماتيك	2900	الف فقط 100	3	2016-09-19	جيب سانغ يونغ ريكه مكيف إغلاق مركزي,جهاز		https://shobiddak.c	6+1	إمكانية التقسيـ	للبيع أو التبديل	الكتروني
فولكسفاجن باسات	2010	960000	أسود ميتالك	ديزل	خصوصي	نصف اوتوماتيك	فلسطينية	2000	195000 km	يد ثانية	2016-09-19	إمكانية التبديل على سم مكيف إغلاق مركزي,جهاز		https://shobiddak.c	4+1	نقدا فقط	للبيع أو التبديل	الكتروني
ستروين بيكاسو C4	2013	100000	فيراني	ديزل	خصوصي	فلسطينية	اوتوماتيك	1600		0	2016-09-20	مكيف إغلاق مركزي,جهاز		https://shobiddak.c	6+1	نقدا فقط	للبيع فقط	الكتروني
بي ام دبليو X5	2008	260000	زيتي	ديزل	خصوصي	فلسطينية	اوتوماتيك	3000	220000	ثانيه	2016-09-20	مكيف إغلاق مركزي,جهاز		https://shobiddak.c	6+1	إمكانية التقسيـ	للبيع أو التبديل	الكتروني
فولكسفاجن كبينة	1985	35000	أصفر	ديزل	تجاري	فلسطينية	عادي	1900			2016-09-20	مطور باسات مرخص ، فرش جلد		https://shobiddak.c	5+1	نقدا فقط	للبيع فقط	يدوي

Figure 2: shows the Data when scrape, before any cleaning

4.2 Data Cleaning and Preprocessing

For this project, we are collect the Data from a Palestinian web Site فلسطين شو بدك من . The dataset contains the prices and attributes of over 200,000 advertisement was Published on the website. Our dataset contains 17 features for each advertisement (car-name,year,price,colour,fuel-type,history of the vehicle,gear-type, engine-size,kilometres that vehicle run,number of previous owners,vehicle ad date publish, accessories features,notes,URL,number of passengers,payment-method,windows)that can affect the price of the vehicle, This chapter will explain all cleaning process that done in each column.

```
Final Feature that Must use --->

Car_Name --- >                                  ===   make_model_trim     [ text ]
Year   --- >                                    ===   year               [ int  ]
Price  --- >                                    ===   price              [ int  ]
Color     --- >                                 ===   color              [ text ]
Fuel Type   --- >                               ===   fuel               [ diesel, petrol ]
Car Origin ( Private , Taxi )  --- >            ===   history            [ private, taxi, rental ]
Gear Type ( Automatic , Manuel )  --- >         ===   gear               [automatic, manual]
Engine power --- >                              ===   engine_size        [ in cc ]


Distance that car run  --- >                    ===  Kilometers          [ Kilometers ]
Who many user was used the car --- >            ===  owners              [ number ]
The Date of publishing the Ad  --- >            ===  ad_date             [ date   ]
Accessories --- >                               ===  accessories.        [boolean list ]
notes  --- >                                    ===  information about car  [text]
Number of passenger --- >                       ===  passengers          [int ]
The way of payment --->                         ===  payment_method      [ cash, installment ]
Windows (electric , Manuel) --- >               ===  windows type        [text]
wheels --- >                                    ===  wheels              [ text ]
```

Figure 3: features that determine to used in our project

4.2.1 Remove Noise in Data

This step was used to remove the noise row which affects negatively on the final result such as vehicle from outside region نمرة صفراء and the car that is considered as illegal car مشطوبة. as follow ...

Note:- Most illegal car are from outside our regions.

فسبة sym غير قانونية ماتور 125 بودي 200 صلاة النبى عنها بتجنن مش ناقصها اشى محافظ عليها ومعتنى فيها سبب البيع ازمة مالية والله يبارك لصاح	نمرة صفراء
سيارة غير قانونية بس ما شالله نضيفة بودي 200 السع 2300 السيارة للبيع بسبب ازمة مالية	فلسطينية
الفسبا ماشاء الله نضيفة عليها اضافات .. غير قانونية	فلسطينية
السيارهِ غير قانونية بس ولا خدش كرتونه مع فتحة سقف	نمرة صفراء
سيارة غير قانونية	فلسطينية
السيارة نمرة بحالة ممتازة غير قانونية ويوجر فيها ورقة شطب و يوجد الورقة الحمراء شطب	نمرة صفراء
شباب يسعد صباحكم موجود مرسيدس w202 ديزل 220 #قانوني و سيارة غير قانونية للتشقيف بسعر مغري الى معنى فيهم يتواصل عبر الـ	فلسطينية
السيارة غير قانونية	نمرة صفراء
السيارة بحالة ممتازة من جميع النواحي وهي للقطع لانها غير قانونية - السيارة كل شي فيها شغال وهي شطب رسمي -	نمرة صفراء
السيارهِ بحاله جيده جدا للبيع مع امكانيه التقسيط مع فارق بسعر ويوجد معاها سيارهِ غير قانونية قطع	فلسطينية
سيارة جديدة نقرة بتجنن السيارة غير قانونية السعر 5000 شيكل كاش من الاخر السعر رجاء عدم الاحراج	نمرة صفراء

Figure 4: noise data that affect price prediction

4.2.2 Color Column

since there's many set of gradations group for the same color, This step was used to consider the base color is the same color for all gradations to this color as follow ...

Figure 5: Final Color

4.2.3 previous Owners Column

since there's no restriction on input type for the previous owner's field on the website that we scrape data from it. There are some people was prefer to add the number on this filed such as (2) this mean that there's 2 person was buy this car, others people used Text input to represent this filed such as يد ثانية , the problem was happening with the last. Since there's a huge miss-spelling for words that must represent the same thing. So This step was used to identify one representation for each sets

There is 1700 words that represent previous owners we will put some here as follow ...

Figure 6: different representation for previous owner

4.2.4 Passenger Column

the problem in this section that there's two different way we can used to represent passenger filed.
for example
4+1 and 5 is represent the same thing (5 passenger)
6+1 and 7 is represent the same thing (7 passenger) and so on..

The second way(5) has no fault, the problem was happen on the first way(4+1) , since there's some people use other symbols for representation (4*1 , 4%1 , 4#1 , 4@1 , 4-1 , 4_1 ,4=1 , 4/1 , 4!1 , 4$1). So This step was used to determine one way to represent

we select second way to represent this column. because it will reduce the encoded data, since symbol was represent as a Text and need to encode

4.2.5 kilometers column

this step was taken a long time, a lot of mathematical problems was present on the values of this column (extra zeros, fewer zeros, Arabic number, decimal number, words, symbol, unit(km, mile,كم,...), Unreasonable numbers, ...), in this section we will explain carefully what we did and how we solved these problems.

Note1: we used Scripts code to solve this problem not Manuel Visit — > GitHub

Note2: see fig 7,8 to understand the noise in distance column

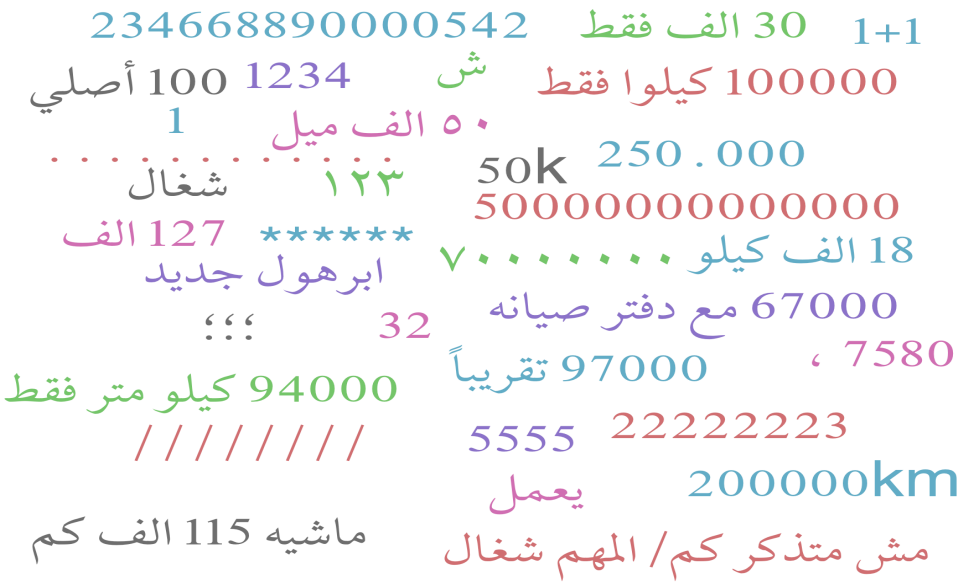


Figure 7: sample of noise data in Kilometers Column

كيلوا متر 120.000	1400	اوتوماتيك	فلسطينية	خصوصى	بنزين	أبيض	38000	2005	كيا ريو
مش متذكر كم/المه	1900	عادي	فلسطينية	عمومى	ديزل	أبيض عاجى	54000	2009	سكودا نيو
_____	1400	عادي	فلسطينية	خصوصى	بنزين	أبيض	65000	2015	اوبل كورسا
عدادات واطية	1600	اوتوماتيك	فلسطينية	خصوصى	بنزين	أبيض عاجى	68000	2013	هونداي اكسنت
103	1400	عادي	فلسطينية	خصوصى	بنزين	رمادي	46000	2008	اوبل كورسا
---------	1800	عادي	فلسطينية	خصوصى	بنزين	أبيض	26000	1999	اوبل فيكترا
-*	1250	عادي	فلسطينية	خصوصى	بنزين	أسود	43000	2009	فيات بونتو
10 تالاف كيلو	1400	نصف اوتوماتيك	فلسطينية	خصوصى	بنزين	أبيض	75000	2015	هونداي اكسنت
غير واضح	2000	عادي	فلسطينية	خصوصى	ديزل	أبيض	4500	2002	اوبل استرا
0كيلو	1600	عادي	فلسطينية	خصوصى	ديزل	بنى	85000	2015	بيجو 301
1111111111111	1600	نصف اوتوماتيك	فلسطينية	خصوصى	بنزين	أبيض	70000	2011	كيا سول
..ء	2400	عادي	فلسطينية	خصوصى	ديزل	أبيض عاجى	30000	1994	فولكسفاجن ترانسبورتر
103 شغال	1995	اوتوماتيك	فلسطينية	خصوصى	ديزل	فضى	92000	2011	هونداي توسان
متئن الف فى المانيا	2000	اوتوماتيك	فلسطينية	خصوصى	ديزل	أسود	10000	2013	سكودا اوكتافيا
عدادات واطية	1200	عادي	فلسطينية	خصوصى	بنزين	عدة الوان	67000	2014	فولكسفاجن بولو
1.21212E+12	1600	اوتوماتيك	فلسطينية	خصوصى	بنزين	فضى	54000	2002	فولكسفاجن جولف
----------	2500	عادي	فلسطينية	خصوصى	تجاري	أبيض عاجى	56000	2001	شاحنة مان
_____	2200	اوتوماتيك	فلسطينية	خصوصى	ديزل	أسود ميتالك	75000	2008	هونداي سانتافيه
&&&&&&&&&&	125	اوتوماتيك	فلسطينية	خصوصى	بنزين	أحمر	8500	2013	دراجة نارية أصل برايفت
غير معروف	1900	عادي	فلسطينية	عمومى	ديزل	أبيض	96644	2008	سيت قرطبة
٩١٣٤٥٦٧٨٩٨٦٥٤	3000	عادي	فلسطينية	عمومى	ديزل	أبيض	54000	1999	مرسيدس سبرنتر
0كيلو	2500	عادي	فلسطينية	خصوصى	ديزل	أسود ميتالك	25000	2016	ميتسوبيشى L200
يعمل بدون مشاكل	1600	عادي	فلسطينية	خصوصى	بنزين	رمادي	19500	1993	اوبل فيكترا
10000000000000	1800	اوتوماتيك	فلسطينية	خصوصى	بنزين	رمادي	35000	1999	اودي A4
من 200وانزل	1600	عادي	فلسطينية	خصوصى	ديزل	أسود	90000	2014	سكودا اوكتافيا
00000000000000	1400	عادي	فلسطينية	خصوصى	بنزين	أبيض	30000	1998	فولكسفاجن بولو
كيلو 114.....	5985	عادي	نمرة صفراء	تجاري	ديزل	أبيض	185000	2013	مرسيدس 519 دبل كابينا
!!!!!!!!!!!!	2500	عادي	فلسطينية	خصوصى	ديزل	أبيض	70000	1999	ميتسوبيشى ماجنوم
🛢🛢🛢	1600	اوتوماتيك	فلسطينية	خصوصى	بنزين	زيتى	35000	2007	شفرليت اوبترا
1015 الف كيلو	1500	اوتوماتيك	فلسطينية	خصوصى	بنزين	ذهبى	40000	2004	هونداي اكسنت
ميتين وخمسين	1600	عادي	فلسطينية	تجاري	بنزين	أبيض	25362	2000	هوندا CR-V
++++++	2500	عادي	فلسطينية	عمومى	ديزل	بنى	40000	2007	هونداي H1
ميتين وخمسين	1600	عادي	فلسطينية	خصوصى	بنزين	أبيض	21000	2000	هونداي اكسنت
102000	1400	اوتوماتيك	فلسطينية	خصوصى	بنزين	رمادي	37000	2006	هونداي كليك
00008876غفف	1600	عادي	فلسطينية	عمومى	ديزل	أبيض	50000	2011	كيا سيد
,,,,,,,,,	1300	عادي	فلسطينية	خصوصى	بنزين	أبيض عاجى	14000	1984	اوبل اسكونا
1111111كيلو متر	1900	عادي	فلسطينية	عمومى	ديزل	خمري	33000	2001	سكودا اوكتافيا
فقط 800 كم أقل م	1400	اوتوماتيك	فلسطينية	خصوصى	بنزين	أبيض	72000	2017	هونداي اكسنت
102000	1600	عادي	فلسطينية	خصوصى	ديزل	أسود	115000	2014	فولكسفاجن كادي
!!!!!!!!!!!!	2500	عادي	فلسطينية	خصوصى	ديزل	أبيض	68000	1999	ميتسوبيشى ماجنوم
102000	2000	اوتوماتيك	فلسطينية	خصوصى	ديزل	أبيض عاجى	116000	2014	هونداي سانتافيه
يعنى معقول	903	عادي	فلسطينية	خصوصى	بنزين	بنى	5300	1983	فيات 127
ماشيه 113 الف كم	1500	عادي	فلسطينية	خصوصى	بنزين	أبيض	26000	1999	دايو لانوس
//////////	480	عادي	فلسطينية	تجاري	ديزل	أصفر	145000	2004	داف ٢٠٠٤
١٠٢٠٠٠	1600	اوتوماتيك	فلسطينية	خصوصى	بنزين	سكنى	50000	2008	كيا سيراتو
102000	2000	اوتوماتيك	فلسطينية	خصوصى	ديزل	أسود	57000	2006	هونداي توسان
عدادات واطية	2000	اوتوماتيك	فلسطينية	خصوصى	ديزل	فضى	98000	2015	هونداي توسان
ماشيه 113 الف كم	1500	عادي	فلسطينية	خصوصى	بنزين	أبيض	26000	1999	دايو لانوس
103 قابل الزيادة	1600	اوتوماتيك	فلسطينية	خصوصى	ديزل	أزرق	107000	2015	سكودا اوكتافيا
١٠٠٥٢ الف كيلو	1900	عادي	فلسطينية	خصوصى	ديزل	فيرانى	95000	2010	فولكسفاجن توران
100000الف وشوي	1600	اوتوماتيك	فلسطينية	خصوصى	بنزين	كرميدي	40000	2007	شفرليت اوبترا
ماشيه 113 الف كم	1500	عادي	فلسطينية	خصوصى	بنزين	أبيض	25500	1999	دايو لانوس
ماشيه 100 الف كيل	1400	عادي	فلسطينية	خصوصى	بنزين	أسود	47500	2010	سيت ايزا
يعمل بدون مشاكل	2200	اوتوماتيك	فلسطينية	خصوصى	بنزين	أبيض	35500	1996	مرسيدس 220
102000	2000	اوتوماتيك	فلسطينية	خصوصى	ديزل	بنى	82000	2011	كيا سبورتاج
1019الف كيلو متر	1400	اوتوماتيك	فلسطينية	خصوصى	بنزين	فضى	40000	2006	كيا ريو
١٠٢٠٠٠	1600	عادي	فلسطينية	خصوصى	ديزل	سكنى	90000	2014	فولكسفاجن كادي
١٠٢ الف	1600	عادي	فلسطينية	خصوصى	ديزل	ذهبى	90000	2016	بيجو بارتنر
تحت 50 الف	1200	عادي	فلسطينية	خصوصى	بنزين	عدة الوان	64999	2017	سيت ايزا
0كيلو	1600	عادي	فلسطينية	خصوصى	ديزل	عدة الوان	115000	2017	بيجو بارتنر
مش واصل 80 الف	1500	اوتوماتيك	فلسطينية	خصوصى	بنزين	أبيض	40000	2005	هونداي فيرنا

Figure 8: sample of noise data on distance column before cleaning(2)

- **handle general noise in KM column**

  This step was deal with the noise that notice in this column such as (symbol, Unreasonable numbers, large number, wrong sentence) and how to handle it as follow:

  1. there's some value contain symbol such as (6@0000 , 76?000 , 120!234 , 45/300,...) so we remove symbol as follow (60000 ,76000 ,120234 ,45300).

  2. there's some Unreasonable symbol such as (********,!@,%@$*) so we remove it.

  3. there's some Unreasonable number we notice such as (00000000 ,11111111111 ,0123456789 , 0932187813) so we remove it.

  4. there's some wrong sentence that doesn't represent any thing such as (ما بعرف كم ماشية, مطور جديد , مش ماشية كثير) so we remove it.

- **handle noise in numbers within KM column**

  There's a lot of person used word to represent the value of the Kilometers that car runs such as(١٠٠ الف , ١٠٠ الف , اقل من ١٠٠ الف , اكبر من ١٠٠ الف ٧٠٠٠٠ ميل , مصفر, ٥٠ الف كيلو متر , صفر كيلو , بين ال ٥٠ الف و ٦٠ الف, etc...) so in this step we need to convert all word to number

  1. convert the word الف to (000) if the length of number less than three digit such as ١٠٠ الف it become ١٠٠٠٠٠ otherwise we will extract number and remove word الف such as ١٠٠٠٠٠ الف it become ١٠٠٠٠٠

  2. deal with range number like ,بين ال ٥٠ الف و ٦٠ الف and consider one value 55000

  3. extract the number contain Unit like (٥٠٠٠٠ كم,٤٠٠٠٠ ميل,120000km)

  4. convert Arabic number ٧٦٠٠٠ to English number 76000

4.2.6 Car name column

This step was take a very long time since there's a lot of misspelling words for naming the vehicle. In addition, there are some people who use the nickname to represent the vehicle, so the process was done in 3 steps as follow:

- **Remove Duplicate:** we create a separate copy from the data and remove all duplicate rows in the vehicle name column, so we ended up with 4500 vehicle names
- **split car name** since there's a different nicknames and spelling that represent the same car as show in Fig(9) we collect each sets in separate file that must represent the same car with one name then we decide to split the car name into two different columns

  make: represent the company name.
  model: represent the derived name.
  Then to increase the Accuracy for description we suggest to add new column that deal with car body style called Type see Fig(10) to notice the different Type.

  Note1:there's some car that have multiple type(body style) such as opel astra , so we decide to use (sedan/hatch) to represent this car and similar car

  Note2:while splitting the car we noticed that there's some car has a different number of passengers while there's a unique number of passenger for this car
  example: Kia forte is only a sedan car so the number of passengers must be (5) so we set all Kia forte to be (5) passengers

- **modify Original file:** in this step, we Replace the Vehicle name in the Original file with the three columns that modify in step 2, see Fig(11) to see the Final result.

Figure 9: sample of noise data on car name column

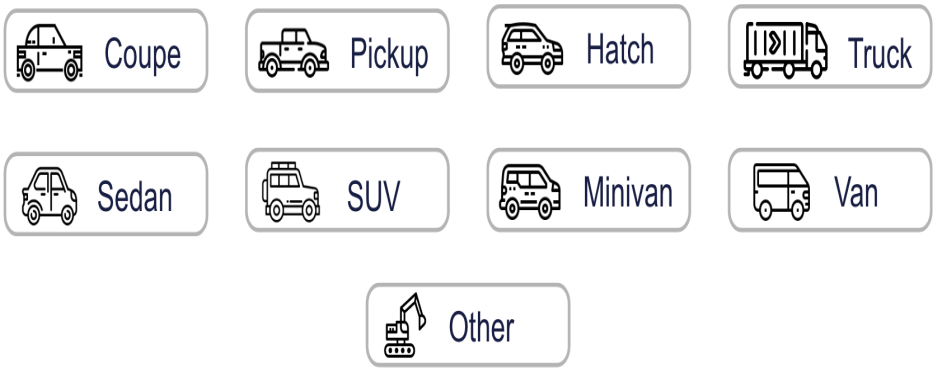


Figure 10: Car body style that used in data

فولكسفاجن LT	1+8
هونداي اكسنت	1400
هونداي H1	6+1
كيا سورينتو	6+1
فيات فرينو	6+1
فولكسفاجن ترانسبورتر	١+2
فولكسفاجن بورا	4+7
سيت ابيزا	5+1
هونداي توسان	20000
فورد ترانزيت	0
فيات 500	1
باص lt 46	16+1
بي ام دبليو 525	5+1
فولكسفاجن كرافيل	8=2
هونداي تراجيت	9
لاندروفر ديفندر	7 + 1
فولكسفاجن كرافيل	ترخيص 8 ركاب
فولكسفاجن GTI	4+1
ميتسوبيشي باجيرو	6x1
شفرليت اوبترا	5+1
ستروين جامبي	8+1
بي ام دبليو 335 sport	3
تويوتا لاند كروزر	7+1
فورد كارجو	2+1
شفرليت سلفرادو	2+1
Beige307 و	6+1

Volkswagen	LT	van	15
hyundai	Accent	sedan	5
hyundai	H1	van	8
kia	Sorento	SUV	7
Fiat	Fiorino	minivan	5
Volkswagen	Transporter	van	8
Volkswagen	bora	__sedan/hatch__	5
SEAT	Ibiza	hatch	5
hyundai	tucson	SUV	5
Ford	transit	van	8
Fiat	500	hatch	5
Volkswagen	LT	van	17
BMW	5 Series	sedan	5
Volkswagen	Caravelle	van	5
hyundai	Trajet	minivan	7
LandRover	Defender	SUV	7
Volkswagen	Caravelle	van	5
Volkswagen	Golf-gti	hatch	5
mitsubishi	pajero	SUV	7
Chevrolet	optra	sedan	5
Citroen	Jumpy	van	8
BMW	3 Series	coupe	4
Toyota	Land Cruiser Pra	SUV	7
Ford	cargo	truck	2
Chevrolet	silverado	pickup	5
Peugeot	307	__sedan/hatch__	5

Figure 11: Final result for car name column

4.2.7 additional features

The website we scrape the data from it provide a user to select an additional feature from a group of Text checkbox, as a step to reduce the non-numeric value in data, we generate a code that replace a list of text into a Boolean num-

ber so (1) will represent that vehicle contain this feature, (0) vehicle not contain this feature as follow ...

accessories	alarm	ac	radio_cd	sunroof	eather.chair	central.Lock	nesium_wh	air_bag
مُكيّف,إغلاق مركزي,جهاز إنذار,فتحة سقف	1	1	0	1	0	1	0	0
مُكيّف,إغلاق مركزي,جهاز إنذار,مسجل CD,وسادة حماية هوائية	1	1	1	0	0	1	0	1
مُكيّف,إغلاق مركزي,جهاز إنذار,مسجل CD,جنطات مغنيسيوم,فرش جلد,وسادة حماية هوائية	1	1	1	0	1	1	1	1
مُكيّف,إغلاق مركزي,جهاز إنذار,مسجل CD,جنطات مغنيسيوم,وسادة حماية هوائية	1	1	1	0	0	1	1	1
إغلاق مركزي,جهاز إنذار,مسجل CD,فتحة سقف,جنطات مغنيسيوم,فرش جلد	1	0	1	1	1	1	1	0
مُكيّف,إغلاق مركزي,جهاز إنذار,مسجل CD,جنطات مغنيسيوم,وسادة حماية هوائية	1	1	1	0	0	1	1	1
مُكيّف,إغلاق مركزي,جهاز إنذار,مسجل CD,جنطات مغنيسيوم,وسادة حماية هوائية	1	1	1	0	0	1	1	1
مُكيّف,إغلاق مركزي,جهاز إنذار,مسجل CD,فتحة سقف,جنطات مغنيسيوم,فرش جلد,وسادة حماية هوائية	1	1	1	1	1	1	1	1
فرش جلد	0	0	0	0	1	0	0	0
مُكيّف,إغلاق مركزي,فتحة سقف,جنطات مغنيسيوم,وسادة حماية هوائية	0	1	0	1	0	1	1	1
مُكيّف,جهاز إنذار,مسجل CD,وسادة حماية هوائية	1	1	1	0	0	0	0	1
مُكيّف,إغلاق مركزي,جهاز إنذار,مسجل CD,وسادة حماية هوائية	1	1	1	0	0	1	0	1
إغلاق مركزي,جهاز إنذار,مسجل CD,جنطات مغنيسيوم,فرش جلد,وسادة حماية هوائية	1	0	1	0	1	1	1	1
مُكيّف,إغلاق مركزي,جهاز إنذار,مسجل CD,فرش جلد	1	1	1	0	1	1	0	0
مُكيّف,إغلاق مركزي,جهاز إنذار,مسجل CD,جنطات مغنيسيوم,فرش جلد,وسادة حماية هوائية	1	1	1	0	1	1	1	1
مُكيّف,إغلاق مركزي,جهاز إنذار,مسجل CD,جنطات مغنيسيوم,فرش جلد,وسادة حماية هوائية	0	1	1	0	1	1	1	1
مُكيّف,إغلاق مركزي,مسجل CD,جنطات مغنيسيوم,وسادة حماية هوائية	0	1	1	0	0	1	1	1
مُكيّف,إغلاق مركزي,جهاز إنذار,مسجل CD,جنطات مغنيسيوم,وسادة حماية هوائية	1	1	1	0	0	1	1	1
مُكيّف,إغلاق مركزي,جهاز إنذار,مسجل CD,فتحة سقف,جنطات مغنيسيوم,فرش جلد,وسادة حماية هوائية	1	1	1	1	1	1	1	1
إغلاق مركزي,مسجل CD,فتحة سقف,جنطات مغنيسيوم	0	0	1	1	0	1	1	0
مُكيّف,إغلاق مركزي,جهاز إنذار,فتحة سقف,جنطات مغنيسيوم,فرش جلد,وسادة حماية هوائية	1	1	0	1	1	1	1	1

Figure 12: accessories before and after change

4.2.8 Day-life column

As a step to improve our project, we suggest to improve our data by adding a new feature that tells the age of this vehicle, this feature is a combination of two features (year and published ad) so we consider the year of vehicle production to be the date of vehicle registration and ad publication as the age of the vehicle **now** ..

Example:
year of vehicle production: 2018
ad publication: 27/5/2021
Day-life(new feature): (1/1/2018) - (27/5/2021) = 1242 day

According to the previous information we calculate this feature and notice that it has a very high correlation with the target (price)

5 Machine learning part

After finish all Data cleaning process that took a long time, we start to apply different ML Models to Test the Data and see its behavior, so according to the results, we find the best Model that get the best Accuracy and less MAE to be used..

Note1: The next sections will explain exactly the process step-by-step
Note2: you can see the fully code for both ML Models and Deep Neural network by Visit our Repository on this link —-> GitHub

5.1 Load and Understand The Data

As any Machine learning project before applying any Technique we have to load the data to understand it and gain the knowledge from it ..
The next Figs will show the Result
Fig(13) will show the final shape of data
Fig(14) will show some information about data

	make	model	type	passenger	year	price	color	fuel	history	gear	...	url	alarm	ac	radio_cd	sunroof	leather-chair	central-Lock	magnesium_wheels	air_bag	Days_Live
0	Volkswagen	Passat	sedan	5	1999	28500	dark blue	petrol	private	automatic	...	https://shobiddak.com/cars/432099	1	1	0	1	0	1	0	0	6472
1	Volkswagen	Passat	sedan	5	2001	43000	gray	petrol	private	automatic	...	https://shobiddak.com/cars/432100	1	1	1	0	0	1	0	1	5741
2	hyundai	Accent	sedan	5	2014	68000	white	petrol	private	automatic	...	https://shobiddak.com/cars/432104	1	1	1	0	1	1	1	1	993
3	kia	Pride(rio)	_sedan/hatch_	5	2008	49000	silver	diesel	private	manual	...	https://shobiddak.com/cars/432105	1	1	1	0	0	1	1	1	3185
4	BMW	3 Series	sedan	5	1986	30000	blue	petrol	private	manual	...	https://shobiddak.com/cars/432107	1	0	1	1	1	1	1	0	11219
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
200460	Mercedes	Vario	truck	7	1996	70000	white	diesel	Commercial	manual	...	https://shobiddak.com/cars/517019	0	1	1	0	1	1	0	0	7985
200461	Mercedes	C-Class	sedan	5	2002	50000	brown	diesel	taxi	automatic	...	https://shobiddak.com/cars/679985	0	1	1	0	1	1	1	1	7184
200462	Mercedes	Sprinter	van	7	1989	25000	white	diesel	private	manual	...	https://shobiddak.com/cars/570091	0	0	0	0	0	0	0	0	10860
200463	Mercedes	Atego	truck	3	1996	85000	white	diesel	Commercial	manual	...	https://shobiddak.com/cars/463031	1	1	1	1	0	1	0	0	7853
200464	Mercedes	Sprinter	van	7	2008	80000	white	diesel	Commercial	manual	...	https://shobiddak.com/cars/481989	1	1	1	0	0	1	0	1	3420

Figure 13: final content of data after cleaning

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200465 entries, 0 to 200464
Data columns (total 26 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   make                200465 non-null  object
 1   model               200465 non-null  object
 2   type                200465 non-null  object
 3   passenger           200465 non-null  int64
 4   year                200465 non-null  int64
 5   price               200465 non-null  int64
 6   color               200465 non-null  object
 7   fuel                200465 non-null  object
 8   history             200465 non-null  object
 9   gear                200465 non-null  object
 10  engine_size         200465 non-null  object
 11  Kilometers          117422 non-null  float64
 12  owners              149417 non-null  float64
 13  ad_date             200465 non-null  object
 14  payment_method      200465 non-null  object
 15  windows             200465 non-null  object
 16  url                 200465 non-null  object
 17  alarm               200465 non-null  int64
 18  ac                  200465 non-null  int64
 19  radio_cd            200465 non-null  int64
 20  sunroof             200465 non-null  int64
 21  leather_chair       200465 non-null  int64
 22  central_Lock        200465 non-null  int64
 23  magnesium_wheels    200465 non-null  int64
 24  air_bag             200465 non-null  int64
 25  Days_Live           200465 non-null  int64
dtypes: float64(2), int64(12), object(12)
memory usage: 39.8+ MB
```

Figure 14: some information about the loaded data

from the previous block we Notice that:

- The total Number of Data is 200465 and the total number of Features is 26
- The Data is contain both Categorical and numeric value
- There's Some Missing Data in Some Column (Kilometers,owners)
- There's Some Dtype for some column(engine-size,owners,km) needed to change

  So based in This Information we will Do some Prepossessing on Data To be Ready as follow ...

  - *Delete Some unnecessary Column(URL,ad date publish)*
  - *encode the Categorical Data*
  - *Convert type for (engine-size) to the correct type*
  - *impute Numeric Missing Data in (owners,Kilometers)*

5.2 encode the Categorical Data

Machine learning models require all input and output variables to be numeric. but our data contain a non-numeric in (9) Columns , so we must encode them before we can fit and evaluate the model.

There's 2 common Type of encoding:

- Ordinal encoding
- One-hot-encoding **used**

Fig(15) will show the shape for Data after get encoded
Fig(16) will show the content of data after encoded

```
The shape for Data before encode:  (200465, 26)
The shape for Data after encode:  (200479, 688)
```

Figure 15: shape for Data after get encoded

engine_size	Kilometers	owners	alarm	ac	radio_cd	sunroof	leather-chair	...	private	rental	school	taxi	automatic	manual	cash	installment	electric
1600	NaN	NaN	1	1	0	1	0	...	1	0	0	0	1	0	1	0	1
1600	100000.0	2.0	1	1	1	0	0	...	1	0	0	0	1	0	1	0	1
1500	45000.0	0.0	1	1	1	0	1	...	1	0	0	0	1	0	1	0	0
1500	NaN	NaN	1	1	1	0	0	...	1	0	0	0	0	1	1	0	1
1600	NaN	10.0	1	0	1	1	1	...	1	0	0	0	0	1	1	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	NaN	4.0	0	1	1	0	1	...	0	0	0	0	0	1	1	0	0
2200	NaN	NaN	0	1	1	0	1	...	0	0	0	1	1	0	0	1	1
3000	NaN	1.0	0	0	0	0	0	...	1	0	0	0	0	1	1	0	0
170	NaN	4.0	1	1	1	1	0	...	0	0	0	0	0	1	1	0	0
2150	300000.0	NaN	1	1	1	0	0	...	0	0	0	0	0	1	0	1	1

Figure 16: content of data after get encoded

As you see there's Some (NaN) Value on data ,so the next step will impute this Missing data

5.3 impute the Missing Data

since there's some Missing Data in (owners,Kilometers) column , our Target in this section is to fill the Missing with Different Techniques and determine the best Technique that we can used as follow ...

Fig(17) will show the result between all Technique that used

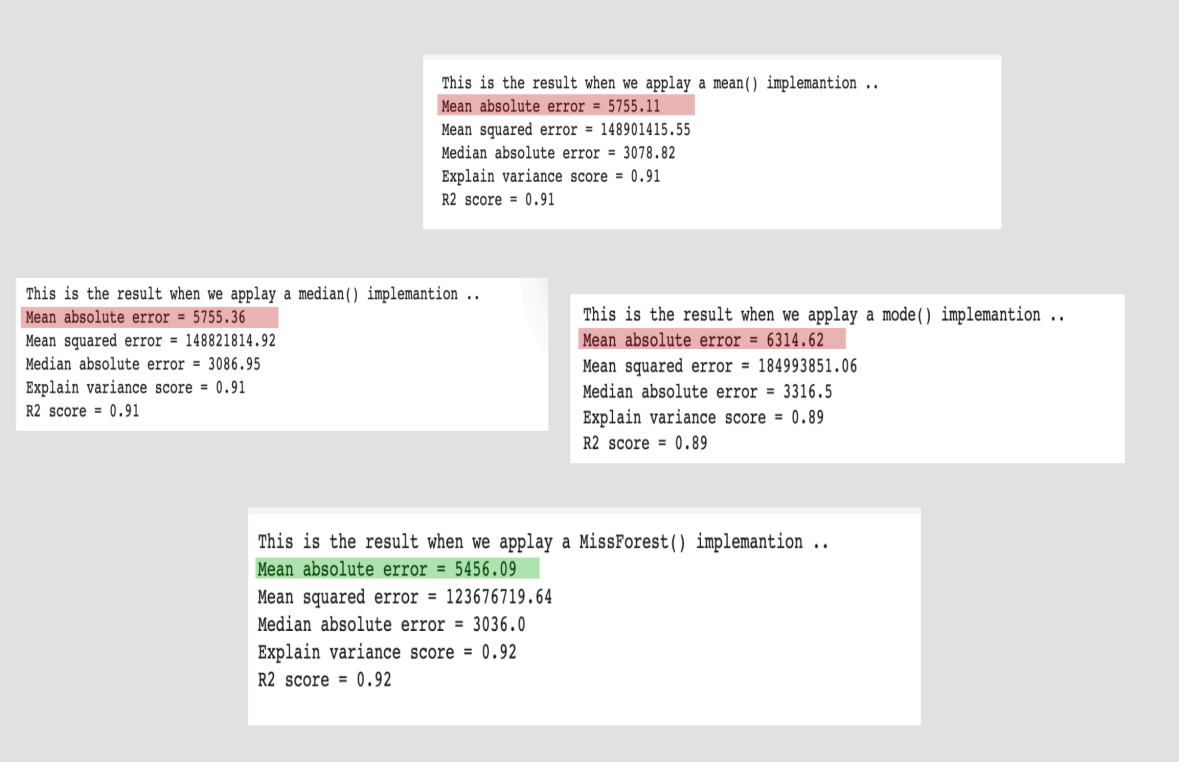


Figure 17: result between imputation Technique that used

As you see that the Miss-Forest Technique has gain the best Evaluation , so the next pieces line will describe what Miss-Forest and how to use it ..

Definition:"Miss-forest is a non parametric imputation method for basically any kind of data. It can cope with mixed-type of variables, nonlinear relations, complex interactions and high dimensionality"

Steps:

- Firstly, we must apply Encoded Technique in Categorical Data. **Done previous**
- Secondly,we Must split the Target value from the Data to avoid Data Leakage problem
- Finally, we perform the imputation in all Data. Then we create a new columns to use this imputation to fill Missing Data on it.

Fig(18) will show the imputation process
Fig(19) will show the result for missing data before and after

```
%%time
# Make an instance and perform the imputation
imputer = MissForest()
X_imputed = imputer.fit_transform(MyData)
```

```
Iteration: 0
Iteration: 1
Iteration: 2
Iteration: 3
Iteration: 4
CPU times: user 6h 9min 20s, sys: 1min 43s, total: 6h 11min 3s
Wall time: 33min 40s
```

```
# Add imputed values as columns to the untouched dataset
MyData['MF_kilometers'] = X_imputed[:, 3]
MyData['MF_owners'] = X_imputed[:, 4]
```

Figure 18: imputation process

	Kilometers	MF_kilometers	owners	MF_owners
0	NaN	131187.56	NaN	2.78
1	100000.0	100000.00	2.0	2.00
2	45000.0	45000.00	0.0	0.00
3	NaN	136610.36	NaN	2.68
4	NaN	126869.61	10.0	10.00
...	...	...	...	...
200460	NaN	257603.06	4.0	4.00
200461	NaN	273561.17	NaN	4.25
200462	NaN	102848.80	1.0	1.00
200463	NaN	492897.78	4.0	4.00
200464	300000.0	300000.00	NaN	1.69

[200465 rows x 4 columns]

Figure 19: The result for missing data before and after

5.4 Data Graphical Representation

In This step we do some visualizing on both categorical and numeric data , then we look for the correlation with Target(price) as follow ...

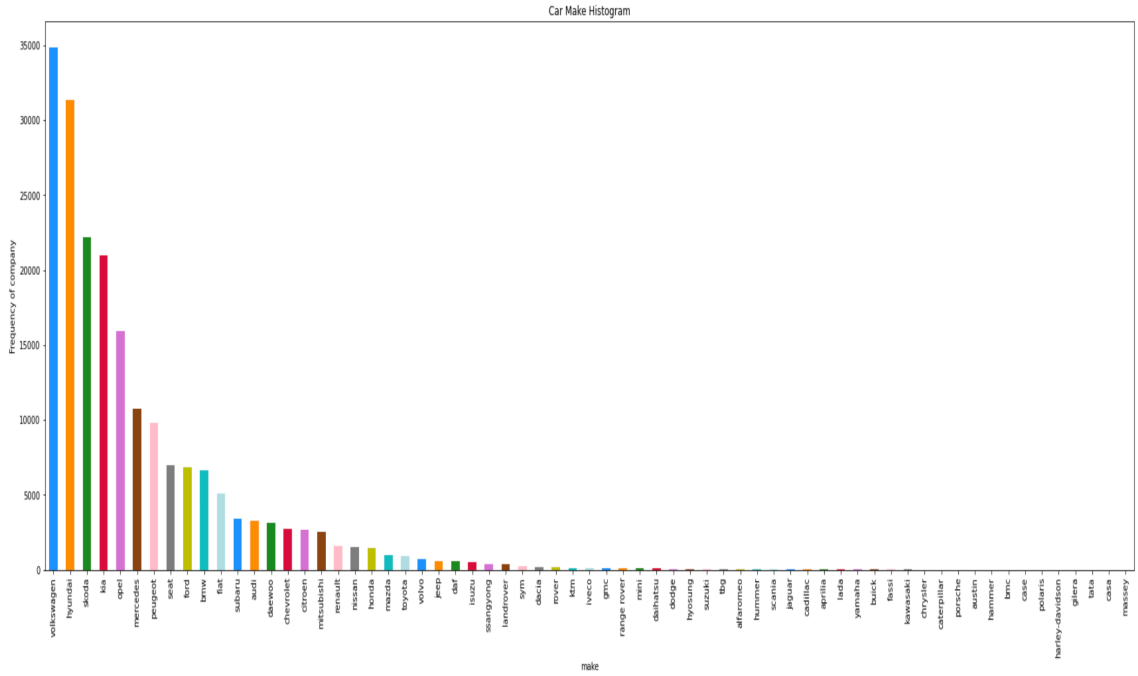


Figure 20: car-make column visualizing

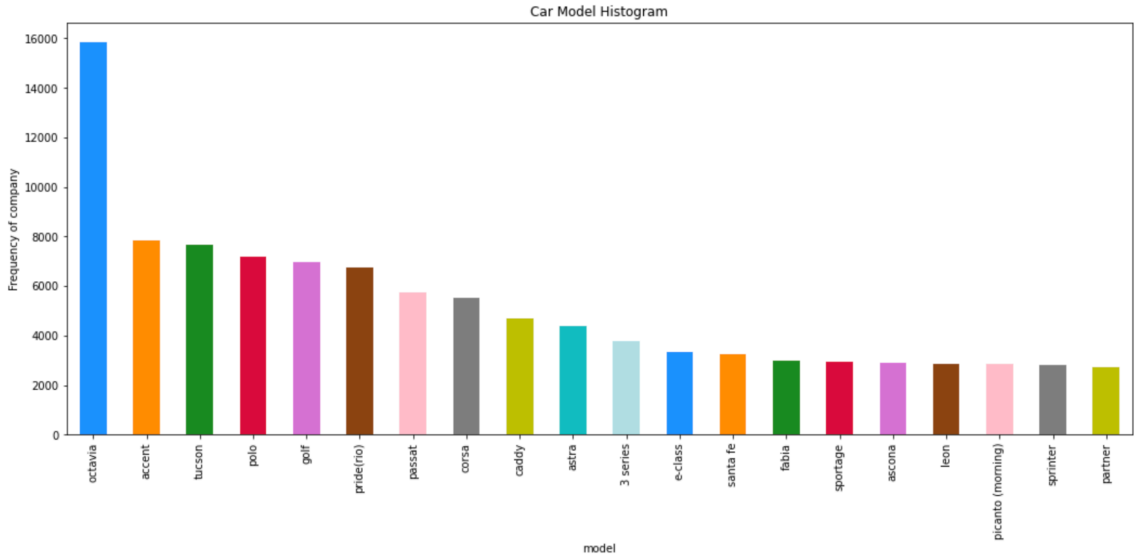


Figure 21: car-model column visualizing

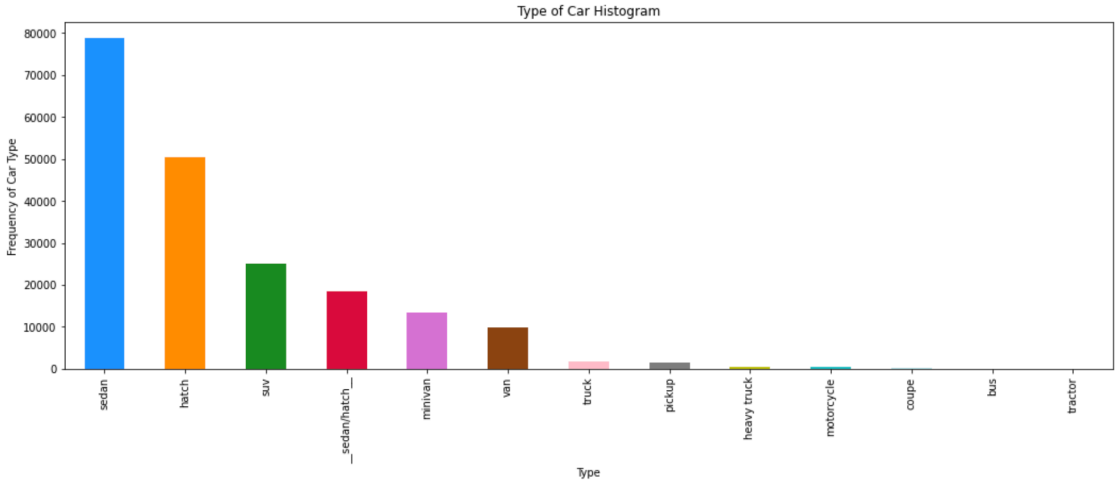


Figure 22: car-type column visualizing

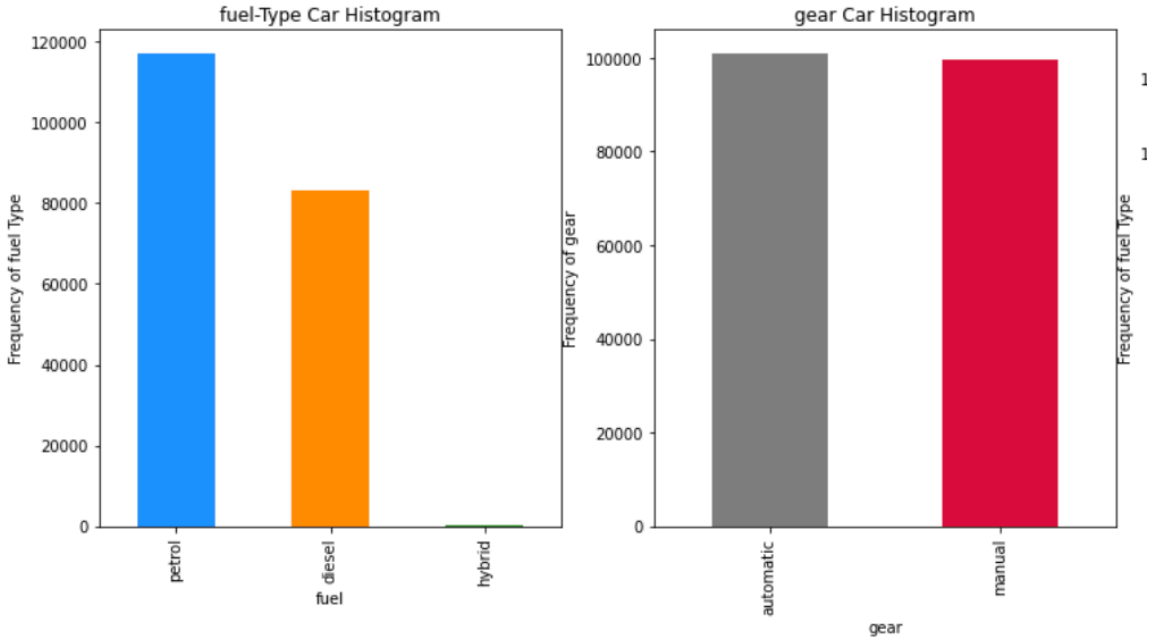


Figure 23: car-fuel&gear columns visualizing

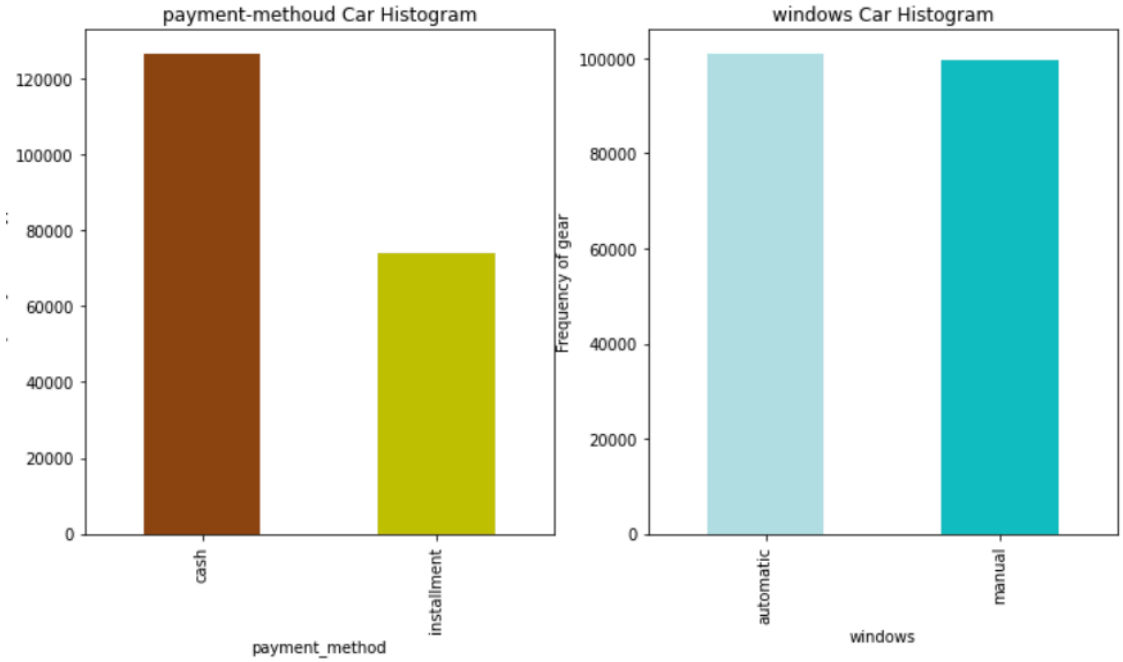


Figure 24: car-payment-method&windows columns visualizing

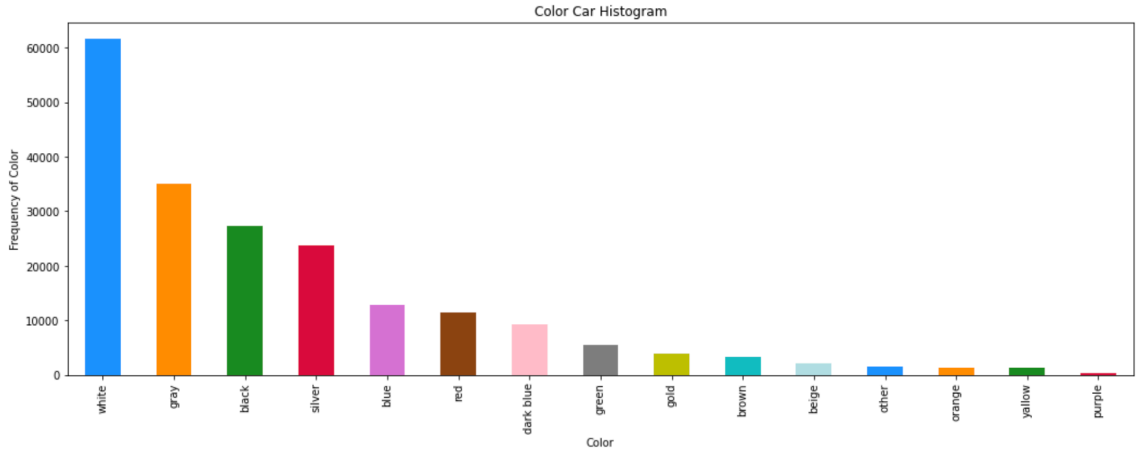


Figure 25: car-color column visualizing

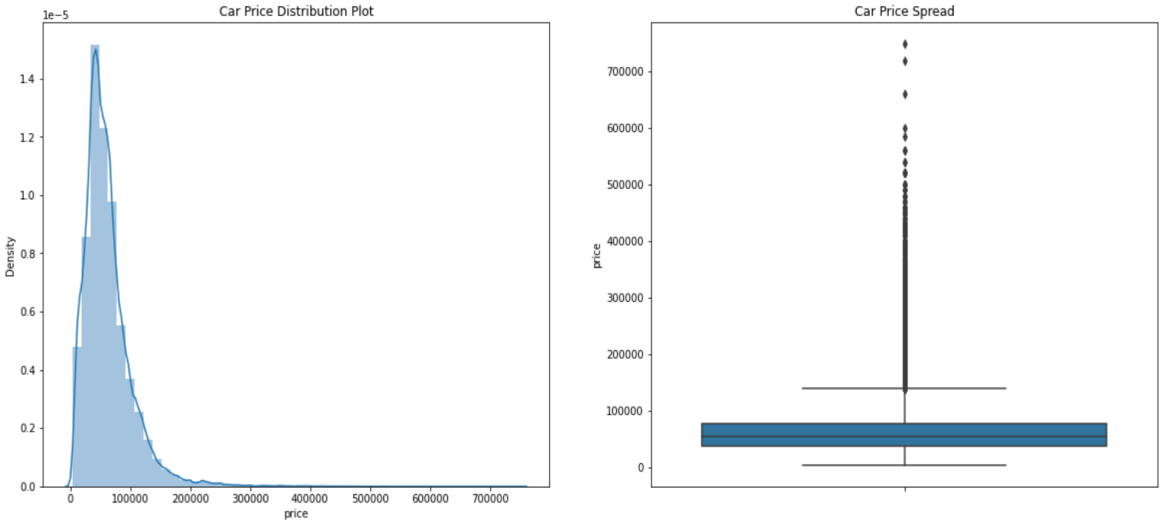


Figure 26: car-price column visualizing

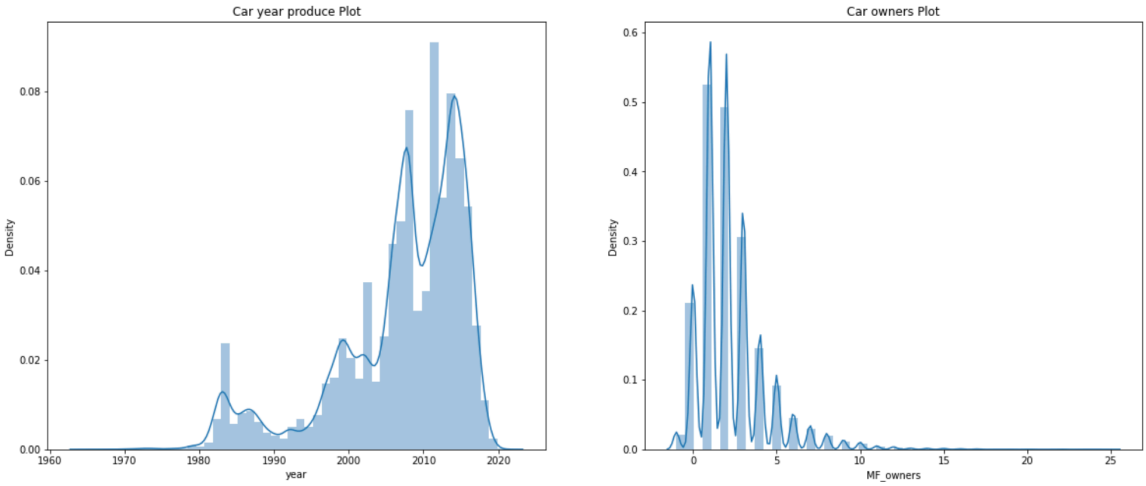


Figure 27: car-year Produce&owners columns visualizing

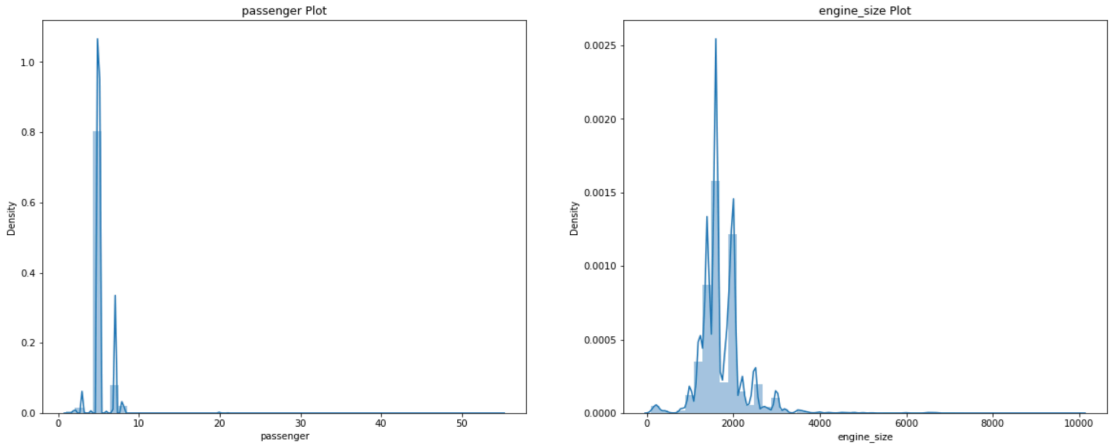


Figure 28: car-passenger&engine size columns visualizing

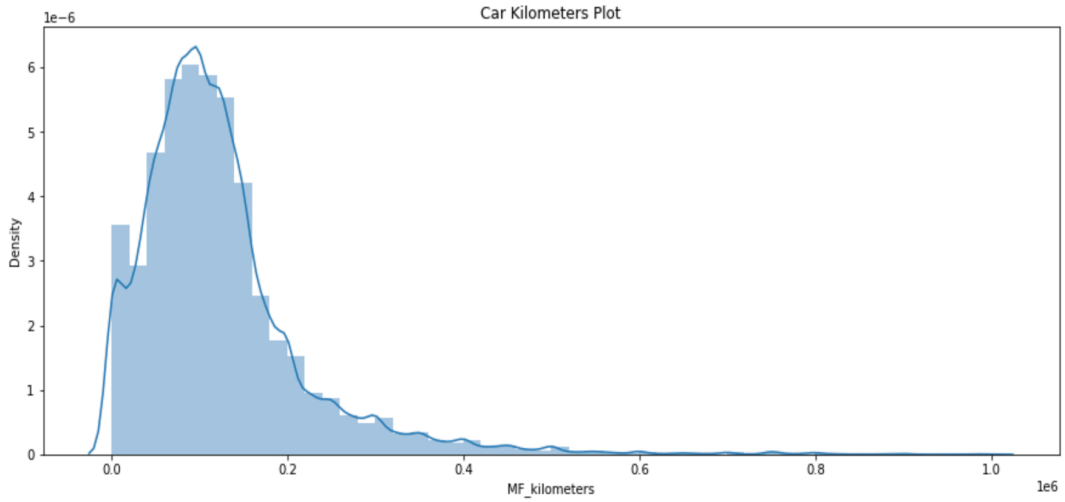


Figure 29: car kilometers column visualizing

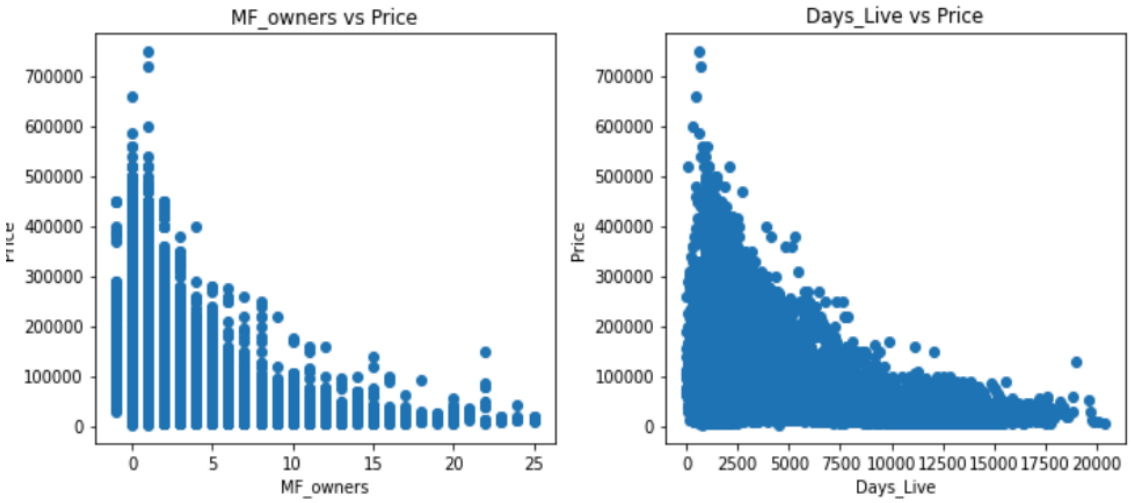


Figure 30: Correlation between price and MF-owner,Days-live

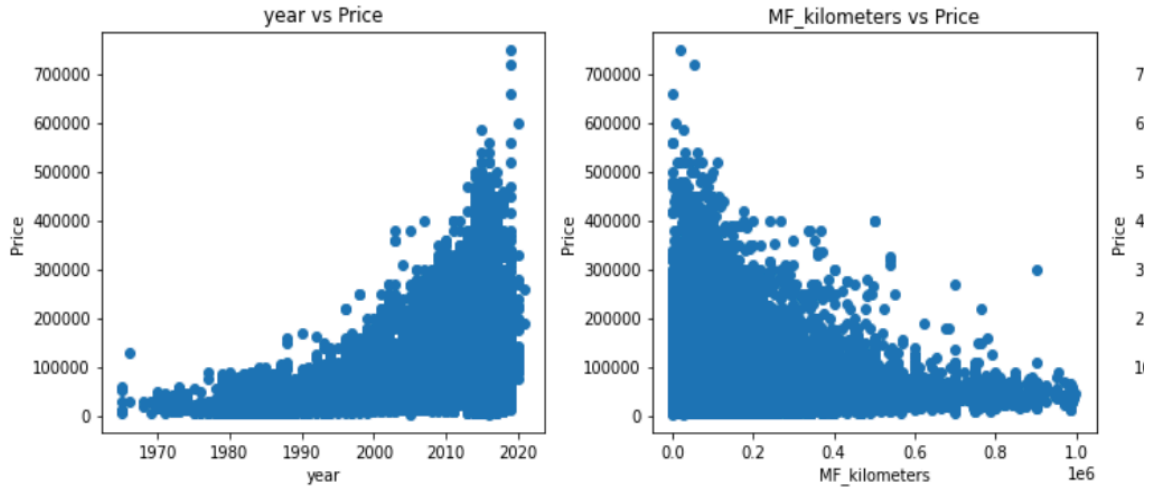


Figure 31: Correlation between price and MF-kilometers,year

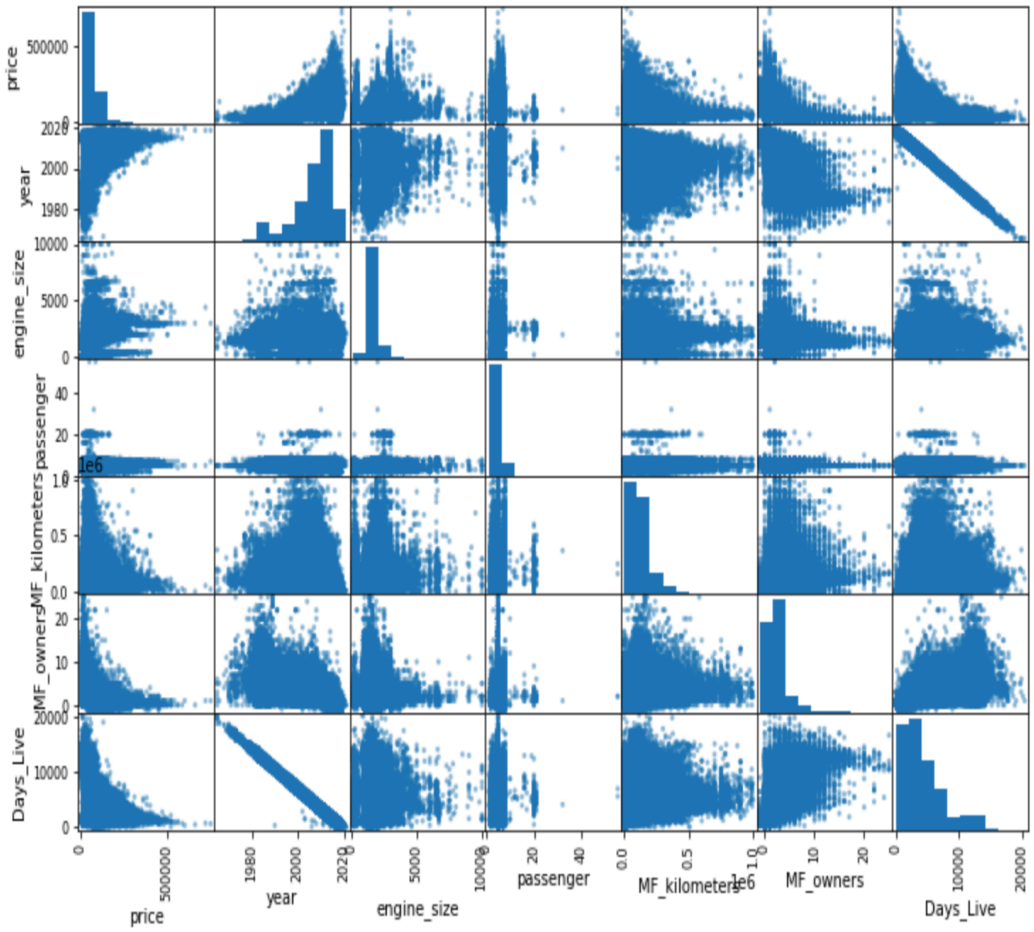


Figure 32: Correlation between all features

Important Notes

- **from the previous Categorical plots we notice that:**
  - The Most company car that exist in Data is **Volkswagen**
  - The Most model car that exist in Data is **Octavia**
  - The Most car Type that exist in Data is **Sedan**
  - The Most fuel Type that exist in Data is **Petrol**
  - The Most car Type that exist in Data is **Sedan**
  - The Most car Color that exist in Data is **white**
- **from the previous numeric plots we notice that:**
  - The price plot seemed That the most prices in Data are low between **(20,000 and 100,000)** with some car above and below
  - The year plot seemed That the most car in Data are low between **(2005 and 2017)** with some car above and below
  - The owners plot seemed That the most previous owners for the car in Data is about **2**
  - The passenger plot seemed That the most capacity passenger for the car in Data is about **5**
  - The engine-size plot seemed That the most engine-size for the car in Data are low between**(1000 and 1700)**

    Note: the y-axis is standard scientific notion, so if there's a 0.2 on the y-axis and a 1e6 at the top, the value for 0.2 actually indicates 0.2*1e6 = 0.2 * 10^6 = 200,000.

  - from the previous Note , The Kilometers plot seemed That the most car in Data are run about **100,000 Km**

5.5 Normalization

Definition"Normalization is consider a part of preparation that done in data. The goal of normalization is to change the values of numeric columns in the data-set to become in a range of[0,1], without distorting differences in the ranges of values."

since There's Some Machine learning Model insensitive for Normalization and there's Sensitive so we will firstly split the data into 2 part (train-set(80%) , test-set(20%)) , then we create a new copy from (train,test) and apply Normalization

on it , so the final result will have 4 copy's (train,test) without Normalization and copy of this sets but we apply Normalization on it

Note: since the data is not normally distributed we doesn't use Standarization

5.6 Models implementation

in This section we will explain all the Models that we used and which model is get best evaluation as follow ...

- **Support Vector Regression(LinearSVR):**
  "Support Vector Regression its a powerful model that capable of performing linear and nonlinear Regression Problem and can perform outlier detection , we will try to applay this Techniqe on our Data but because the Data is very large and complex there's another class Similar to SVR with parameter kernel='linear', but implemented in terms of liblinear rather than libsvm that SVR implemanted so its more flexiable in the choice of penalties and loss functions and should scale better to large numbers of samples."
  Note: LinearSVR is sensitive for normalization

  Fig(34) show the grid search result
  Fig(35) show the Evaluation for LinearSVR Model

```
%%time
grid_result = gsc.fit(X_train, Y_train)
best_params = grid_result.best_params_
best_params
```

```
Fitting 5 folds for each of 36 candidates, totalling 180 fits
```

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 12 concurrent workers.
[Parallel(n_jobs=-1)]: Done  17 tasks      | elapsed:  5.1min
[Parallel(n_jobs=-1)]: Done 138 tasks      | elapsed: 28.3min
[Parallel(n_jobs=-1)]: Done 180 out of 180 | elapsed: 34.4min finished
```

```
CPU times: user 41.6 s, sys: 2.81 s, total: 44.4 s
Wall time: 35min 2s
```

```
/Users/macbookpro/Documents/naser_python/lib/python3.8/site-packages/sklearn/svm/_base.py:946: ConvergenceWarning: Li
blinear failed to converge, increase the number of iterations.
  warnings.warn("Liblinear failed to converge, increase "
```

```
{'C': 0.0001, 'epsilon': 0.1, 'tol': 1e-06}
```

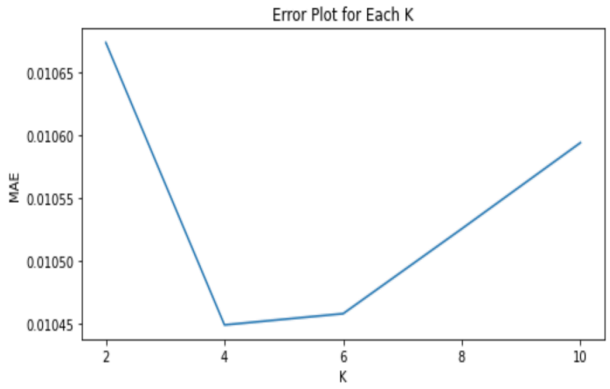
Figure 33: LinearSVR grid search result

```
The Evaluation Result For --->( LinearSVR ) is:
 * Mean absolute error = 0.0138
 * Mean squared error = 0.0007
 * Median absolute error = 0.0077
 * Explain variance score = 0.7641
 * R2 score = 0.76
 * Accuracy = 76.1 %
```

Figure 34: LinearSVR Evaluation result

- **KNeighborsRegressor:**
  ”This model was used local interpolation of the targets associated of the nearest neighbors in the training set to predict the Target value”
  Note: KNeighborsRegressor is sensitive for normalization

  Fig(36) show the behaviour of model with change number of (k)
  Fig(37) show the Evaluation for KNeighborsRegressor




```
as we noticed the erorr does not improve after K=4 and it also increase with more k so we will use it as follow...
```

Figure 35: KNeighborsRegressor behaviour with change value of (k)

```
The Evaluation Result For --->( KNN ) is:
 * Mean absolute error = 0.0104
 * Mean squared error = 0.0004
 * Median absolute error = 0.0055
 * Explain variance score = 0.8634
 * R2 score = 0.86
 * Accuracy = 82.47 %
```

Figure 36: KNeighborsRegressor Evaluation result

- **Random-Forest Regression:**
  since the number of data-set is large , Random-Forest was very helpful since its can handle and work efficiently with large data-set
  we do a grid search to Tune the parameter very well and the result was :
  Note: random-Forest is insensitive for normalization because its depands on decision trees

  Fig(38) show the grid search result
  Fig(39) show the Evaluation for Random-Forest Model
  Fig(40) show the performance for Random-Forest Model

```
#@title Default title text
%%time
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import accuracy_score, mean_squared_error
from sklearn.ensemble import RandomForestRegressor

# Random Forest with Grid Search
rf = RandomForestRegressor(random_state=100,n_jobs=-1)

param_grid = {
    "criterion": ['mse',],
    "n_estimators": [500, 600, 700],
    "max_depth": [60, 70, 80]
}

rf_model = GridSearchCV(estimator=rf, cv=3, param_grid=param_grid, verbose=2)
rf_model.fit(X_trainn, y_trainn)
print(rf_model.best_score_)
print(rf_model.best_estimator_.get_params())
```

```
RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                      max_depth=60, max_features='auto', max_leaf_nodes=None,
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_leaf=0.0,
                      n_estimators=600, n_jobs=-1, oob_score=False,
                      random_state=100, verbose=0, warm_start=False)
```

Figure 37: Random-Forest grid search result

```
The Evaluation Result For --->( RandomForest ) is:
 * Mean absolute error = 5357.7126
 * Mean squared error = 114816972.5589
 * Median absolute error = 3009.8067
 * Explain variance score = 0.928
 * R2 score = 0.93
 * Accuracy = 89.61 %
```

Figure 38: Random-Forest Evaluation result

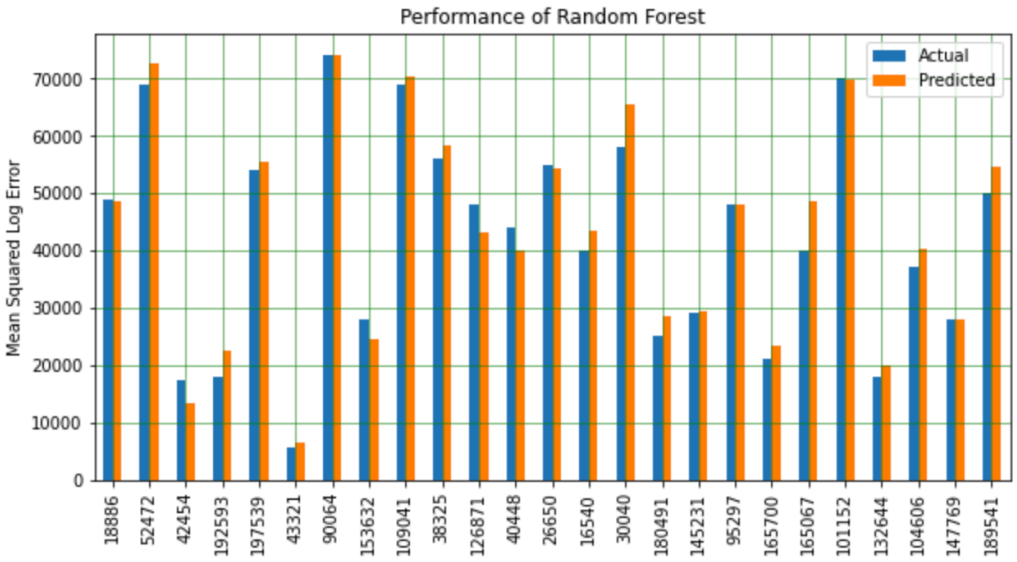


Figure 39: plot describe performance for Random-Forest

Note: as you see Random-Forest was get a best Accuracy and less MAE from other Models so we want to add some enhancement on it aiming to reduce the MAE and increase the Accuracy

- **XGBoost(Gradient Boosted Decision Trees):**
  Definition"Simply : Gradient boosting is a method that goes through cycles to iteratively add models into an ensemble, Random Forest is Consider as a Bagging algorithm so its reduces variance.
  Boosting reduces variance, and also reduces bias. It reduces variance because you are using multiple models (bagging). It reduces bias by training the subsequent model by telling him what errors the previous models made (the boosting part)".

  Note: XGboost is insensitive for normalization because its depands on decision trees

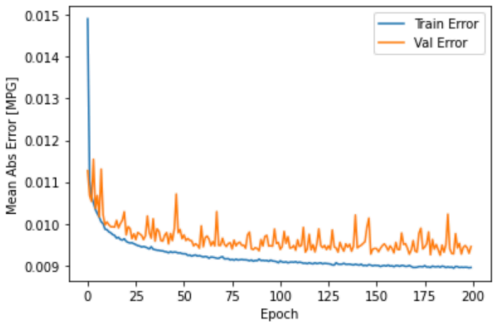
  Fig(41) show the Evaluation for XGBoost Model

```
The Evaluation Result For --->( XGBoost ) is:
 * Mean absolute error = 5400.4009
 * Mean squared error = 113826547.3873
 * Median absolute error = 3038.5273
 * Explain variance score = 0.929
 * R2 score = 0.93
 * Accuracy = 89.71 %
```

Figure 40: XGBoost Evaluation result

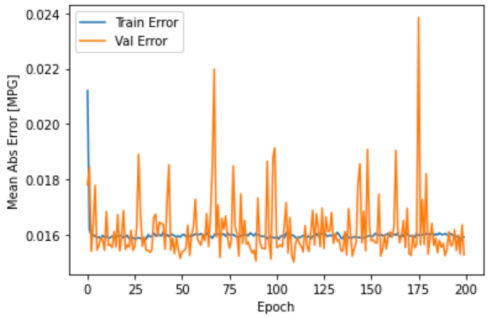
- **Deep Neural Networks**
  As a step to apply more model and use powerful Technique,we decide to deal with Deep Neural Networks (keras) so we try different parameter to tune the Neural Network perfectly and the Result are as follows ...
  Note: Neural Networks is sensitive for normalization

  Fig(42+43+44) will show experiment results Fig(45) will show the Neural Networks Evaluation result
  Fig(46) will show the Final Structure for Neural Networks that we used



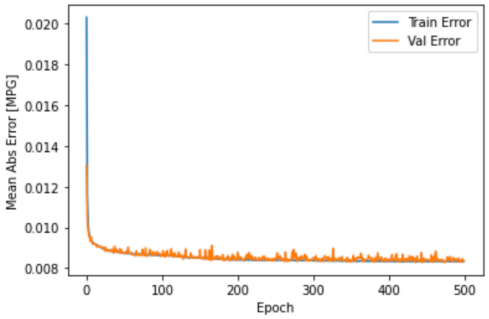

```
----------------------------------------------------------------------------------------------
Overview for the Curve:-
* The number of train_size ----> 0.7
* Normalization ----> MinMaxScaller
* Number of hidden layer is ---> (2) --> First layer(8) , Secand layer(4) with kernal_intalizer is (normal)
* batch Size was ---> 16
* The number of epoch is ------> 200
* the Dropout doesnt applay
```

Figure 41: First attempt by Neural Networks on 100k row




```
--------------------------------------------------------------------------------------------------
Overview for the Curve:-
* The number of train_size ----> 0.7
* Normalization ----> MinMaxScaller
* Number of hidden layer is ---> (2) --> First layer(8) , Secand layer(4) with kernal_intalizer is (normal)
* batch Size was ---> 16
* The number of epoch is ------> 200
* the Dropout Doesnt applay
* we use ---> kernel_regularizer=tf.keras.regularizers.L1(0.001)
```

Figure 42: secand attempt try to enhance by regularization




```
--------------------------------------------------------------------------------------------------
Overview for the Curve:-
* The number of train_size ----> 0.7 and validation split is ----> 0.3
* Normalization ----> MinMax Scaler
* Number of hidden layer is ---> (2) --> First layer(8) , Secand layer(4) with kernal_intalizer is (normal)
* batch Size was ---> 16
* The number of epoch is ------> 500
* the Dropout was applay with 0.001
* we use ---> kernel_regularizer=tf.keras.regularizers.L2(0.0001)
* we Modify the learning rate to ---> 0.0001
* mean_absolute_error is used for loss Function
```

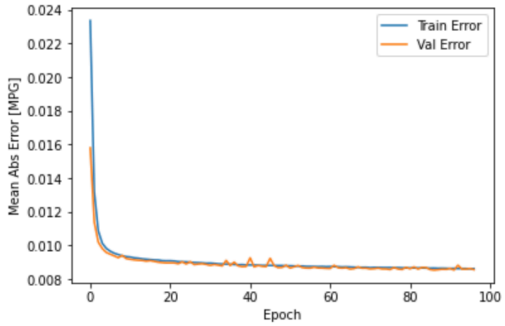
Figure 43: Final Neural Networks Evaluation result

```python
def Build_Model():
    NN_model = Sequential()

    # The input & Hidden Layers :
    NN_model.add(Dense(8,kernel_initializer='normal',activation='relu',input_dim = X_train.shape[1]
                       ,kernel_regularizer=tf.keras.regularizers.L2(0.0001)))
    NN_model.add(Dense(4, kernel_initializer='normal',activation='relu'
                       ,kernel_regularizer=tf.keras.regularizers.L2(0.0001)))

    NN_model.add(tf.keras.layers.Dropout(0.001))

    # The Output Layer :
    NN_model.add(Dense(1, kernel_initializer='normal',activation='linear'))

    opt = tf.keras.optimizers.Adam(learning_rate=0.0001)


    # Compile the network :
    NN_model.compile(loss='mean_absolute_error', metrics=['mean_absolute_error'], optimizer=opt)
    NN_model.summary()

    return NN_model
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 8)                 5504
_________________________________________________________________
dense_1 (Dense)              (None, 4)                 36
_________________________________________________________________
dropout (Dropout)            (None, 4)                 0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 5
=================================================================
Total params: 5,545
Trainable params: 5,545
Non-trainable params: 0
_________________________________________________________________
```

Figure 44: Neural Networks Structure

As you see we use 500 epoch to train the Model but there's a Technique called early-stopping that tell where the last position Neural Networks doesn't improve after , so the next step will be the Result for This Technique as follow...



From the previos earlly Stoping Plot we noticed that Model become stable and doesnt improve after 100 epoch , so 100 epoch is very good

Figure 45: Neural Networks Evaluation result after early-stopping

5.7 Machine Learning Conclusion

This section will show a Table contain the result for each Model we used

Machine Learning Algorithm	MAE	MSE	R2 score	Accuracy
LinearSVR	0.0138	0.0007	0.76	76.1 %
KNeighborsRegressor	0.0104	0.0004	0.86	82.47 %
RandomForestRegressor	0.0072	0.0002	0.93	89.61 %
Gradient Boosting Regresser	0.0097	0.0002	0.93	89.72 %
Deep Neural Networks	0.0083	0.0002	0.92	—-

Table 1: compere between each of ML algorithms

6 SOUQY application

6.1 Brief

After collecting all the problems we noticed during the process of cleaning data and the problem with publishing car advertisement in all Palestinian website, we started to think seriously for create an application that Facilitates this task and solve all problem that we faced. In addition, we integrate a Machine learning Model that allows the user to check for the correct price before buying anything.

6.2 Technology used

since our software is a mobile application, so we search for a UI software development kit that supports development for multi-platform applications. then we found that flutter is a suitable one and achieve our purpose.

for tools that we used in the backend, we decide that a firebase is a great tool in terms of user authentication and data, image storage, in addition, it provides a faster connection.

for integration between ML and application, we generate an API code written by flask framework in python used to retrieve the predicted price value for a specific car from the model.

6.3 User requirements

As we mentioned before the Target for our mobile application is to solve the problem that faced users and add some enhancement that distinguishes our application. So this section will show each page in the application and describe its content and importance as follow.

- **User page:**
  This page only appears for those people how registered in the application.it contains basic information about the user. so every time the user needs to sell a car there's no required to rewrite this information again
  Note: user can modify his information.

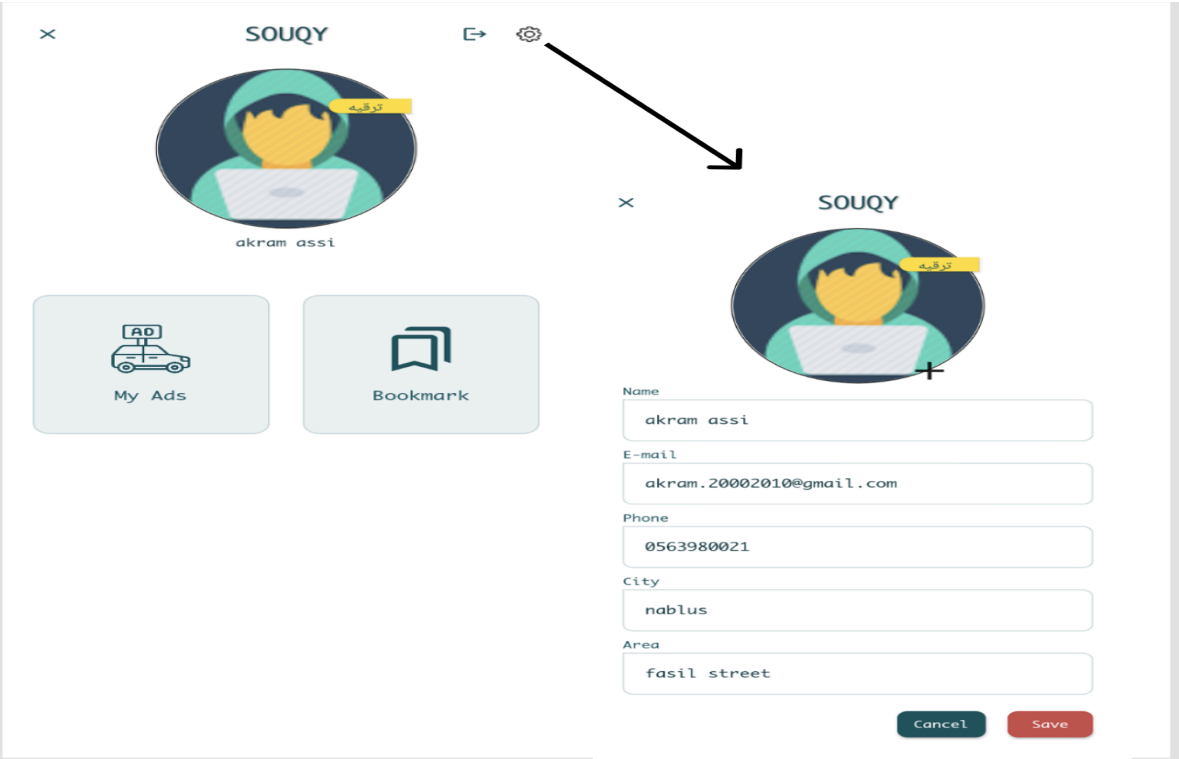


Figure 46: user page in SOUQY app

- **Home Page:**
  we chose this page to be a simple page and at the same time it contains the information that the user is looking for so we divided the page into a set of cards each card represent a car and it contains Important information(make, model, price, if installment, produce year, gear-type, fuel-type) as follow.

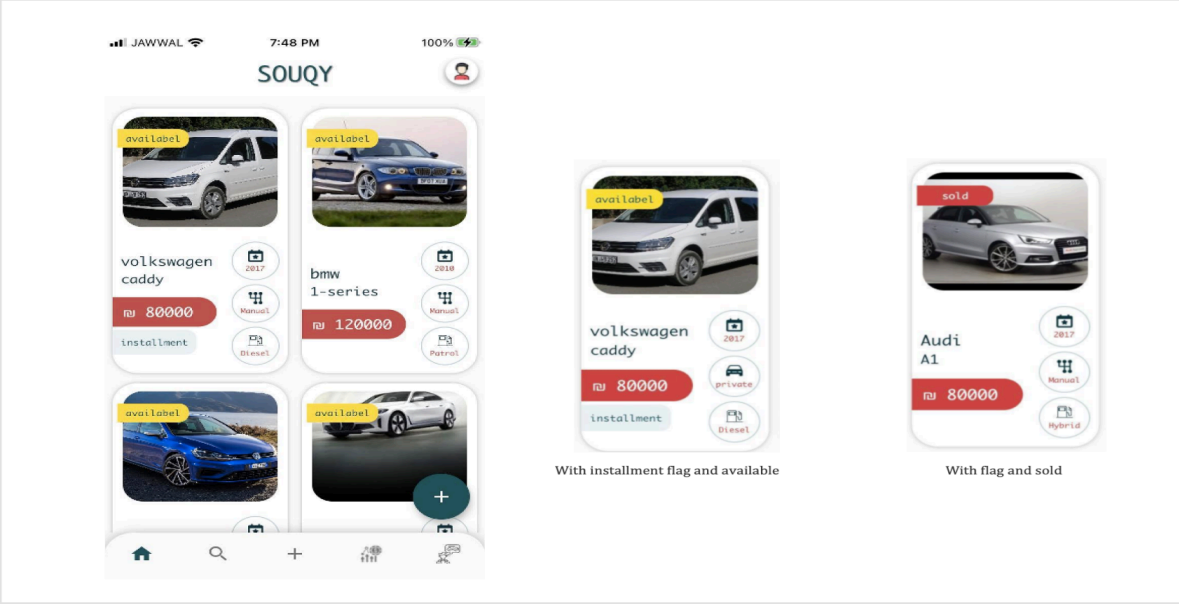


Figure 47: Home page in SOUQY app

- **Add vehicle page:**
  This page is the most important page within the application, this is due to the reason that the user is who enters the information, so we have added several conditions to the input fields to prevent problems that may happen with different structure input by user
  The next lines will describe each filed and how user interact with as follow ...

  - **vehicle Company Name field:** Here we allow the user to interact with the field by writing the name or just choosing the logo for the car

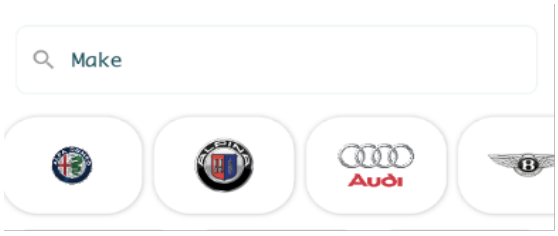


Figure 48: Car Company Name Field

  - **vehicle model Name field:** we improve this section by adding autocomplete text field enhancement, so the user just input on 1 or 2 characters and the system will suggest the vehicle model of a selected make company

Figure 49: Car model Name Field

– **vehicle body style field:** This field helps with the prediction process, the user just select the shape for his vehicle

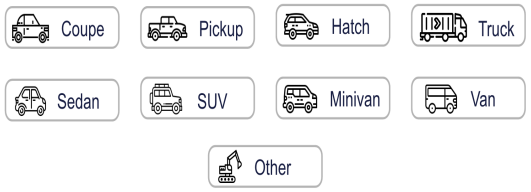


Figure 50: vehicle shape type Field

– **vehicle produce year field:** we use dialog in this field to represent the vehicle produce year.

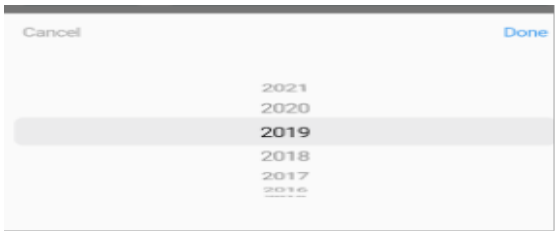


Figure 51: Car produce year Field in SOUQY app

– **Kilometers field:** This field was the most important field for us, because of the big problems that we faced in the previous data due to the different methods of expression between users, so we put restrictions on this field and just allow the user to only entering numbers and not words.

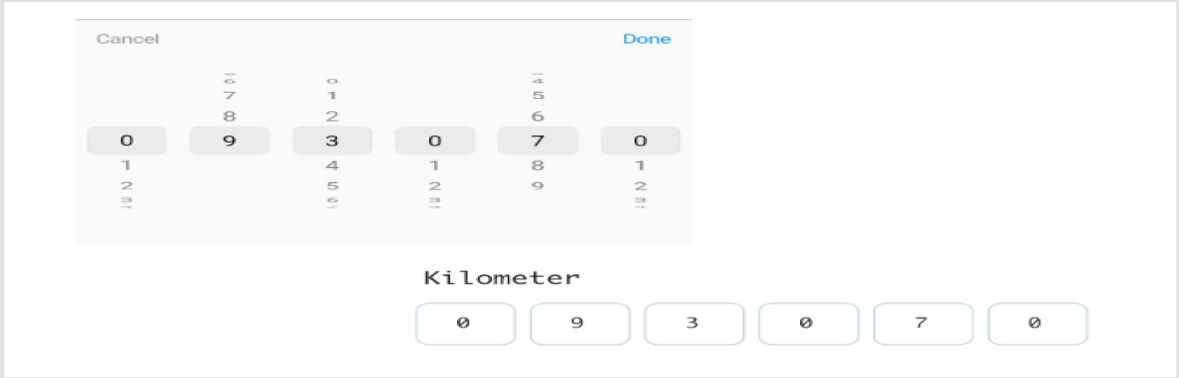


Figure 52: Kilometers field in SOUQY app

– **Engine-size and passenger field:** In these fields, we have given the user freedom to either choose from a dialog or enter a number using a keyboard.

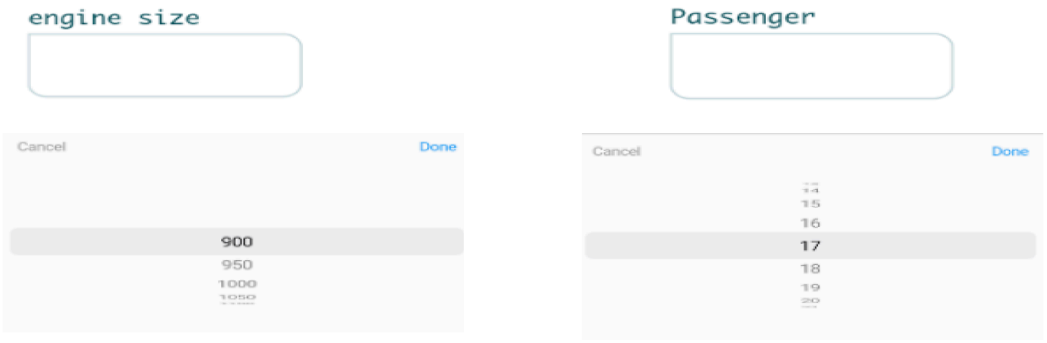


Figure 53: Engine-size and passenger field in SOUQY app

– **color field:** In this field we use Dialog to display a menu for multiple colors

Figure 54: Color field in SOUQY app

– **Gear type, fuel type and origin field:** we used drop list to show the options

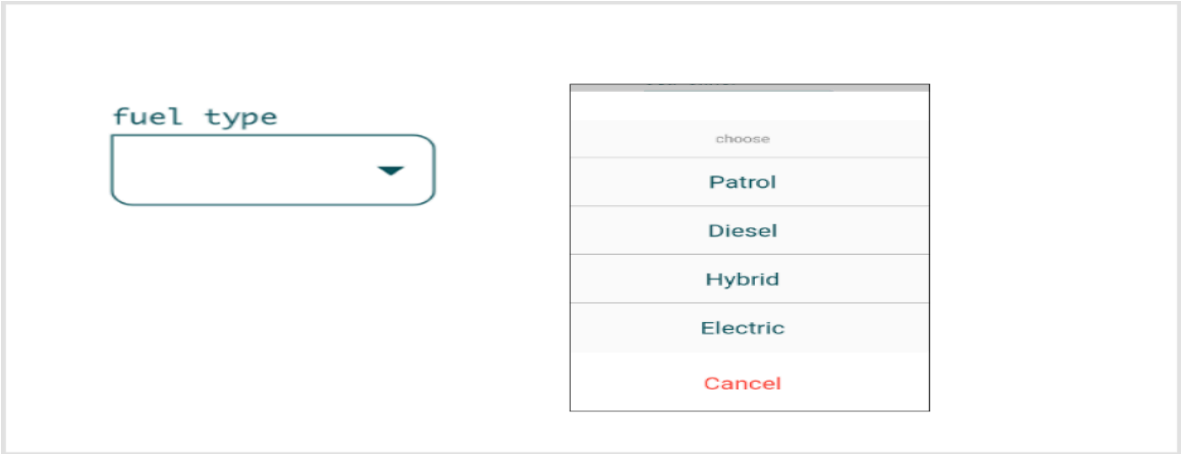


Figure 55: Color field in SOUQY app

– **previous owners:** we put a restrictions on this field and just allow the user to only entering numbers and not words by increasing or decreasing a counter

Figure 56: previous owners Field

– **Payment method field:** we used drop list to show payment option method (cash, installment).in addition if the user choose installment a two additional field will be appear to represent the value of down payment monthly payment as show on Fig ...

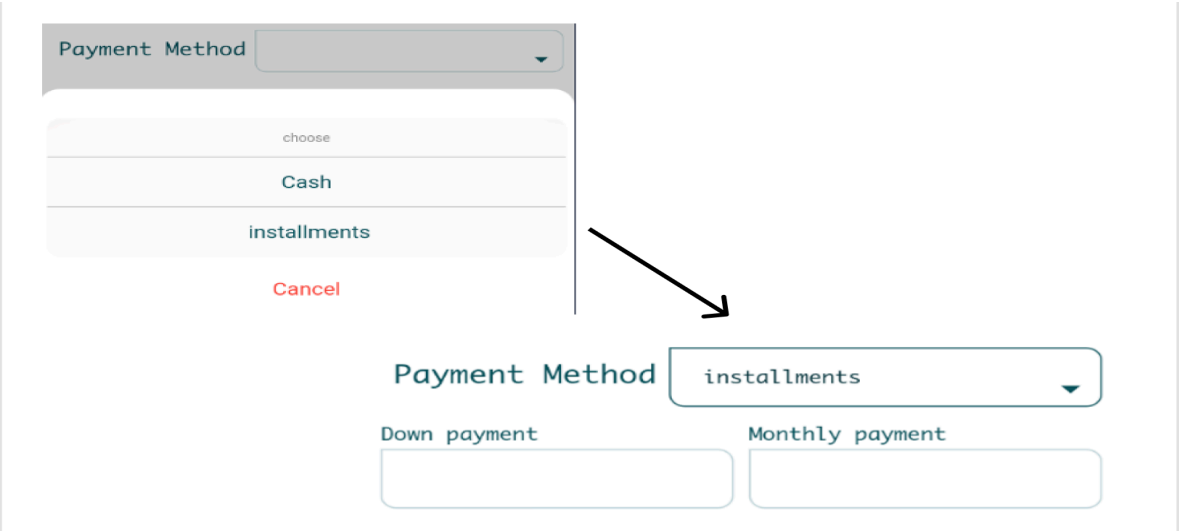


Figure 57: payment-metoud in SOUQY app

– **Additional features:** Here we decided to put some common features that every vehicles maybe contain and the user can select between them, if there's extra information that user need to add it can write in additional information filed

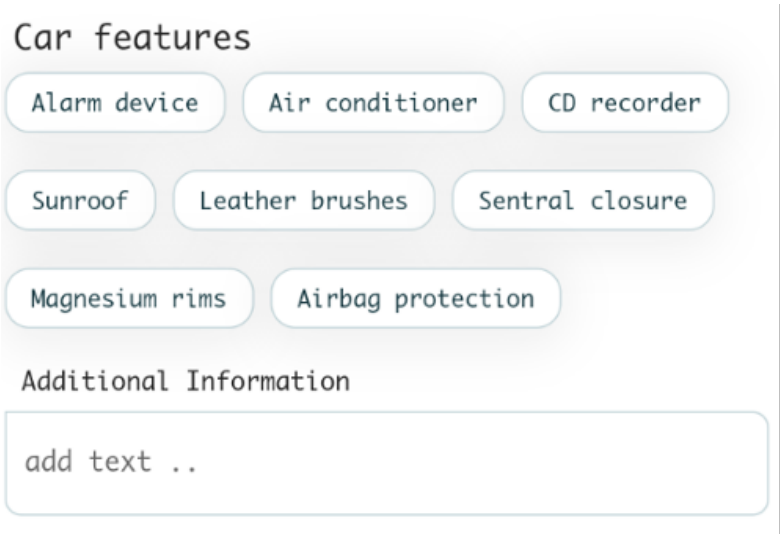


Figure 58: Additional features in SOUQY app

- **Expect page:** similar to add page but the differences between them that the first contain only the fields that necessary for the prediction process. so after fill all the Fields and press expected an object will be prepare and send as a request to the machine learning API to get the prediction value for the entered car.

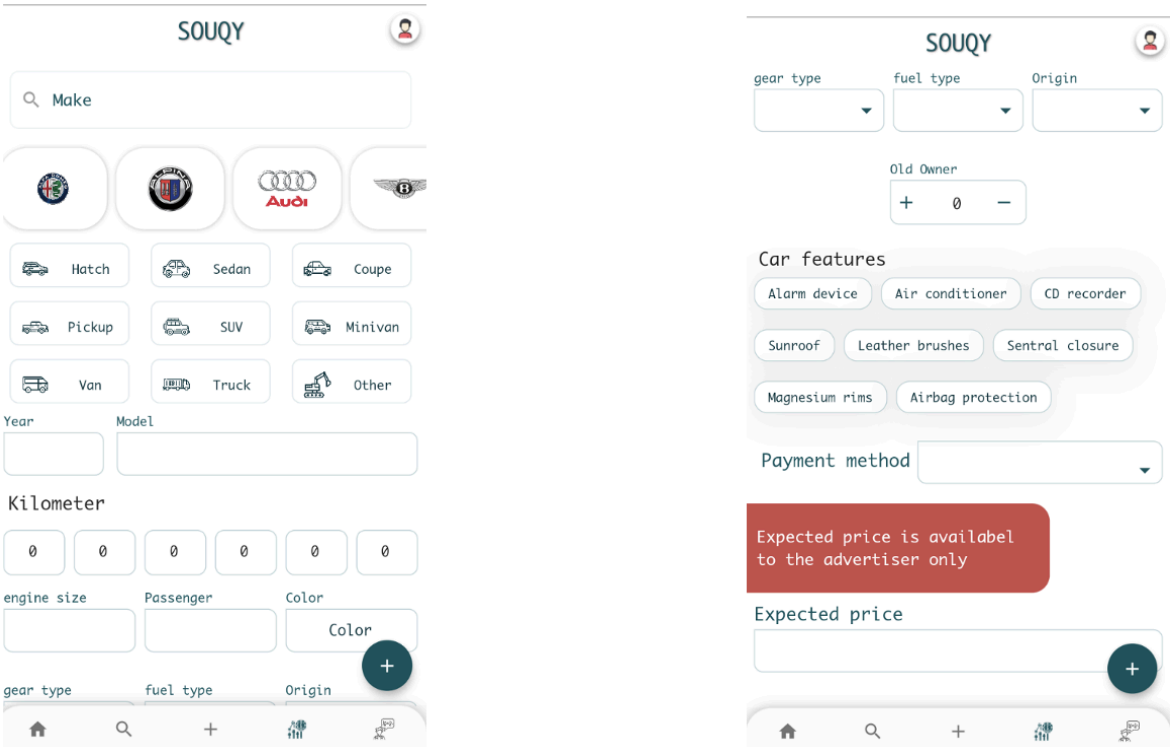


Figure 59: Expact page in SOUQY app

- **more info page** This page contains all details about the owner and the vehicle. In addition, we provide the owner ability to modify or remove the advertisement, while the visitor can put a car in a bookmark so it can reference later time

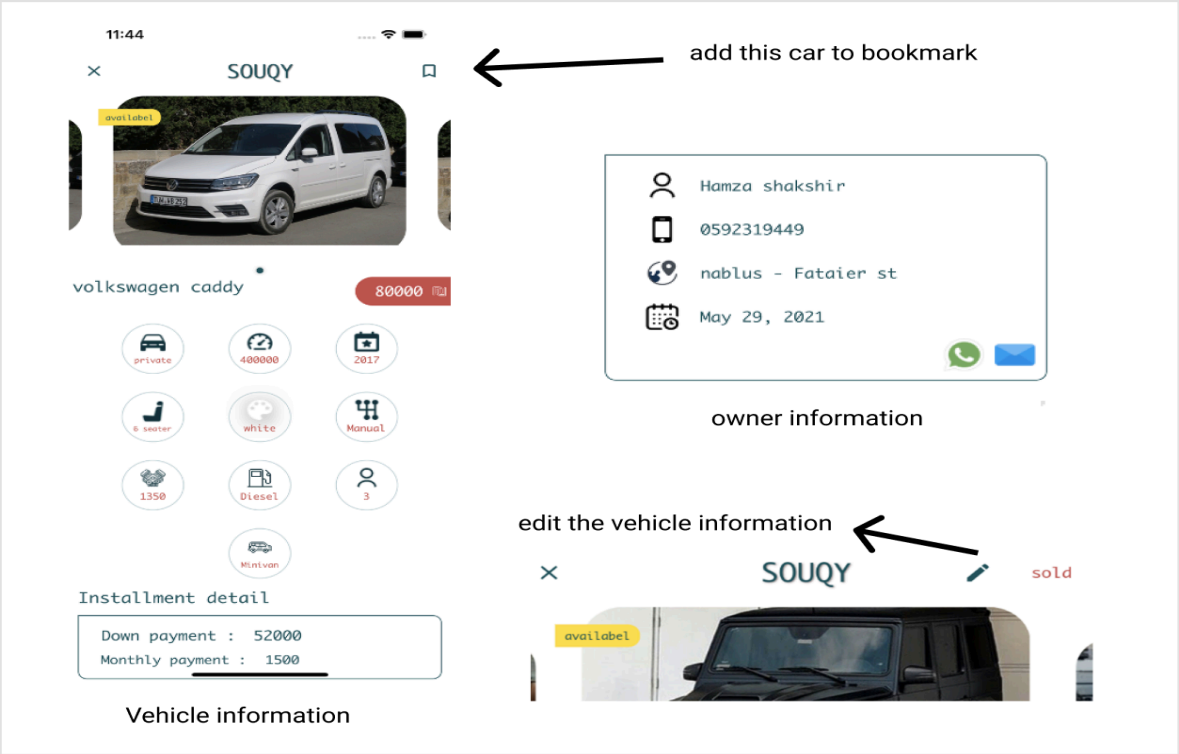


Figure 60: more info page in SOUQY app

- **Statistics page:** This page show how the price for specific vehicle change over a period of time .all you need just enter(make , model , year of produce) and the plot will show as follow

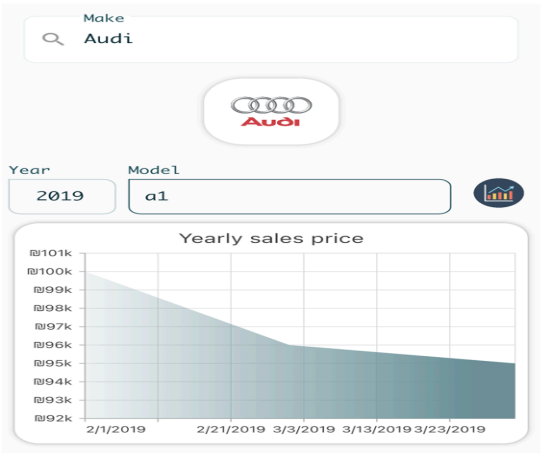


Figure 61: Statistics page in SOUQY app

- **Search page:** This page is the kernel for application since each advertisements will show on it , we provide different filtered search to allow the user find the needed car quickly

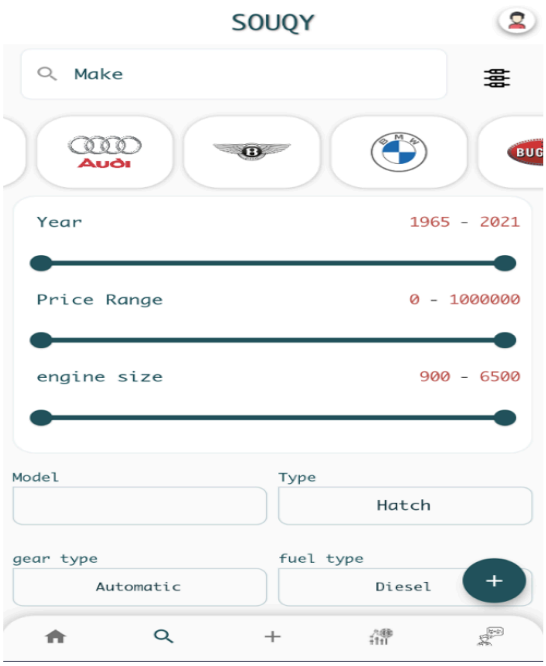


Figure 62: search filtered page in SOUQY app

- **Validation Technique:**
  This steps is consider the important step in the application, since the user doesn't allow to enter any information in any required filled on application,these step leads to get an cleaning and trusted data latter
  Note: Fig(63) will show the restriction that put on required filed

Figure 63: validation filed in SOUQY app

- **System Design:**
  - **Activity Diagram**
    - Create account

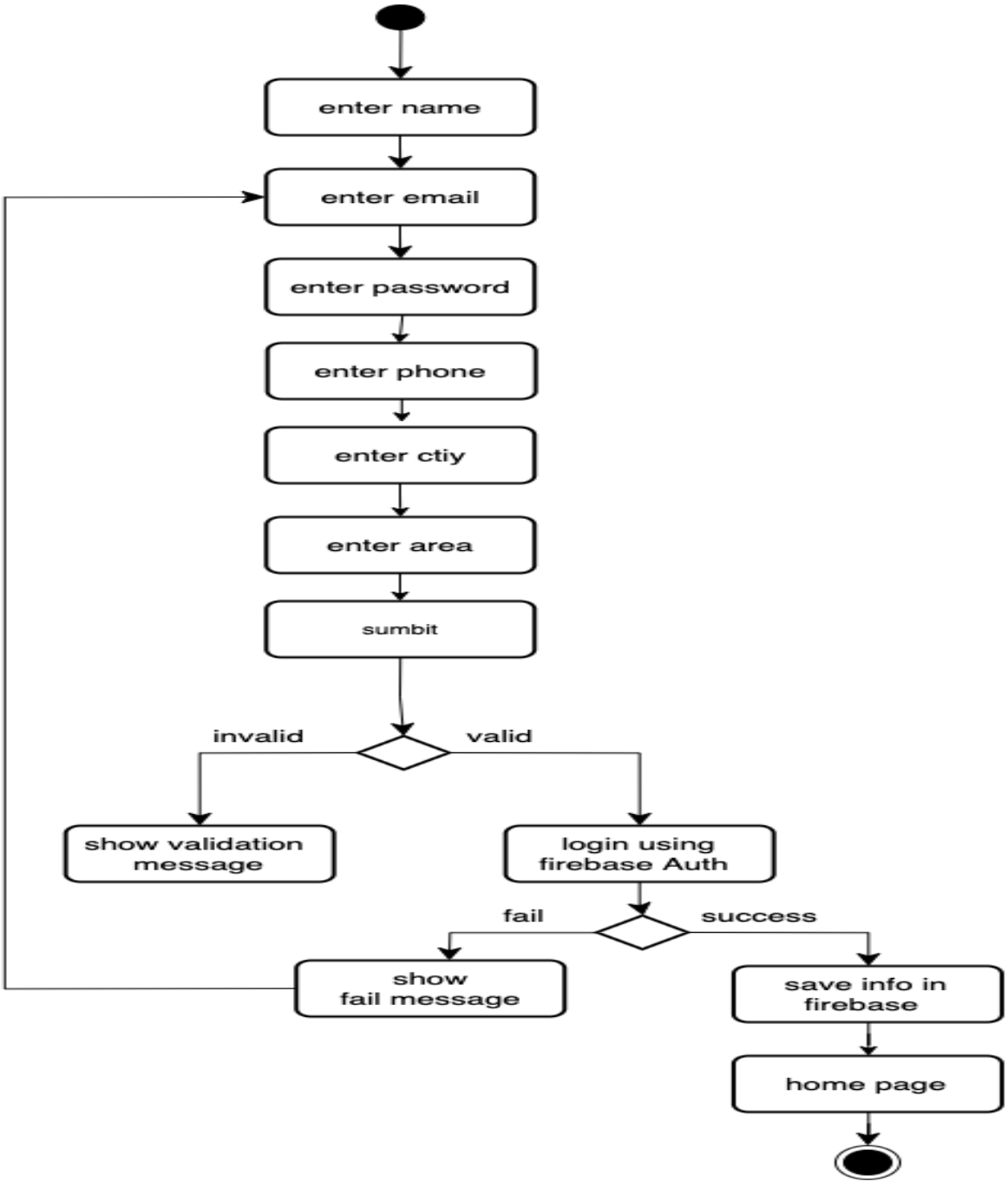


Figure 64: Create account activity diagram in SOUQY app

* Login

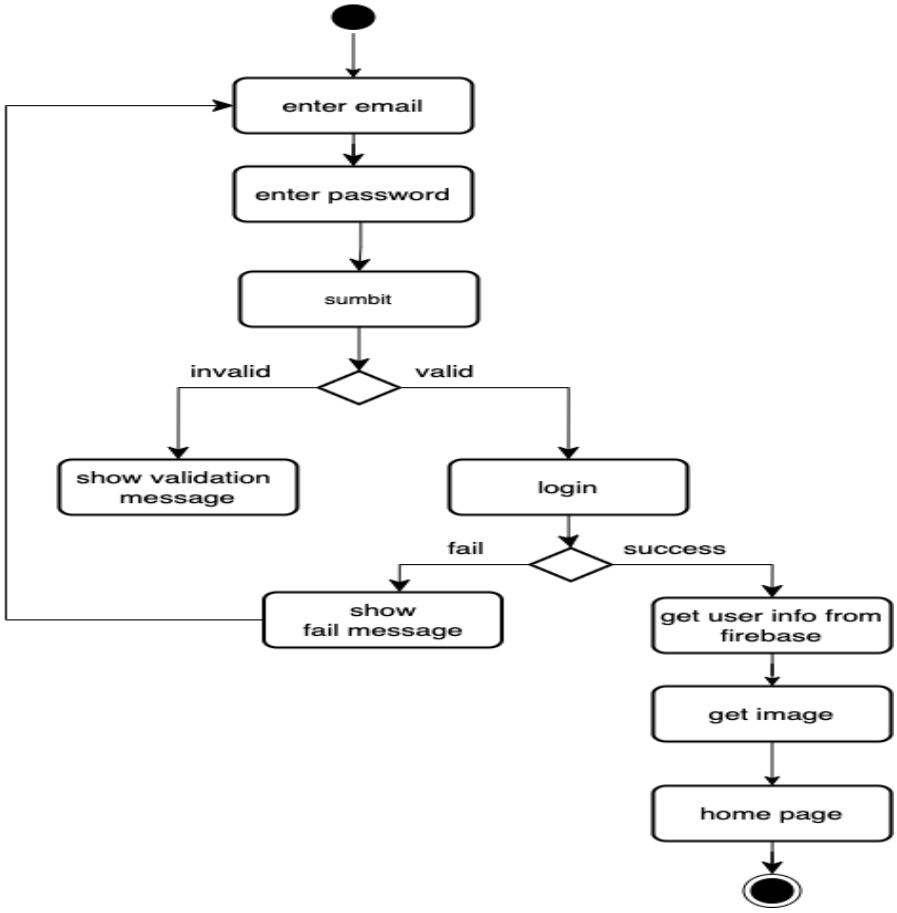


Figure 65: Login activity diagram in SOUQY app

* Add bookmark

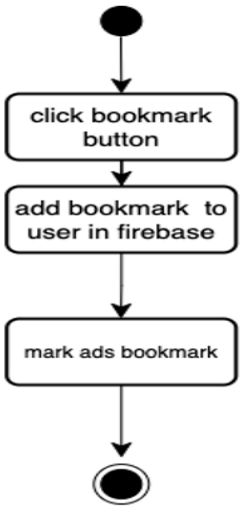


Figure 66: add bookmark activity diagram in SOUQY app

* Google Login

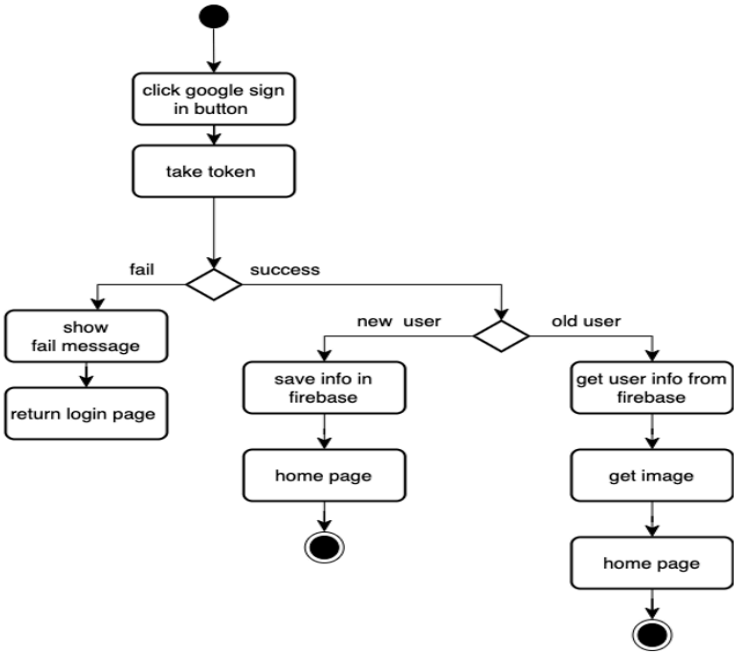


Figure 67: Login by Gmail activity diagram in SOUQY app

* Facebook Login

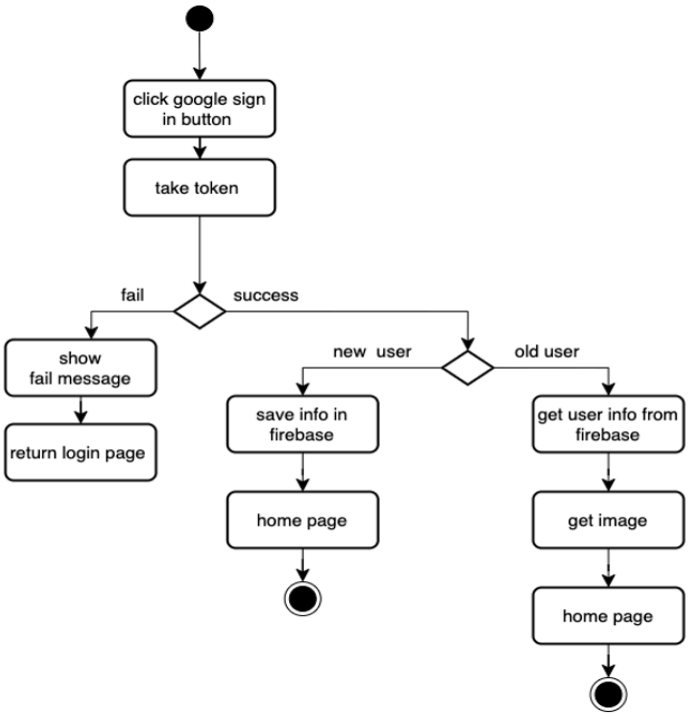


Figure 68: Login by facebook activity diagram in SOUQY app

* More info

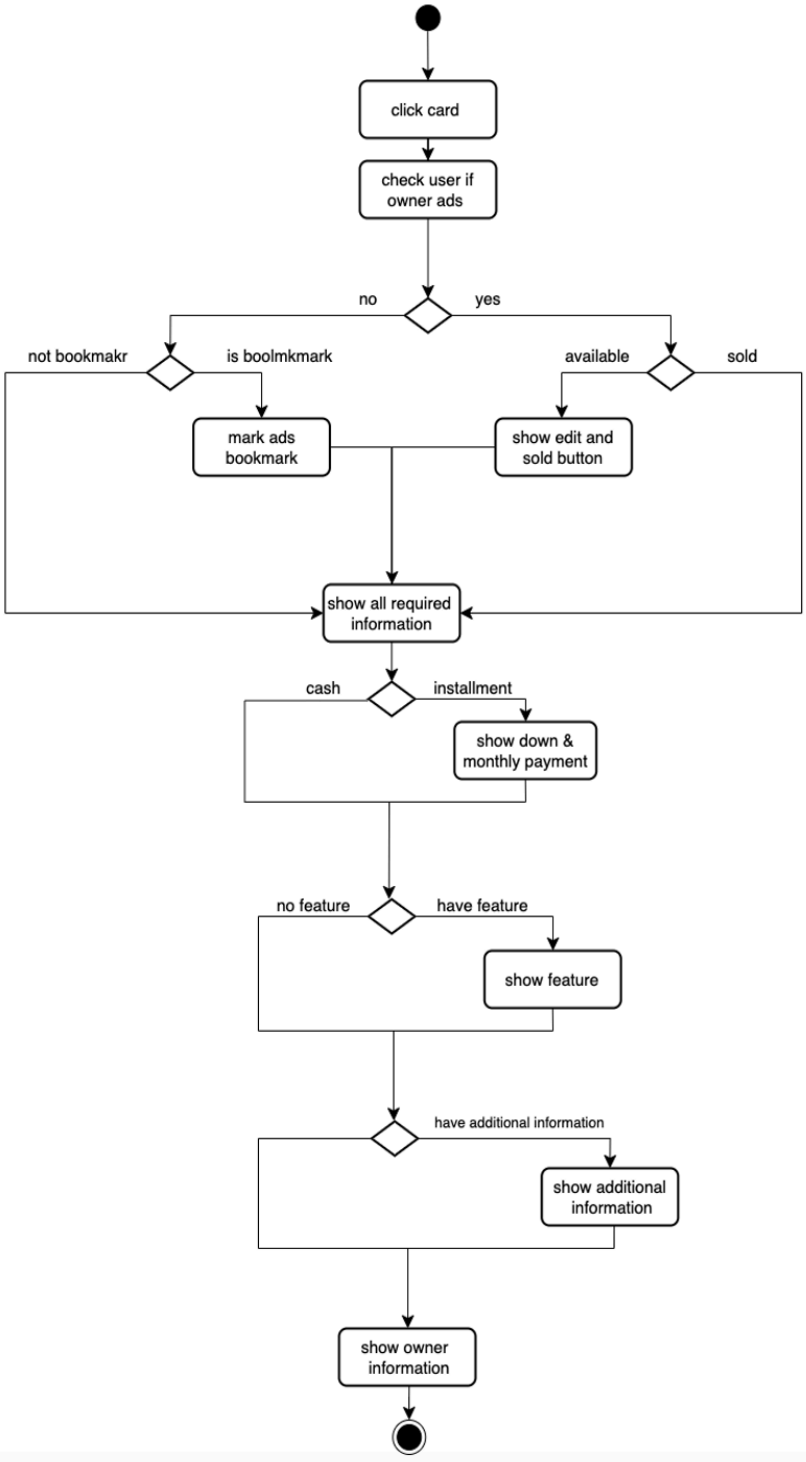


Figure 69: more info activity diagram in SOUQY app

* sold vehicle

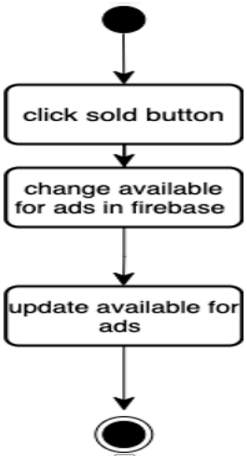


Figure 70: sold vehivle activity diagram in SOUQY app

* Add ads

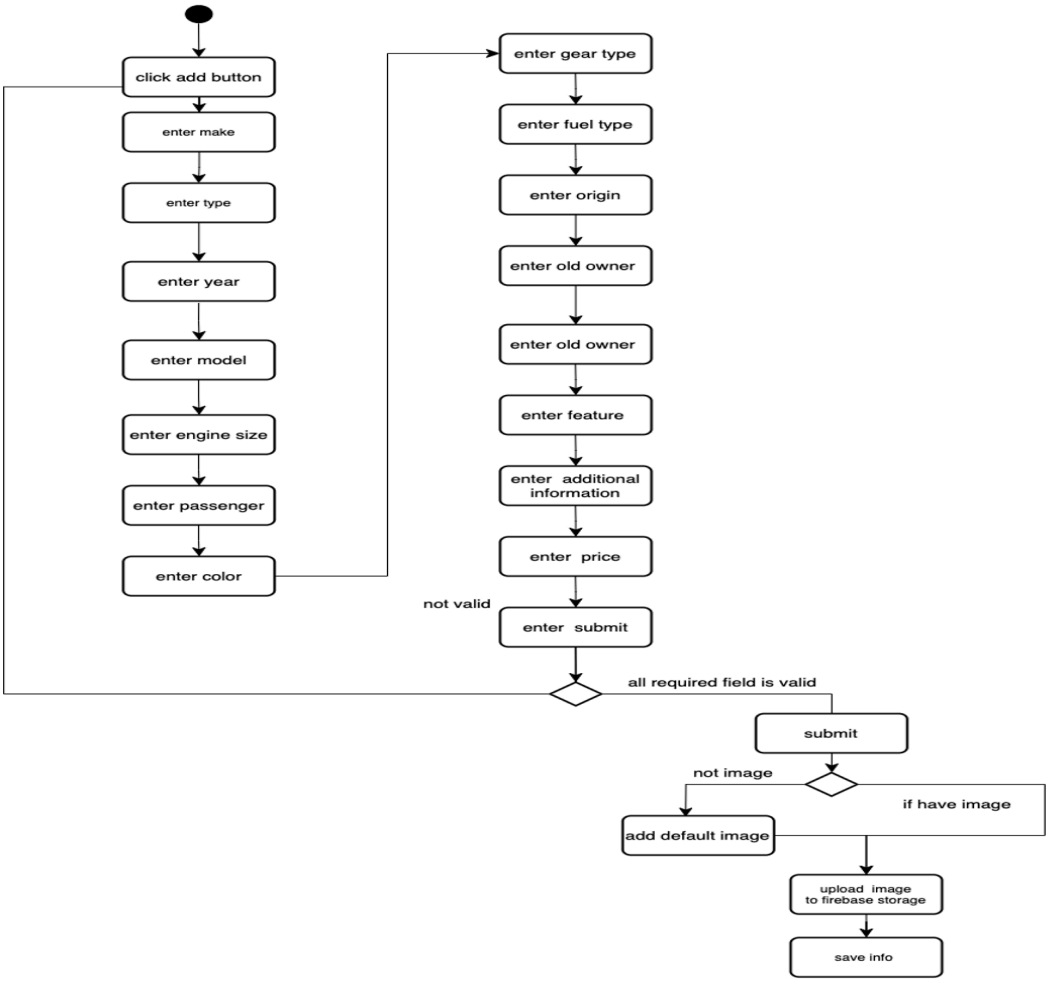


Figure 71: add add activity diagram in SOUQY app

* expect price

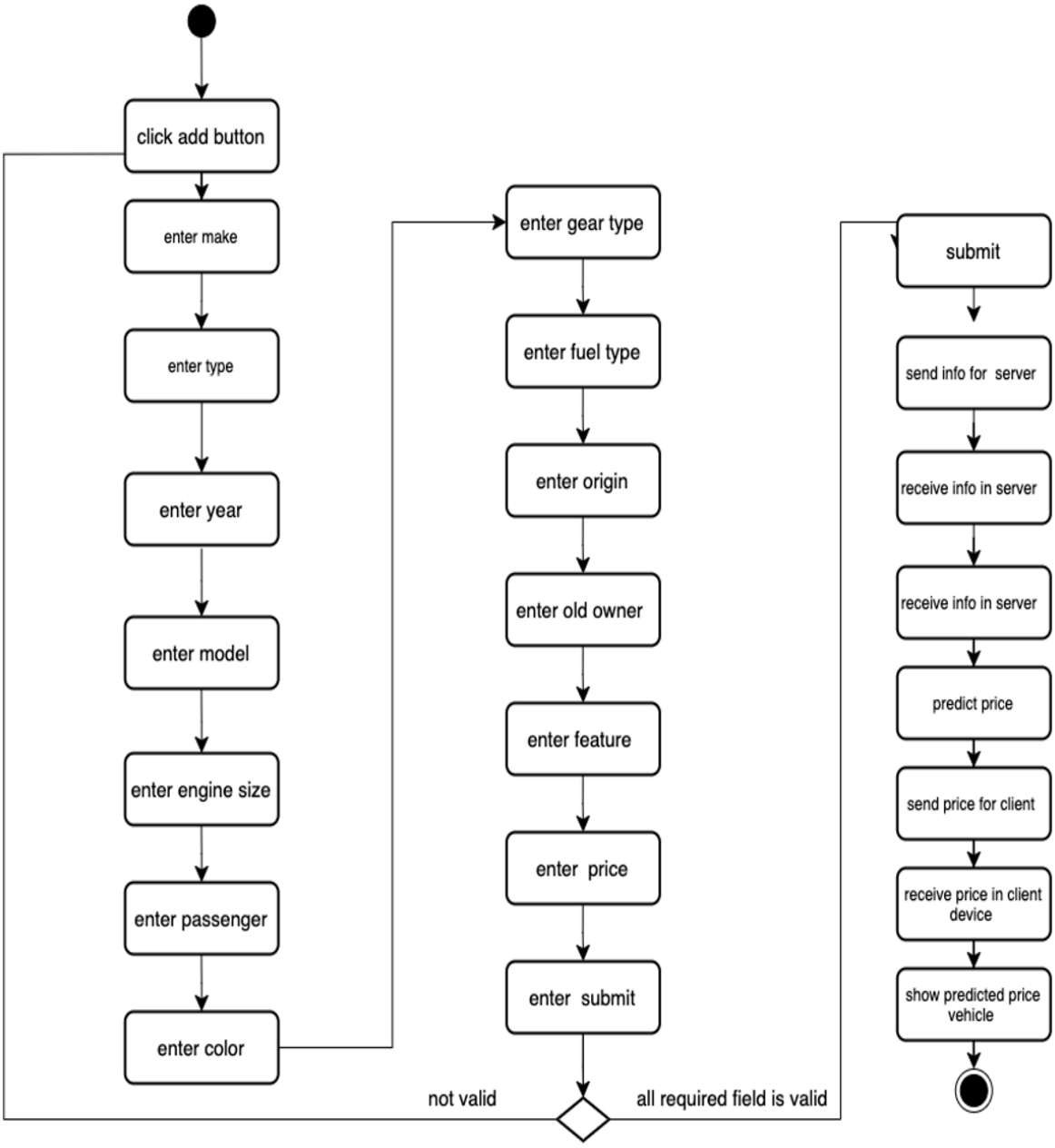


Figure 72: expact price activity diagram in SOUQY app

* statistics

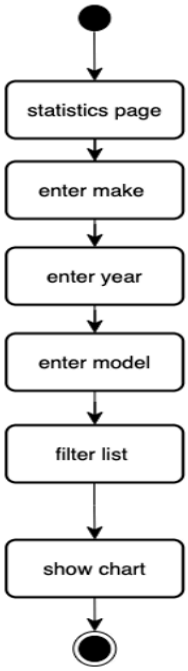


Figure 73: statistics activity diagram in SOUQY app

* update user information

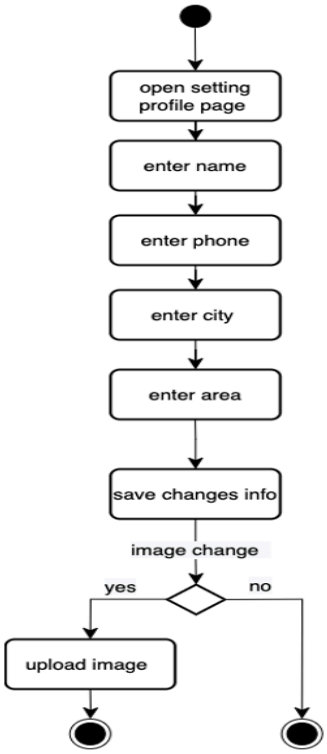


Figure 74: update user information activity diagram in SOUQY app

– **Class Diagram**

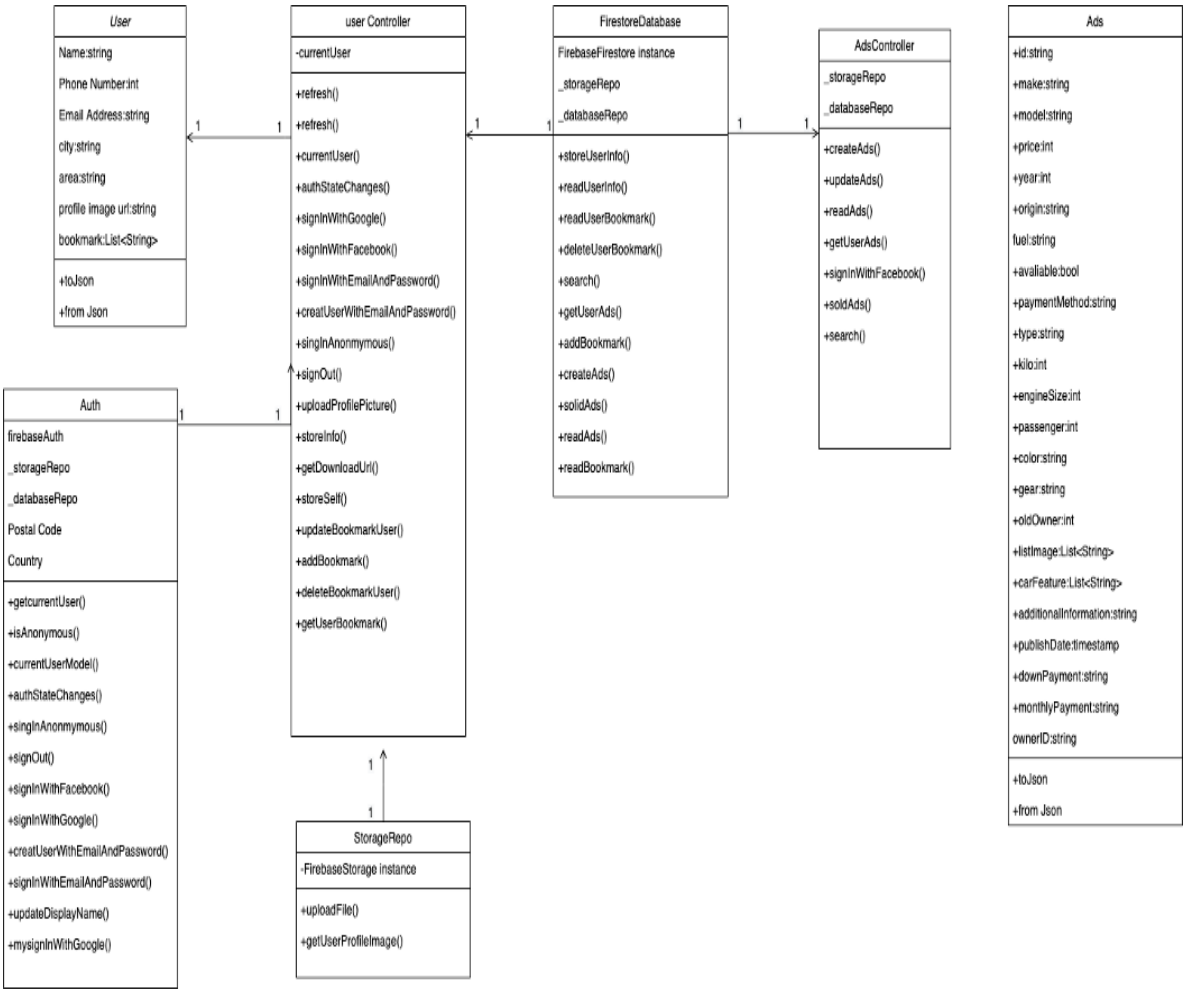


Figure 75: class diagram in SOUQY app

– **Fire-base Diagram**

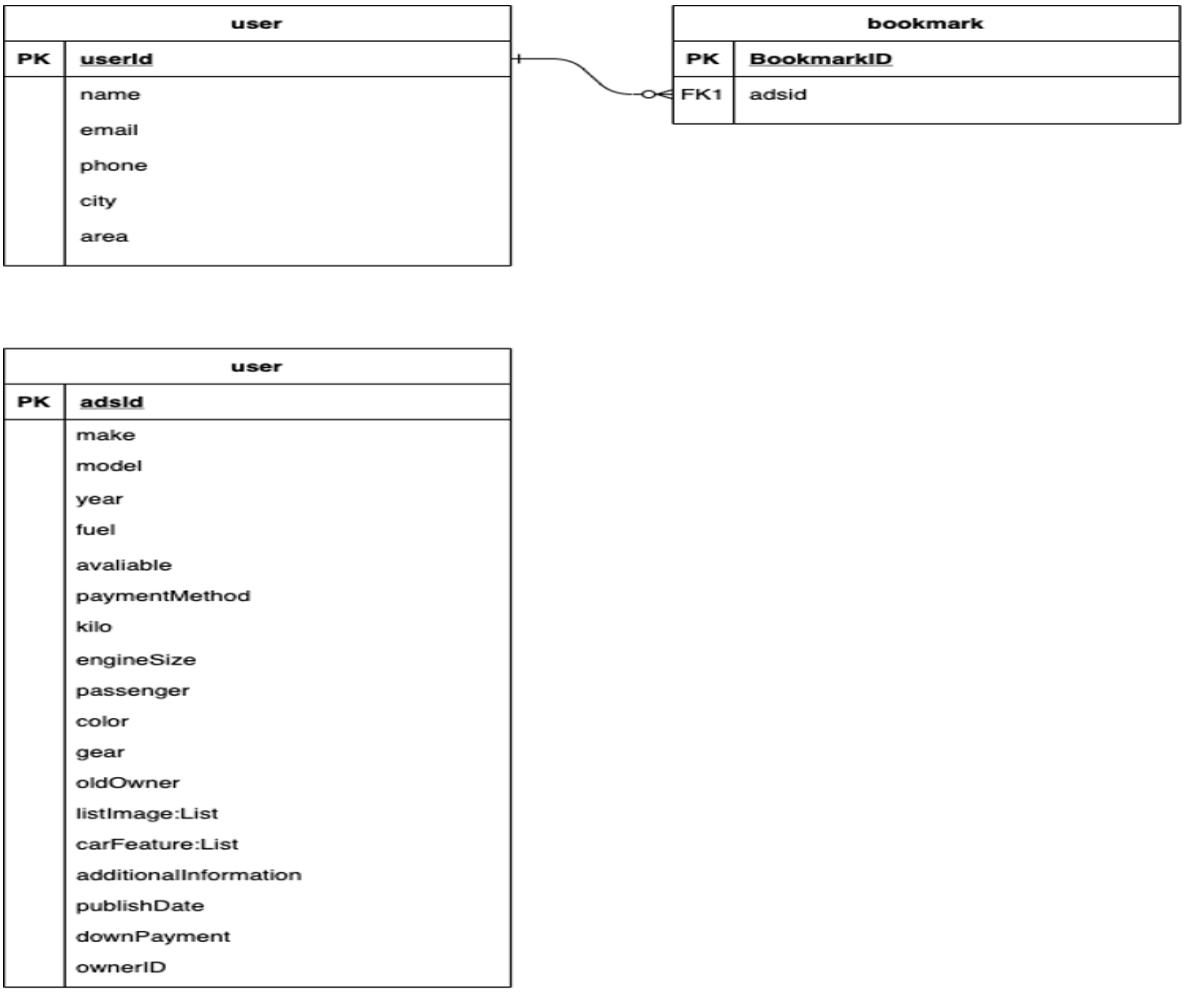


Figure 76: database diagram in SOUQY app

7 Conclusion

By the end of this project, we conclude that the future of Data Science looks exciting and full of challenges, there's a lot of obstacles that we faced during the cleaning process since the data is contained a lot of errors and problems, but Working on this project gained us huge abilities and skills in searching, collaboration and team working we also were able to learn a lot of useful tools which provided by python like TensorFlow, NumPy, Keras, flask and also flutter framework, dart language that we used to develop our application and of course using git and latex to write this report.

8 Future Work

searching for more feature that affects the price will be the best goal to achieve such as (currency difference, state's situation, text additional information that user provide in his advertisement such as "تامين جديد","سواقة سيدة",....).In addition, we will create different models that suggest the user to buy his suitable vehicle(Recommendation system)