**Data Science Challenge**

Dear candidate,

You have made a good impression in the first steps in our recruitment process and we would like to progress with you to the next part of our assessment.

The purpose of the data science challenge is to let you demonstrate the way you think and work.

This exercise should take around 2 to 4 hours to be completed but you have 48 hours to return your solution.

Please take a look at the dataset in the file “Auto1-DS-TestData.csv” (see <https://archive.ics.uci.edu/ml/datasets/Automobile> for information on the features and other attributes) and answer the following questions:

Question 1 (10 Points)

List as many use cases for the dataset as possible.

Question 2 (10 Points)

Auto1 has a similar dataset (yet much larger...)

Pick one of the use cases you listed in question 1 and describe how building a statistical model based on the dataset could best be used to improve Auto1’s business.

Question 3 (20 Points)

Implement the model you described in question 2 in R or Python. The code has to retrieve the data, train and test a statistical model, and report relevant performance criteria.

When submitting the challenge, send us the link for a Git repository containing the code for analysis and the potential pre-processing steps you needed to apply to the dataset. You can use your own account at github.com or create a new one specifically for this challenge if you feel more comfortable.

Ideally, we should be able to replicate your analysis from your submitted source-code, so please explicit the versions of the tools and packages you are using (R, Python, etc).

Question 4 (60 Points)

A. Explain each and every of your design choices (e.g., preprocessing, model selection, hyper parameters, evaluation criteria). Compare and contrast your choices with alternative methodologies.

B. Describe how you would improve the model in Question 3 if you had more time.

**We are looking forward to receiving your solution!**

1. There is many use case for the dataset:
   1. Price forecasting
   2. Loss forecasting
   3. Symbolling Forecasting
   4. Clustering by symbol and normalized loss
   5. Understand If the data missing is random or not.
   6. Segmentation cars (by brand, model, engine, ecc...)
   7. Data visualization
2. I chose to describe the use case “a”, because I think it will be a good attribute for Auto1. Having a good price prediction is very important, especially when your activity is trading. This attribute with normalized losses is very important, and I would like to focus on them.

The gol for a company like Auto1 Group is pay less as possible, preventing disappointments; Build a statistical model can be used to have an high accuracy of the price forecasting.

This can give you a better evaluation of the auto’s purchases and auto’s sales. The forecasting process is often underestimated, and are used improper models. The market of today is biased by a lot of features. The cars market is changed, just think that the car’s brand launch a new model every year, but before was 1 every four year. In order to improve the business of Auto1, I think there are more than one benefit to have a good price prediction: the first is for short term, where we can estimate a car’s price for every day and we can update the daily price of purchase/sell on the website. To do this, I think is necessary use some variable with an high recording frequency (real time/hourly, ecc..). For the long term is very important estimate the assets in the business plan (especially for the storage). This is not less important. A good evaluation of price for the long term period need different variable, and different risk margin. But is necessary for a company use this dataset for define a good strategy of business.

1. I decide to use random forest, because is easy to understand and is very efficient.
2. The first step was detected if it were data missing inside the dataset. After seen the percentage of row was high (like 25.5%), I decided to proceed using kNN technique to imputate data missing. Done this, I proceed to write a code of a Random forest (Out of bag). I split in train and test set the original dataset using a percentage of 70% of random sample. I decide to loop the code between a vector of possible dimension of predictors with maximum length the number of all variable. I stored the mse in a vector rf.err using a mean function. In every position of this vector there are the mean error of the random forest for each number of parameters. I plotted that, and I detect that the lower error rate is with 9 parameters. Than I runned the random forest for the selected parameters, and I predicted the test set, with a MAE of 8%. As we can see in the plot of the random forest, the mse is stabilized than 250 trees.
3. If I had enough time, I tried to use the Principal component analysis to reduce the number of features because i’ve seen with a corplot that some features are correlated. So it can be a good way to mitigate case of correlation inside the forest even if the random forest (OOB) is used to create uncorrelated trees. Also I’d liked to detecte if it were outlier, and use some technique (log price/normalization, ecc..) to mitigate it influence.