**Project Documentation**

Analysis of the used cars market in Egypt (2022)

In this document I’ll showcase the workflow taken to analyze the dataset I’m working with, making models to get different types of variables and trying to optimize them and manually going through the data for the purpose of drawing as many conclusions about the used car market in Egypt

Once again keeping in mind the dataset is an outdated one from 2022 and is limited in scope being scraped from only 1 website

For data accuracy’s sake I’ll use the full 14000 records from the dataset although this might cause orange to freeze every time there’s an edit in the data, for this reason I’ve made a Car-sample.csv with less records incase changes are needed

The document will be divided into 5 sections:

* **Data Models**

Making data models, rating their performance and going

* **Visual data analysis**

Looking at visual representations of the data and drawing conclusions manually

* **Conclusion**

Lining up all the info and conclusions we could

* **Screenshots**

Extra images from the orange project

* **Credits**

Team info and Dataset credits

# Data Models

### Linear Regression

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MSE | RMSE | MAE | MAPE | R2 |
| 998.673 | 31.602 | 21.181 | 0.390 | 0.852 |

Linear regression is the simplest machine learning model which doesn’t allow for much optimization other than simply ignoring irrelevant data (ID in this case), our target will be the only numeric value, the **price**

Linear regression performance table

* **RMSE:** By average, the model gets the value off by 31602 EGP
* **MAE:** The average error is 21181 EGP, difference between this and RMSE is this is less sensitive to outliers, so far off predictions contribute about a third of the error, likely due to this being a public platform and some sellers pricing their cars at ridiculous prices
* **MAPE:** The model has an *absolute average percentage error* of 39%
* **R2:** There is 85% variance in the data, indicating a good fit

### Logistic Regression

For a start, I tried to do the same thing only with the target being **Brand** this time, however the model as a result gets a perfect rating in all aspects and it is apparent why:

Each brand has its model set that would never occur in a record with a different brand making its existence in a record give a 100% change of a certain brand, same thing happens when model is the only feature

I could make the model the target but I’ll leave that for the tree model, instead I’ll try and predict the **Milage**, I’ll try different features to try and remove the unimportant ones and get as accurate of a result as possible using keep in mind ID is always excluded as it is irrelevant, here’s a table representing the tests:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Excluded Features | AUC | Accuracy | F1 | Precision | Recall | MCC |
| None | 0.697 | 0.215 | 0.163 | 0.188 | 0.215 | 0.113 |
| Color | 0.687 | 0.207 | 0.134 | 0.153 | 0.207 | 0.097 |
| Gov | 0.688 | 0.202 | 0.134 | 0.183 | 0.202 | 0.093 |
| Brand, Model | 0.659 | 0.203 | 0.135 | 0.150 | 0.203 | 0.091 |
| Brand, Mode, Gov, Color, Body | 0.617 | 0.180 | 0.094 | 0.098 | 0.180 | 0.058 |
| ALL except price & year | 0.591 | 0.183 | 0.081 | 0.053 | 0.183 | 0.053 |

Excluded features to performance table

Unexpectedly, the model gives best results when we exclude nothing, seems that all features play into the milage even color

Nonetheless the choice of making milage a class instead of a numeric value is a poor one likely by the website of origin for the data

The most important parameters of the final model:

* **AC:** Only about a **fifth** of the predictions are correct
* **Precision:** The proportion of true positive predictions out of all positive predictions made by the model is **0.188**
* **Recall:** The proportion of true positive predictions to all actual positive instances is **0.215**

### Tree Model

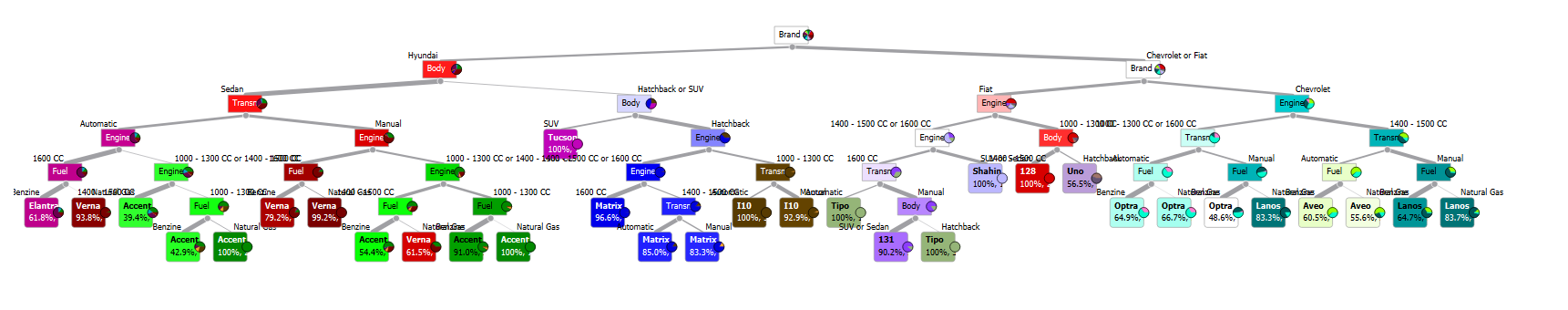
This time I’ll make something less useful and more for the purpose fun and testing the tree model

I’ll try to predict the car model based off only car features (so excluding **Gov**, **Kilometers** and **Price**); I’ll also try removing some features to lower complexity so long as it is within acceptable accuracy

|  |  |  |  |
| --- | --- | --- | --- |
| Features | Accuracy | Nodes | Leaves |
| Brand, body, Engine, Transmission, Fuel, Year, Color | 0.864 | 1795 | 898 |
| Brand, body, Engine, Transmission, Fuel, Year | 0.832 | 485 | 243 |
| Brand, body, Engine, Transmission, Fuel | 0.770 | 61 | 31 |

Included features to performance table

It is clear color gives the most complexity while helping the least in accuracy, removing year is also acceptable to make the tree small enough to fit it in one screenshot



### Neural Network

This time the target will be the year; we will include all the features and only try changing the Regularization (a)

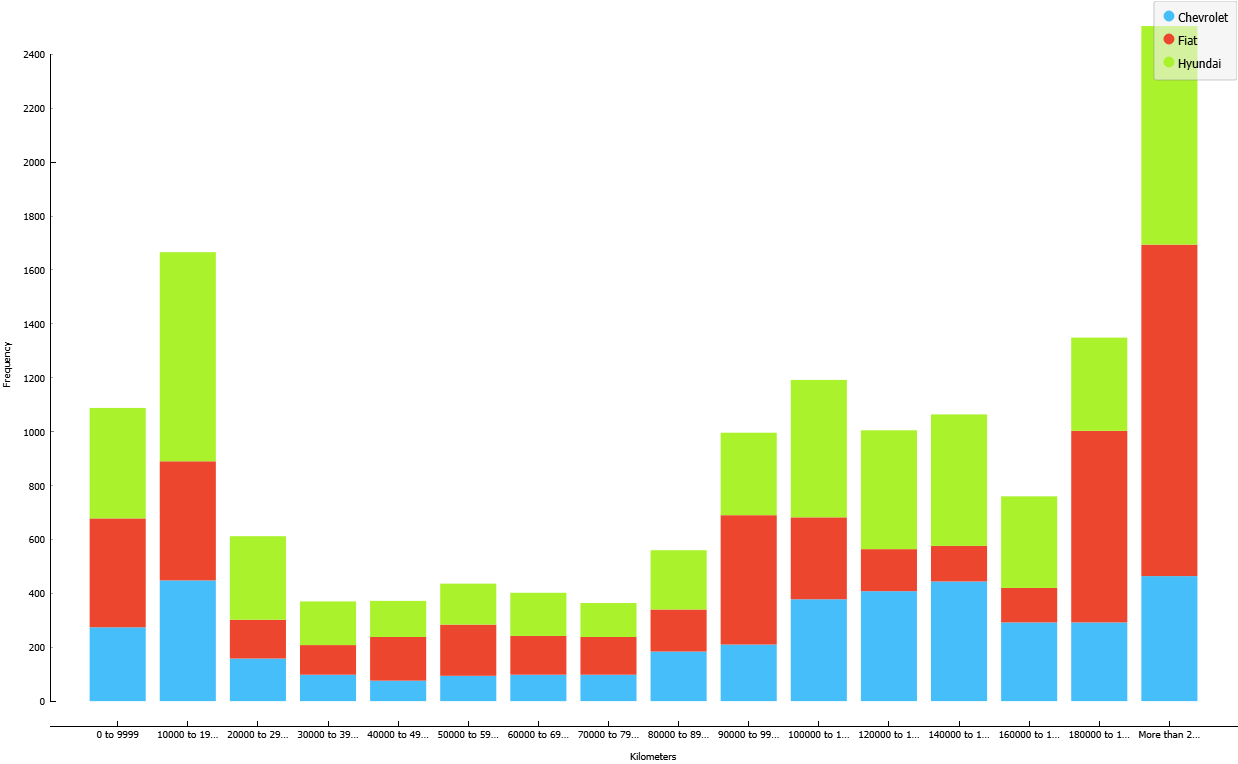
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Regularization | MSE | RMSE | MAE | MAPE | R2 |
| 0 | 93.192 | 9.654 | 7.048 | 0.004 | 0.418 |
| 100 | 54.352 | 7.372 | 5.295 | 0.003 | 0.661 |
| 500 | 17.434 | 4.175 | 3.121 | 0.002 | 0.891 |
| 1000 | 18.757 | 4.331 | 3.299 | 0.002 | 0.883 |

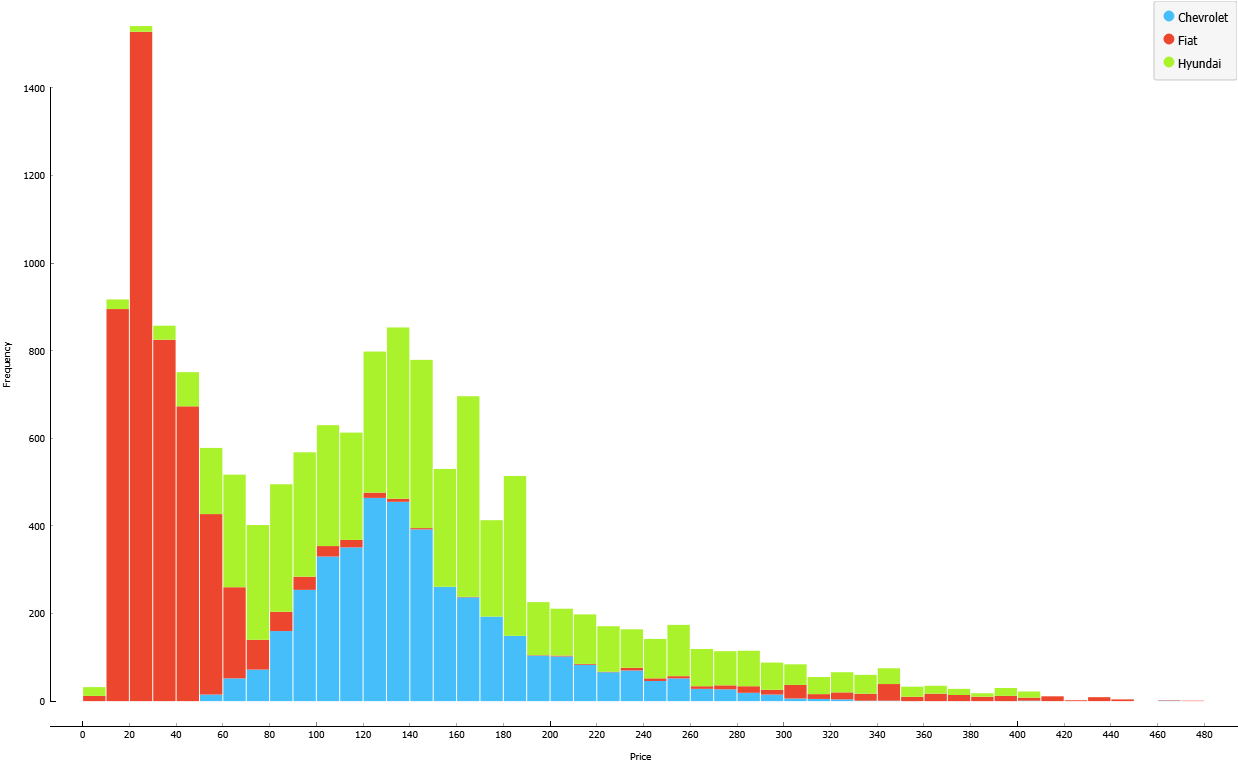
Regularization to performance table

# Visual Data Analysis

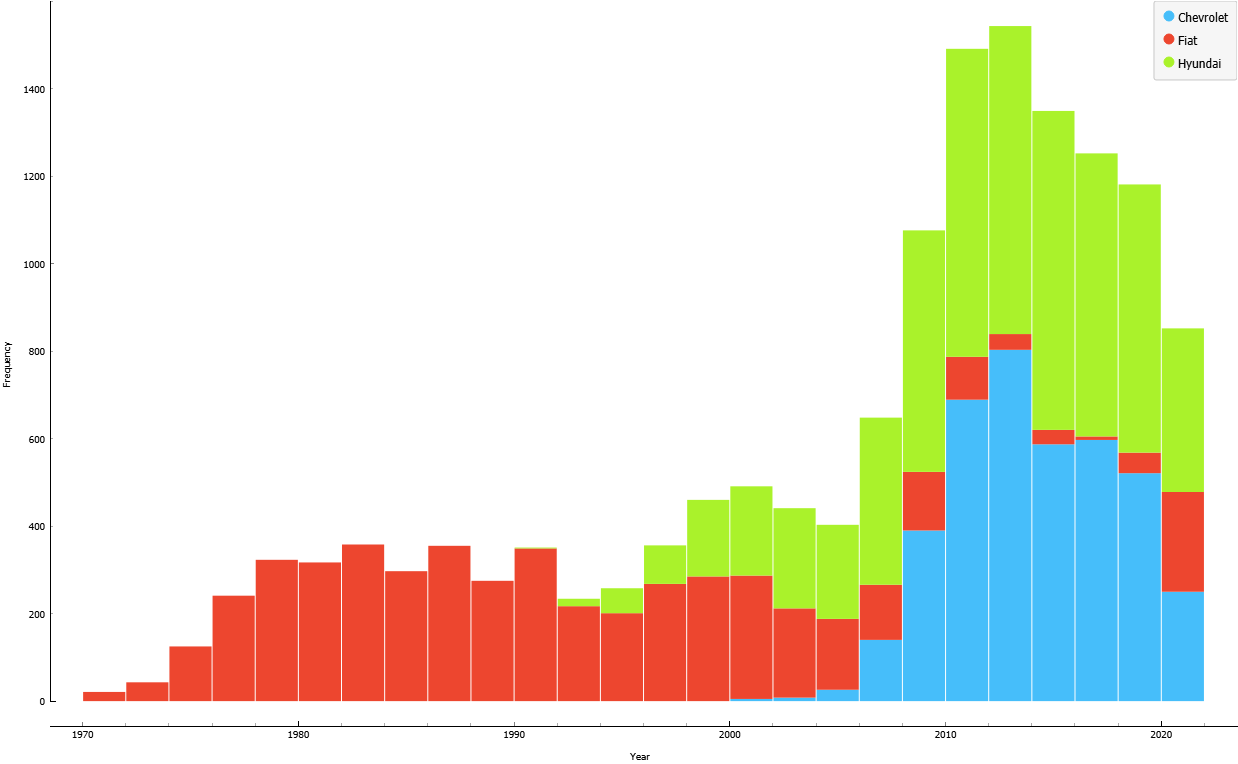
Looking through data visualizations and drawing conclusions from it manually, I’ll mainly be looking at Brand and its relation to other features

First, we’ll look through distributions

We notice the majority of used cars sold are on the high or low end of milage while the middle, seems people can be mostly divided into 2 super classes, those who use cars until their very end before renewing and those who constantly renew cars from the sale price of the old one, Brand is about the same ratio in all of them



Far from a normal disruption, the large majority of cars fall being the ballpark of 200k, Fiat makes up the majority of low-end prices while also having the furthest presence in the highest end (possibly vintage cars), Chevrolet and Hyundai have mostly normal disruption around 130k

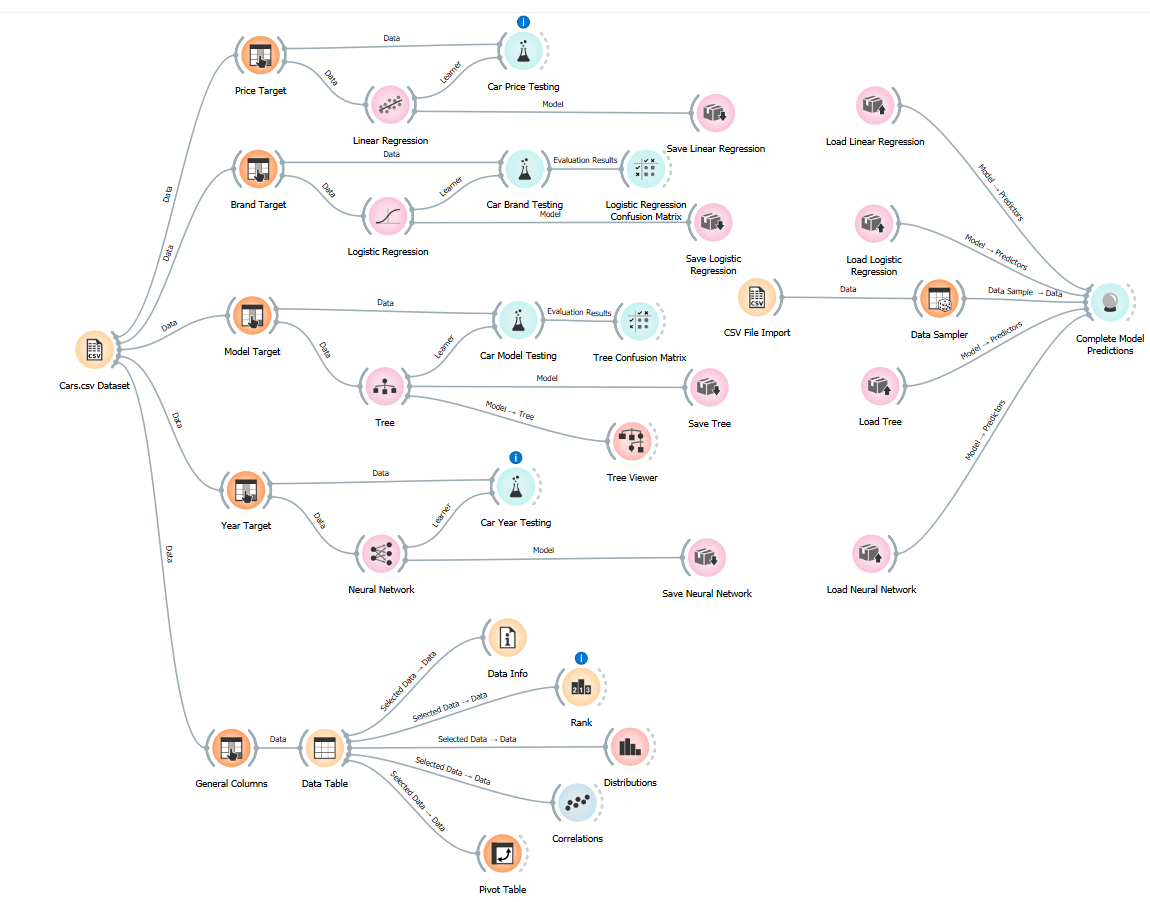
Fiat makes up the majority of vintage and old car presence explaining the variety of price range, 2008 is about the media in which the number of cars after is the same as before

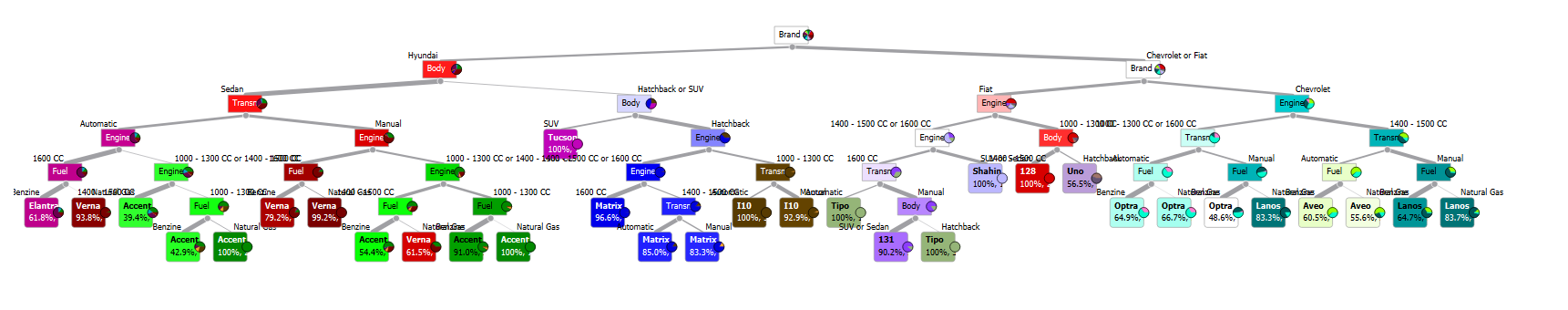
# Conclusion

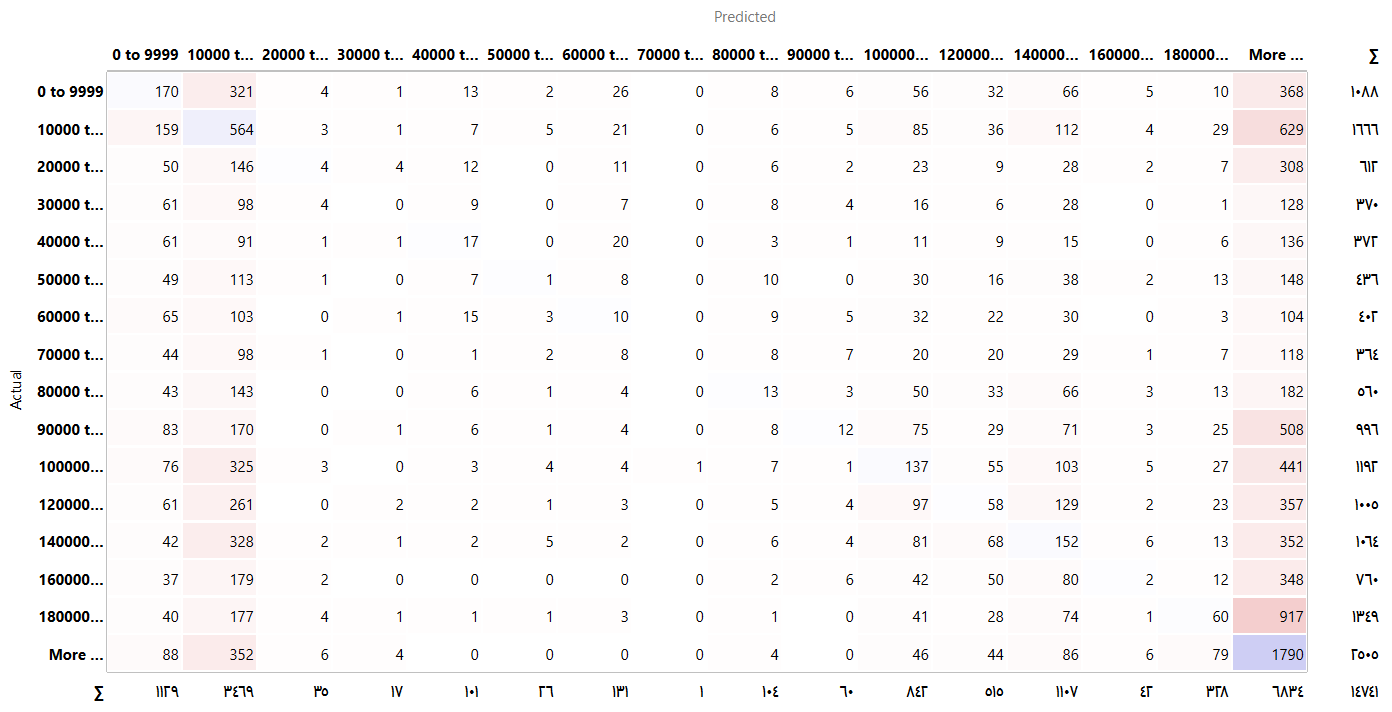
Here are some conclusions I drew about the data laid out in the form of bullet point:

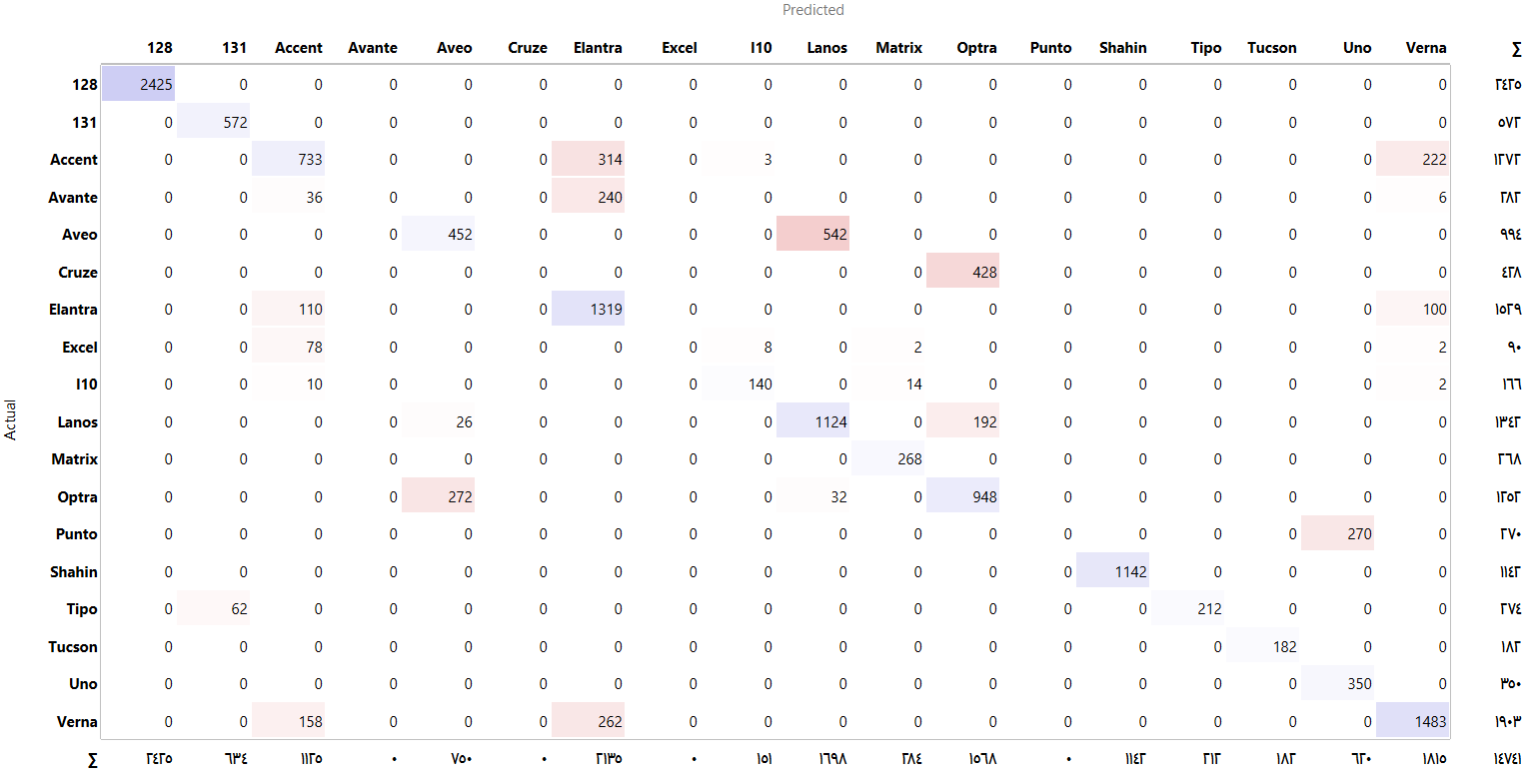
1. As concluded from linear regression **MAE** value, outliers cause about a third of the average error in price prediction, outliers which are typically priced up rather than down and which are likely not exclusive to this specific website, so for analyzing data from a public online marketplace it’s best to either clean out the data from outliers or use less outlier sensitive variables
2. It is best that we keep the milage data numerical since without it we can at best get a 20% accurate logistic model, all features even unrelated ones like color help in getting a more accurate model nonetheless
3. When trying to predict model, color causes more complexity than is worth considering that it can be a shared feature between multiple models
4. The benefit from increasing regularization start to level off after 500 but never stops increasing performance nonetheless
5. The majority of used cars up for sale are either overused or fairly new
6. Fiat makes up the budget car scene while also being in the high-end market, majority of the cars fall under 200k (2 years ago)
7. The biggest variety of available used cars start after **2006**, before that it’s mostly fiat

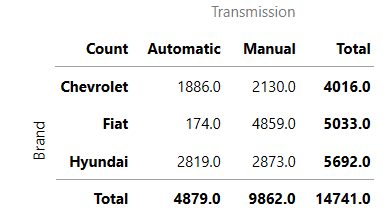
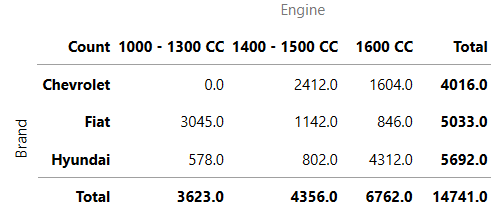
# Screenshots

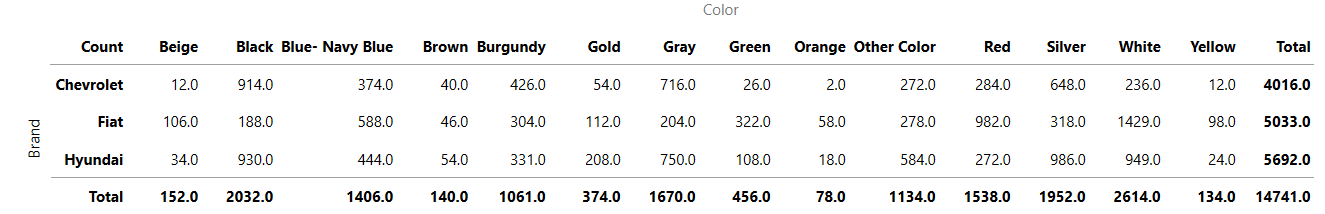
 The complete orange workflow

Model Tree Visualized

Logistic Regression Confusion Matrix

Tree Confusion Matrix

Brand-Engine Pivot Table Brand-Transmission Pivot Table

Brand-Color Pivot Table

The correlation between price and year