IE0005
Mini Project:
Laptop Specs

TABLE OF CONTENTS

01

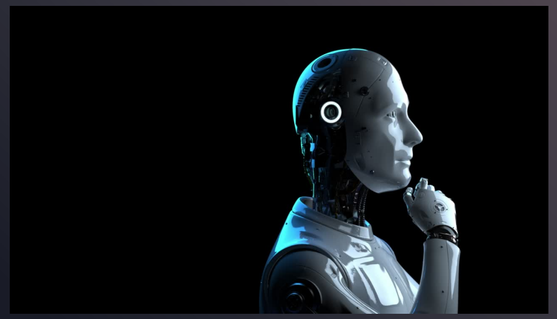
DATASET,OBJECTIVE AND OBSERVATIONS

02

ML TECHNIQUES AND FURTHER EXPLORATION

03

OUTCOME & CONCLUSIONS



What is the dataset?

LAPTOP SPECS: TECH SPECS AND PERFORMANCE DATA FOR A COMPREHENSIVE LAPTOP DATASET

About the dataset

1

WHERE?

Kaggle

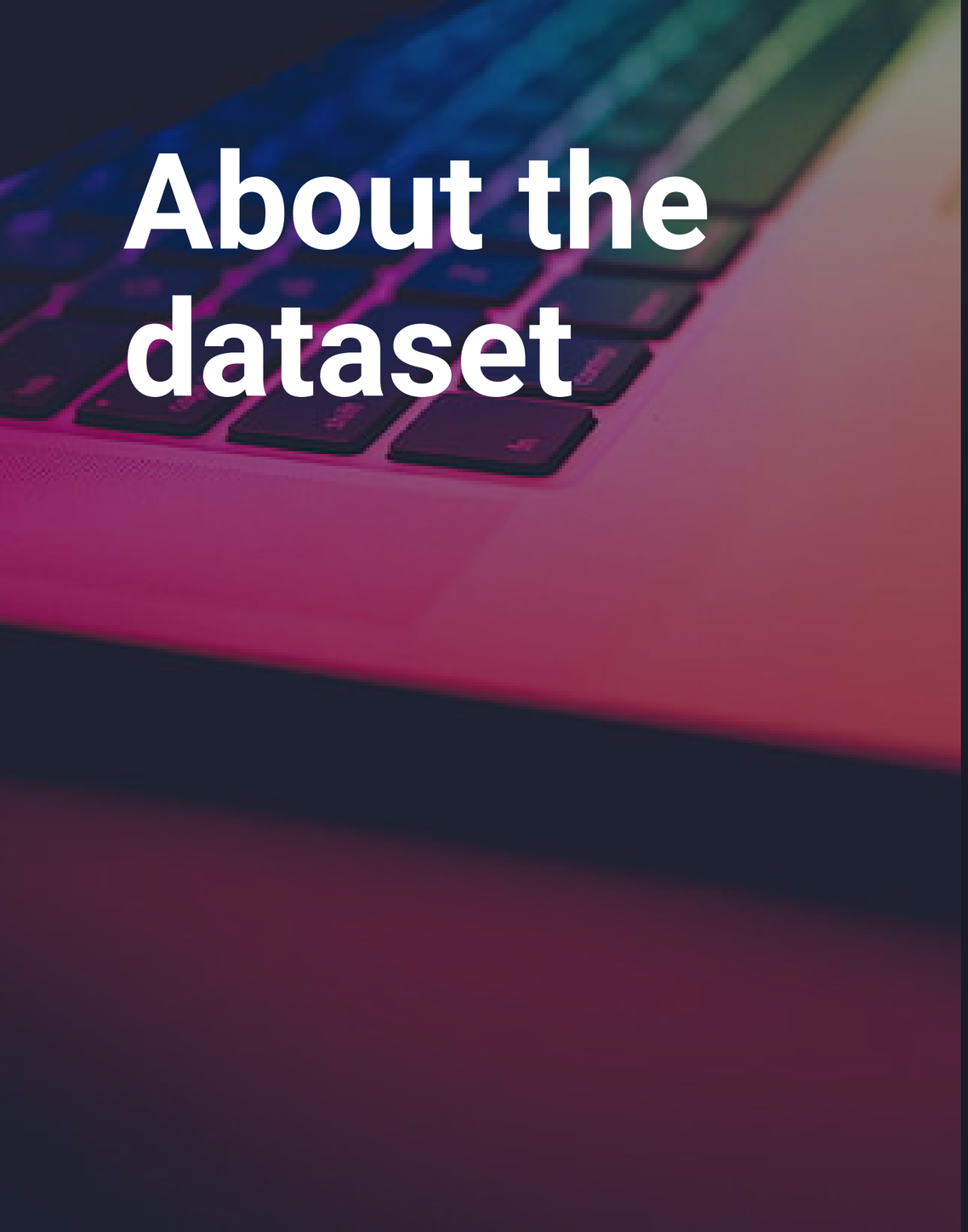
2

WHAT?

Laptop features

- 1251 values (after cleaning)
- 19 unique companies
- Usability index : 9.41

https://www.kaggle.com/datasets/adityamishraml/laptops



About the dataset

3

WHEN?

updated once a month

4

HOW?

python based web scraper

WHY?

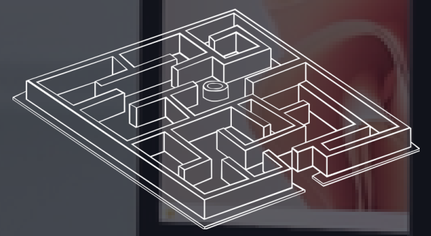
PROBLEM STATEMENT

WHAT ARE THE MOST SIGNIFICANT COST DRIVERS IN THE PRICING OF A LAPTOP AND HOW CAN ONE PREDICT A GOOD PRICE OF A LAPTOP?

What makes this problem interesting?

01

Complex relations



02

Useful knowledge gained

03

Has real world application

DATA CLEANING

```
laptopdata.drop('indx', axis=1, inplace=True)
laptopdata.drop('resolution_width', axis=1, inplace=True)
```

DROPPED IRRELEVANT COLUMNS

- IPS PANEL
- RESOLUTION HEIGHT
- RESOLUTION WIDTH
- MEMORY
- OPS SYSTEM

REASON?

- LACK OF RELEVANCE IN PRICE COMPARISION
- IPS PANEL - VERY COMMON OCCURENCE
- MEMORY - ADDITIONAL STR CHARACTERS ATTACHED
- OPS SYSTEM - 0 CORRELATION

Memory
256GB SSD
128GB SSD
128GB SSD + 1TB HDD
1TB HDD

DATA CLEANING

CLEANED SOME ROWS

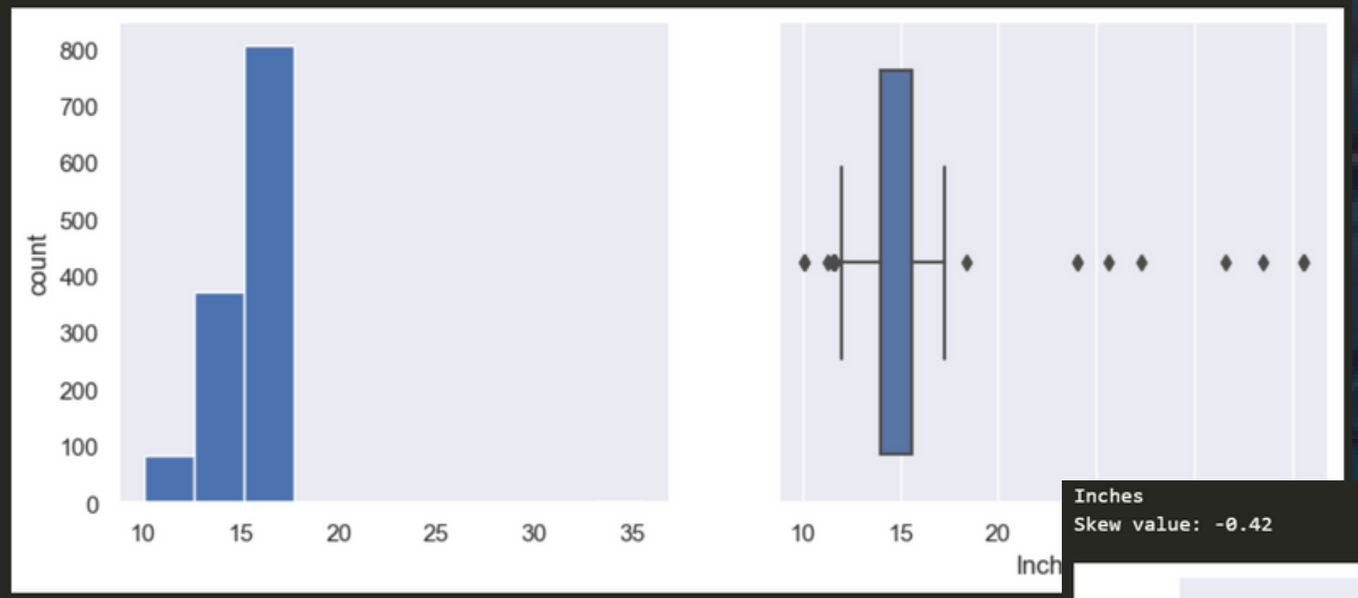
- LAPTOP SIZE > 17.3 INCHES
- LAPTOP RAM > 32 GB
- LAPTOP WEIGHT > 5 KG

```
i = 0
l = list()
for value in iter(laptopdata['Inches']):
    if(value > 17.3):
        to_be_dropped = laptopdata.iloc[i]
        l.append(i)
    i += 1
for index_label in l:
    laptopdata.drop(index=index_label, axis=0, inplace=True)
```

REASON?

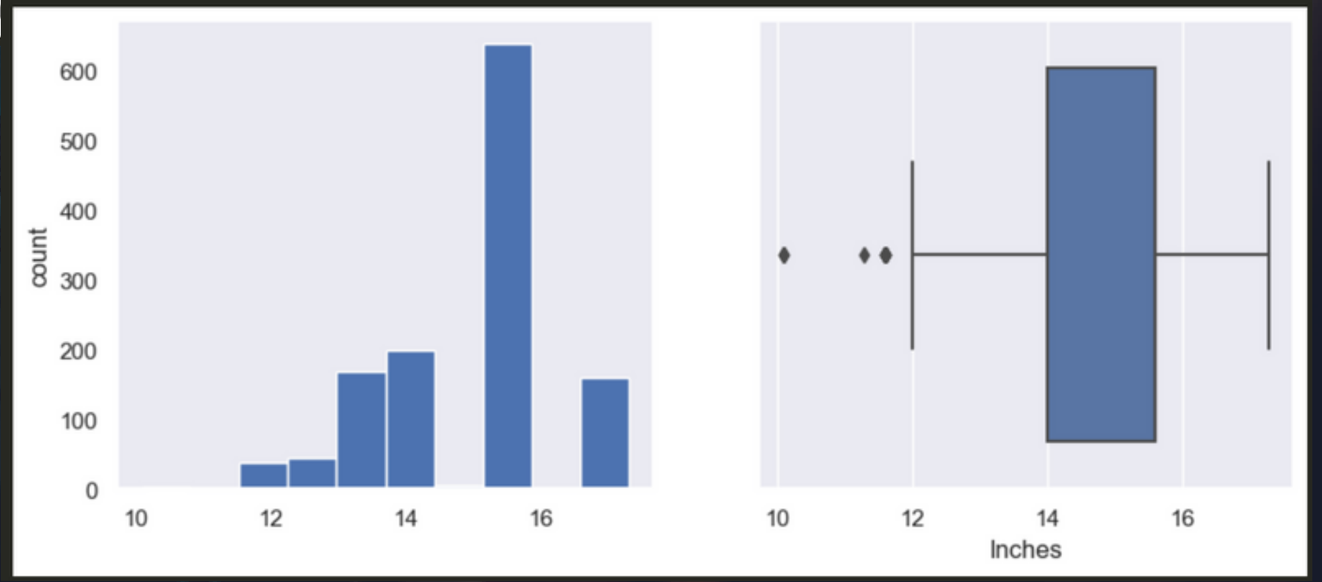
- REMOVE OUTLIERS
- REDUCE THE SKEWNESS OF THE DATA TO BETTER FIT THE MODEL

Inches
Skew value: 4.05



AFTER

Inches
Skew value: -0.42



BEFORE

BASIC STATISTICS

```
laptopdata.nunique()
```

```
Company          19
TypeName          6
Inches           24
touchscreen       2
retinadisplay     2
cpu_brand         3
cpu_name         99
cpu_speed        24
Ram              10
hdd               6
ssd              11
flashstorage      7
hybrid            3
gpu_brand         4
gpu_name        106
Weight_kg       177
Price           764
dtype: int64
```

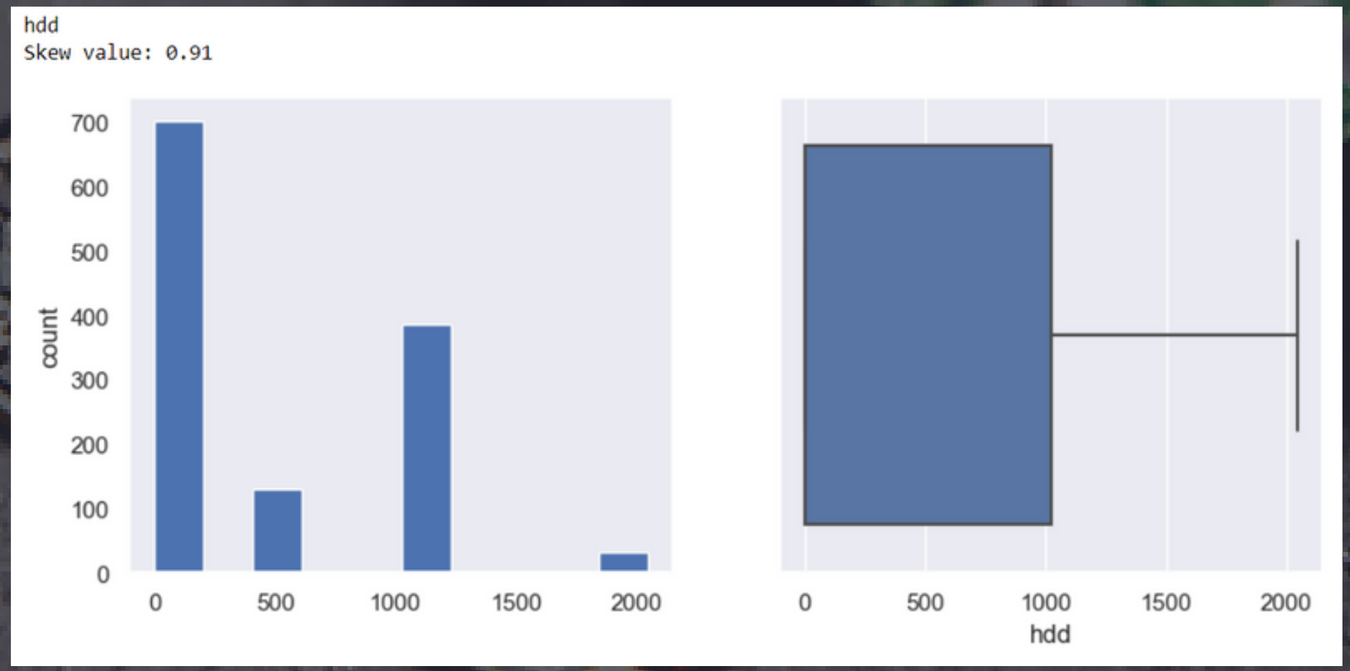
	count	unique	top	freq	mean	std	min
cpu_speed	1272.0	NaN	NaN	NaN	2.298192	0.50768	0.9
Ram	1272.0	NaN	NaN	NaN	8.46305	5.566582	1.0
hdd	1272.0	NaN	NaN	NaN	423.132075	527.514262	0.0
ssd	1272.0	NaN	NaN	NaN	182.534591	185.63666	0.0
flashstorage	1272.0	NaN	NaN	NaN	4.591195	30.578213	0.0
hybrid	1272.0	NaN	NaN	NaN	9.254717	95.876648	0.0
Weight_kg	1272.0	NaN	NaN	NaN	2.076211	0.809869	0.0
Price	1272.0	NaN	NaN	NaN	59902.143082	37297.683852	9271.0

- MEAN
- STD
- MIN
- RELATION BETWEEN TOP AND UNIQUE

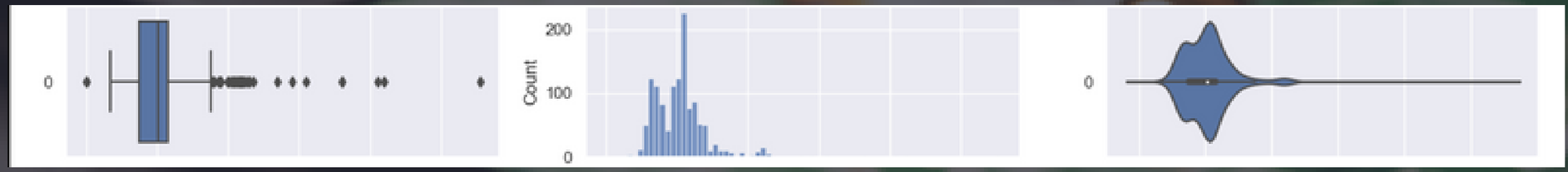
- We notice touchscreen and retina display have only 2 unique values either 0 or 1.

EXPLORATORY DATA ANALYSIS

- We chose to sort the columns into numerical and categorical variables.
- Based on the following visual data we came to some further conclusions :
  - Using histplot and boxplot
- example:

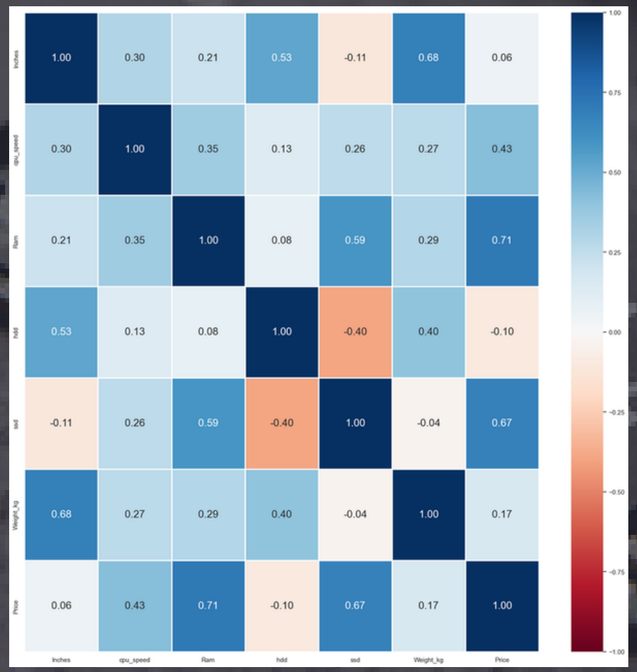


Observations?

- Plotting bar plots to understand distribution of categorical variables
- Next using boxplots, histplots and violin plots for the numerical data
  - 

Observations?

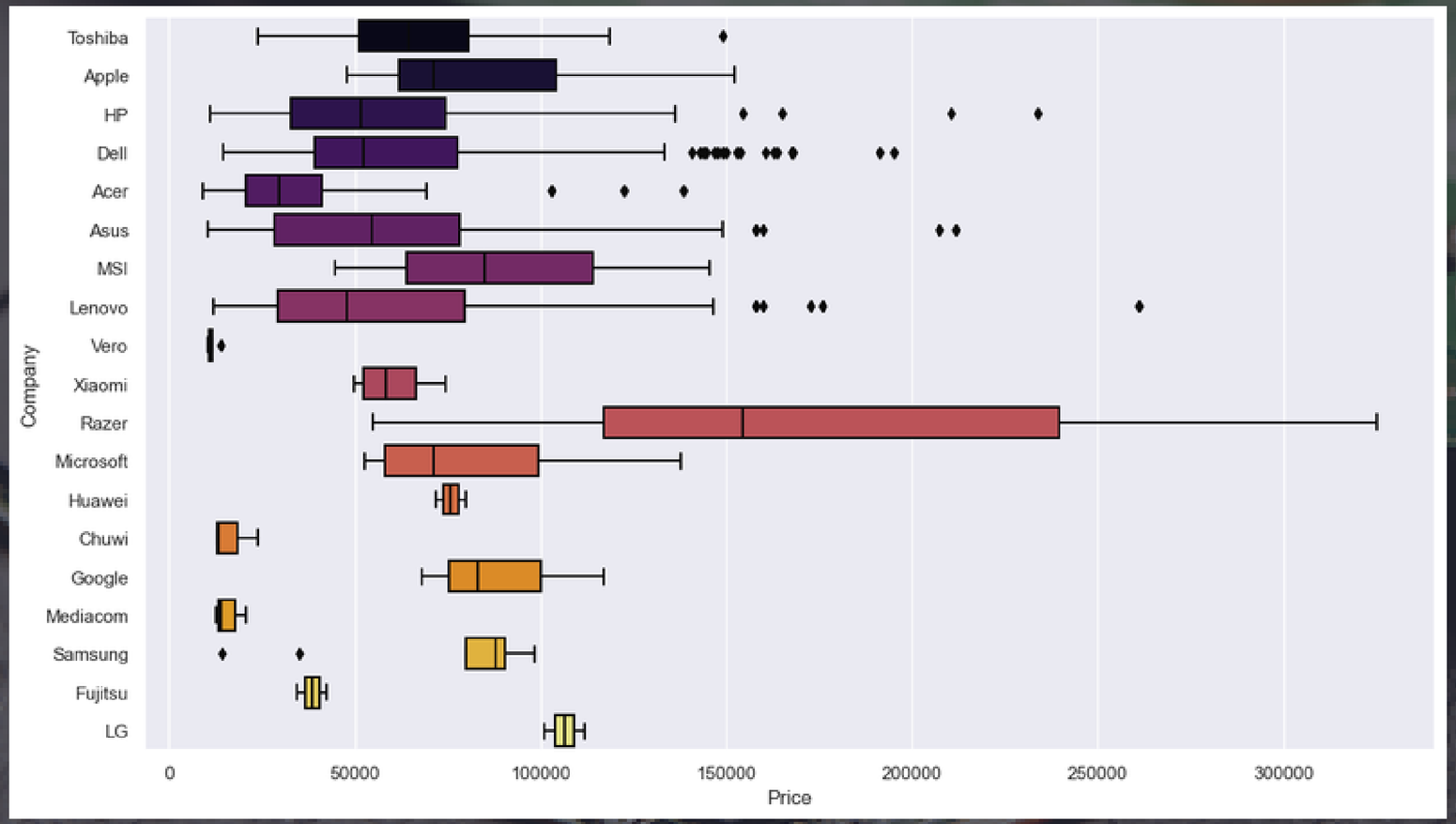
- Next we evaluate the correlation matrix and heatmap



Observations?

- Highest correlation exists between price and ram = 0.71
- As well as ssd as price = 0.67

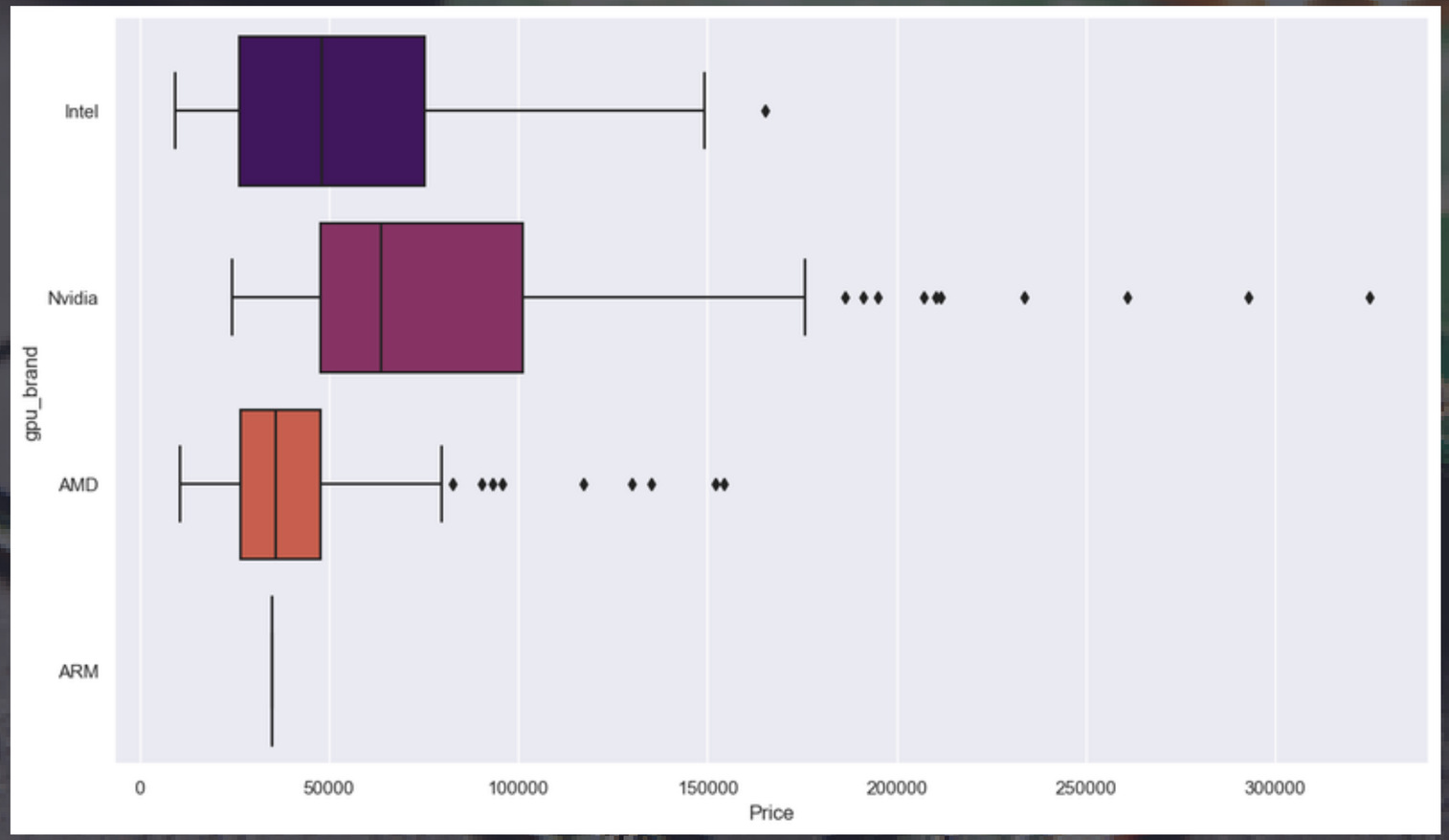
- The following boxplot of Company vs Price provides more insight between laptop features and price



Observations?

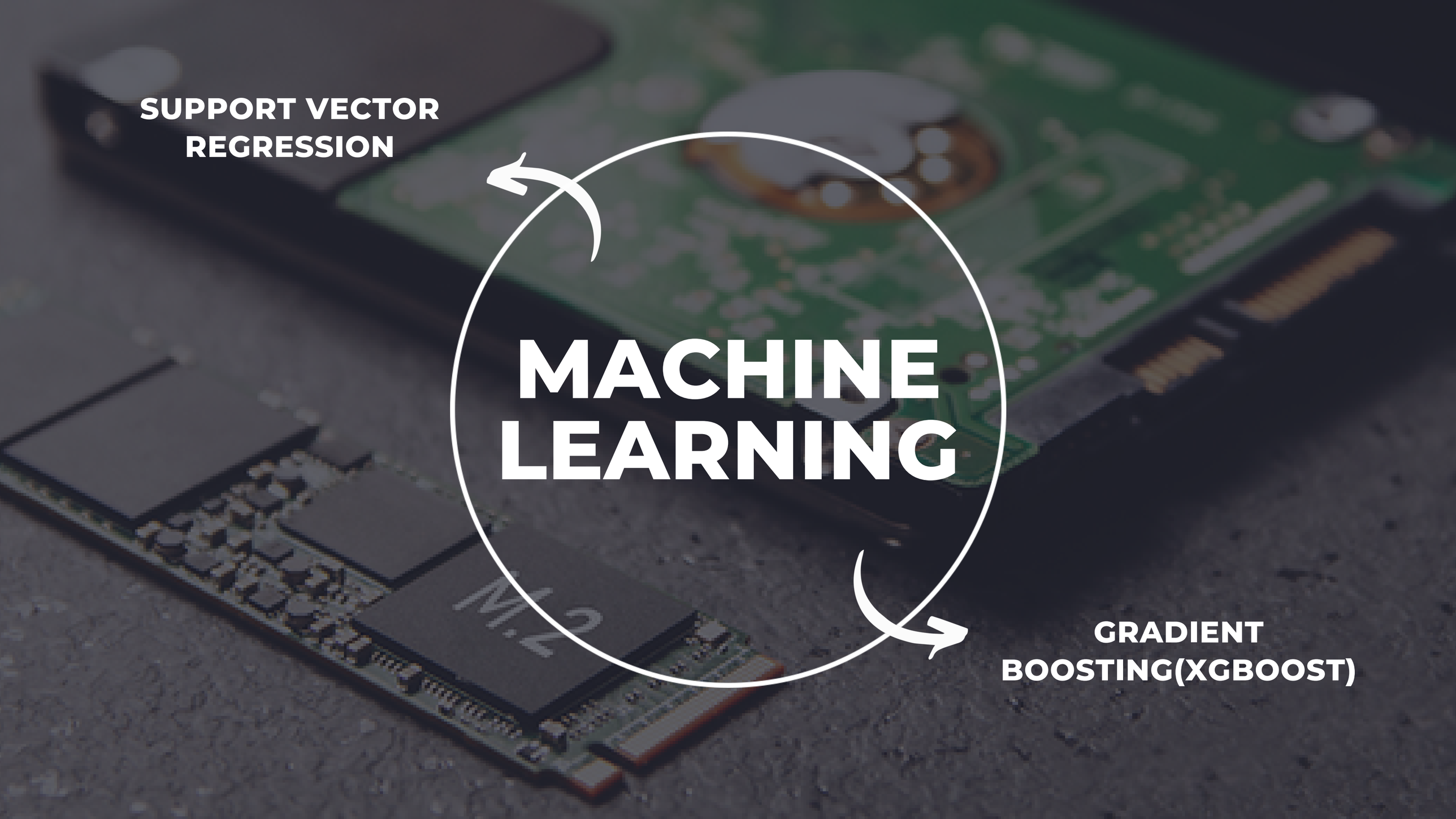
- Companies like Lenovo,Asus,HP and Dell have the most number of outliers.
- Razer makes the most expensive laptops : which are gaming laptops

- The following boxplot of gpu_brand vs Price provides more insight between laptop features and price



Observations?

- Nvidia generally has the most expensive gpu's as well as the largest no. of outliers.


SUPPORT VECTOR REGRESSION
MACHINE LEARNING
GRADIENT BOOSTING(XGBOOST)

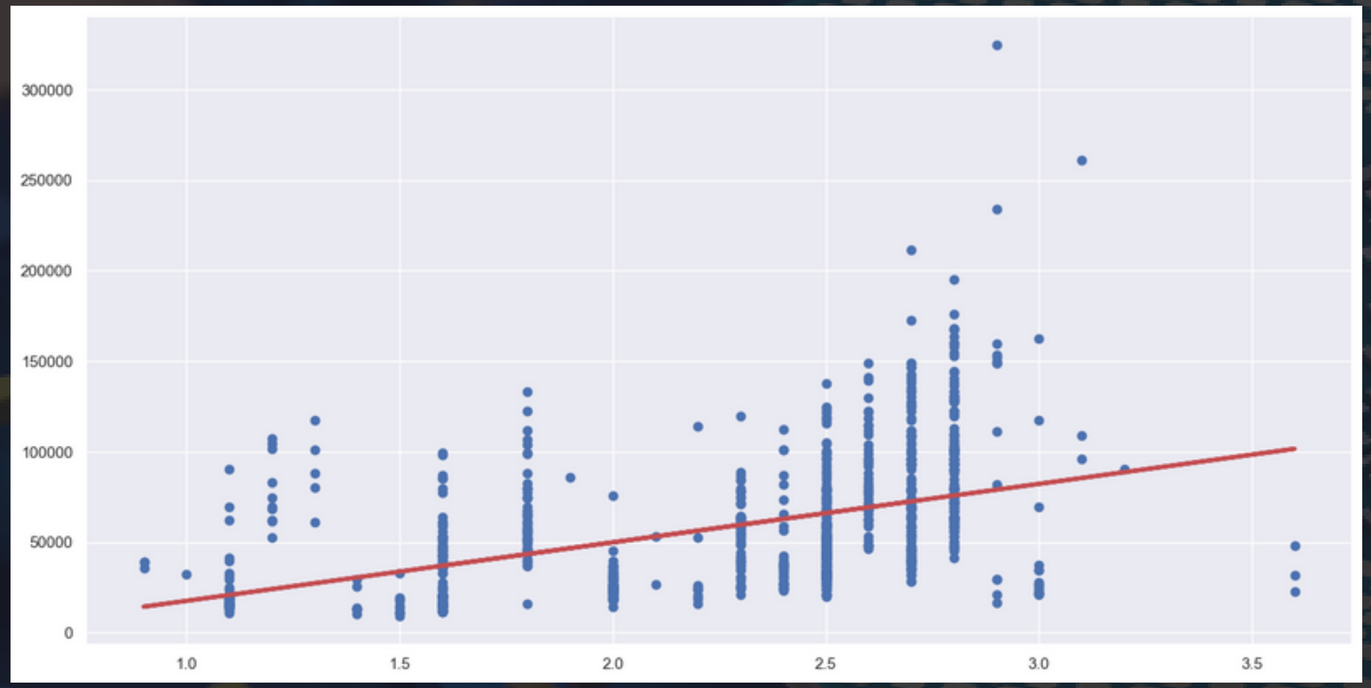
Regression

- **Regression Models**
  Uni Variate Linear Regression
  Multi Variate Support Vector Regression

- **Parameters used**
  RAM, CPU Speed, Weight, SSD against Price
  Default Radial Basis Function (RBF) Kernel used

- **Why use Regression?**
  Powerful tool for Prediction
  Numeric Data
  SVR- Great with non linear processes, Easy Implementation and High prediction accuracy

Working and Analysis



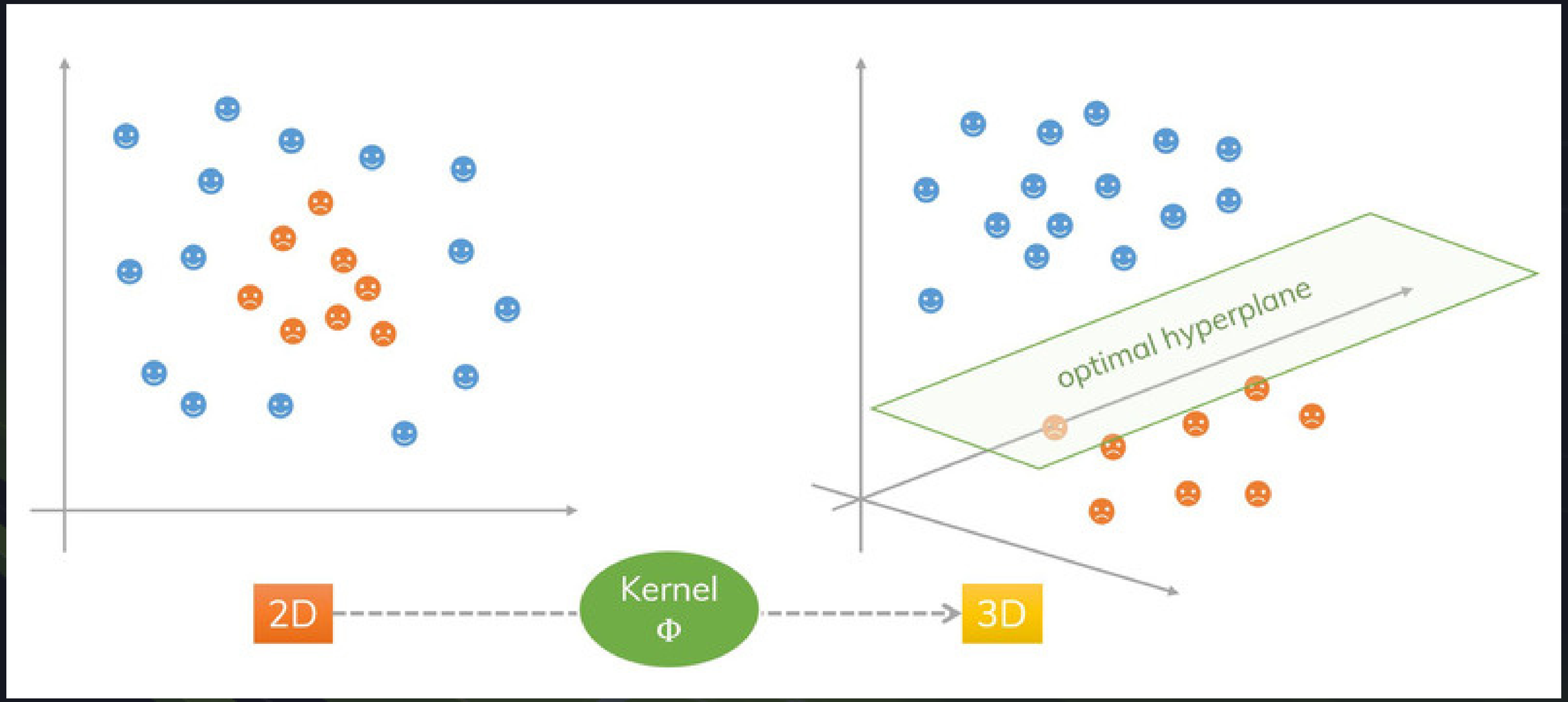
Univariate Linear Regression

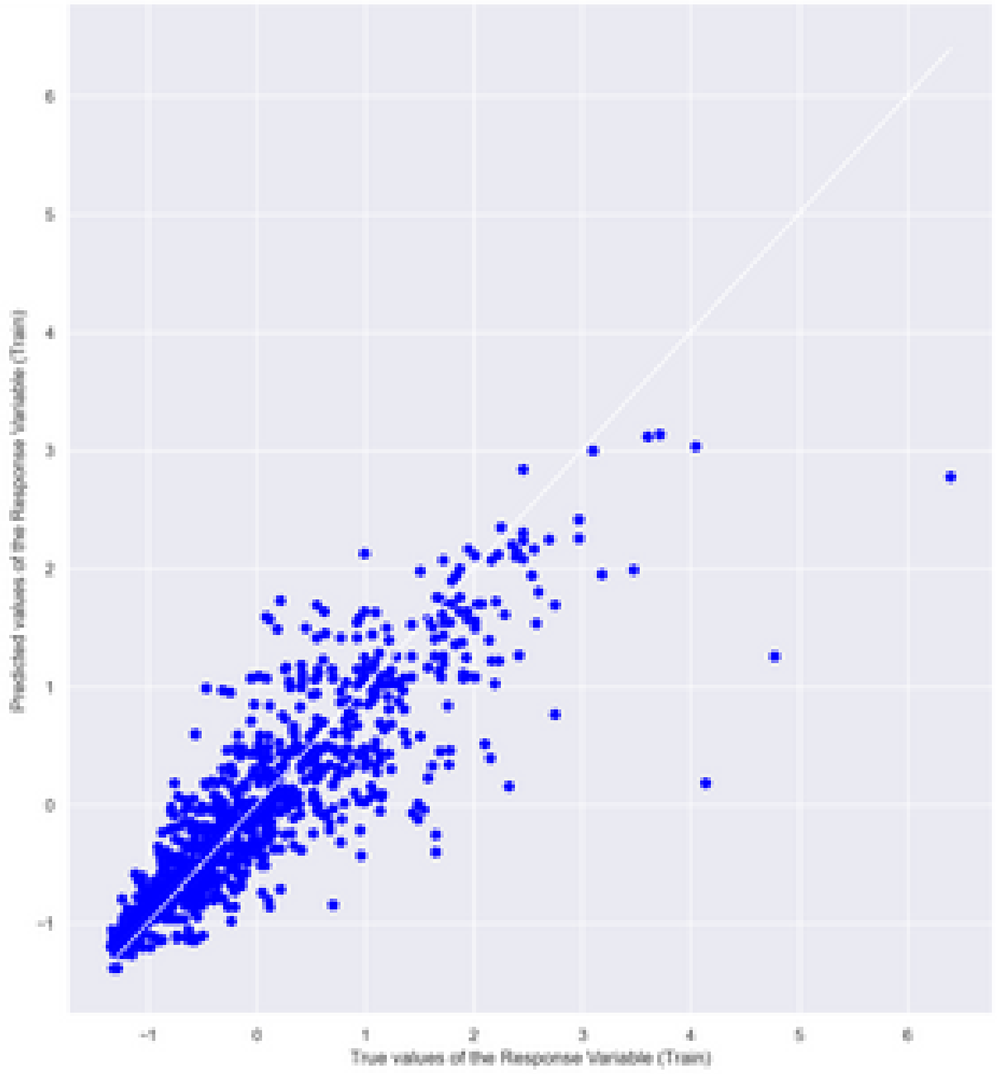
1- Ran Univariate Linear Regressions.
Found some relation but not exactly linear. Low R Square Values

2- Tried Multi Variate Linear Reg.
Gave low R Square Value- Around 0.45

3- Searched for more methods for non-linear relations.

What is Support Vector Regression





4- Chose SVR (Multi Variate) for reasons mentioned earlier (Shown Above)

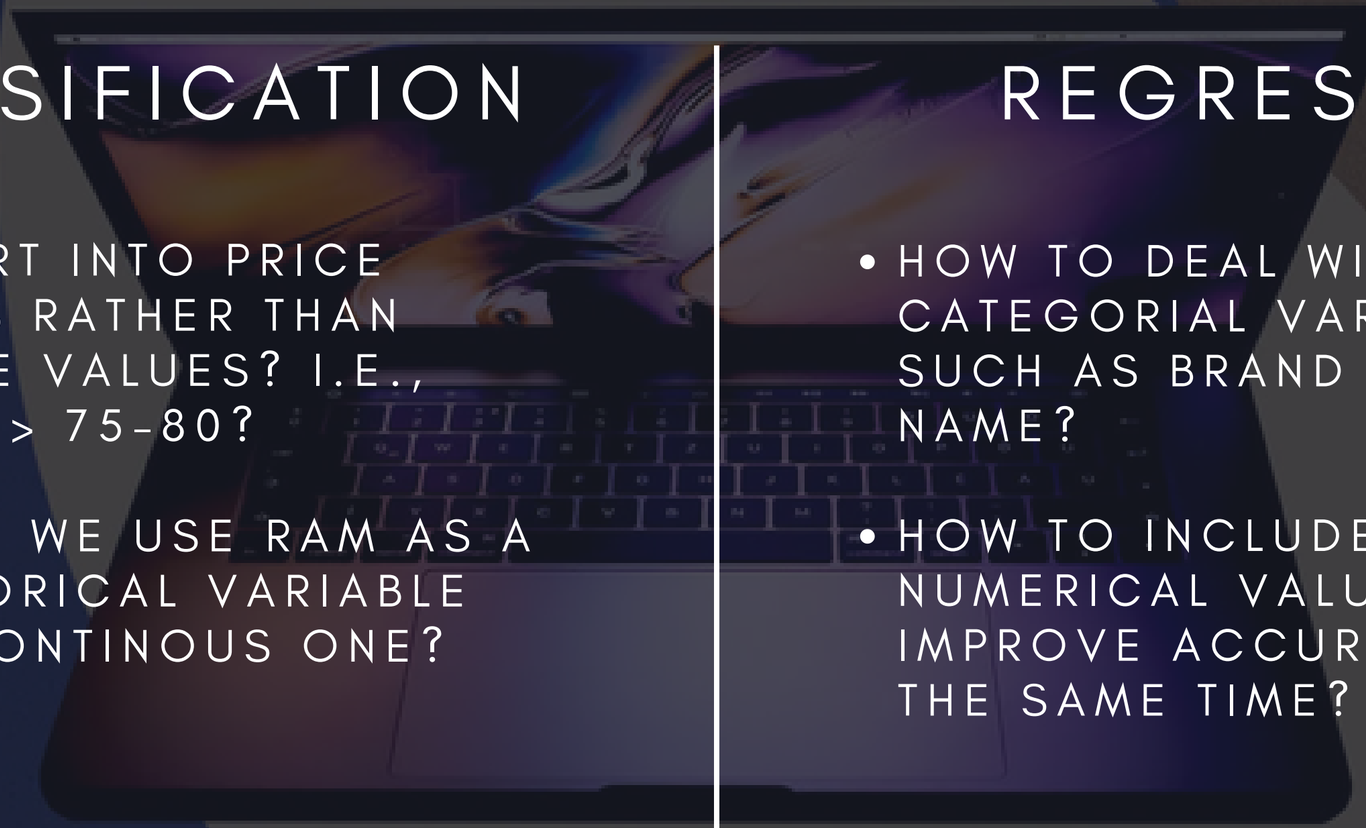
```
Goodness of Fit of Model        Train Dataset
R^2 Score       : 0.7445903476798588
Mean Squared Error (MSE)        : 0.25540965232014123

Goodness of Fit of Model        Test Dataset
R^2 Score       : 0.7133103833597538
Mean Squared Error (MSE)        : 0.28668961664024634
```

5- SVR gave relatively high R Square Values
(Around 0.7)
Found a non linear relationship as 0.7 is a good score

	indx	Price	PredPrice	Error
234	235	69211	62597.553618	9.555485
348	349	74539	89930.801666	20.649327
631	632	37243	35532.500755	4.592807

6- Tried predicting price using trained model.
Error between predicted and actual price is somewhat acceptable
Justified to use SVR and RBF- Helps in achieving outcome of predicting price

PROBLEM TYPE



CLASSIFICATION

- CONVERT INTO PRICE RANGES RATHER THAN PRECISE VALUES? I.E., 77,819 => 75-80?
- SHOULD WE USE RAM AS A CATEGORICAL VARIABLE OR A CONTINOUS ONE?

REGRESSION

- HOW TO DEAL WITH CATEGORIAL VARIABLES SUCH AS BRAND AND GPU NAME?
- HOW TO INCLUDE NON-NUMERICAL VALUES AND IMPROVE ACCURACY AT THE SAME TIME?

GRADIENT BOOSTING

PEAK

most optimizsed

STEP & SLOPE

iterative step

image retrieved from: https://towardsdatascience.com/a-beginners-guide-to-xgboost-87f5d4c30ed7

GRADIENT BOOSTING

XGBOOST

Gradient Boosting specifically is an approach where new models are trained to predict the residuals (i.e errors) of prior models. I've outlined the approach in the diagram below.

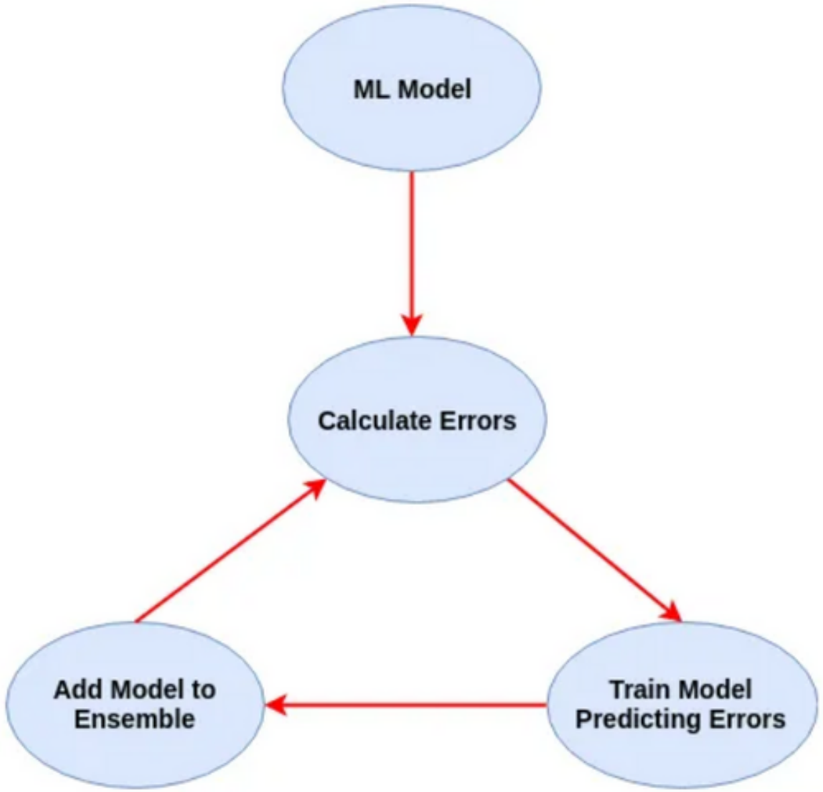


image retrieved from: https://towardsdatascience.com/a-beginners-guide-to-xgboost-87f5d4c30ed7

XGBOOST

WORKING

```
dtrain_reg = xgb.DMatrix(X_train, y_train, enable_categorical=True)
dtest_reg = xgb.DMatrix(X_test, y_test, enable_categorical=True)
```

STEP 1

preparation of DMatrix

```
evals = [(dtest_reg, "validation"), (dtrain_reg, "train")]
model = xgb.train(
    params=params,
    dtrain=dtrain_reg,
    num_boost_round=10000,
    evals=evals,
    verbose_eval=50,
    # Activate early stopping
    early_stopping_rounds=50
)
```

STEP 2

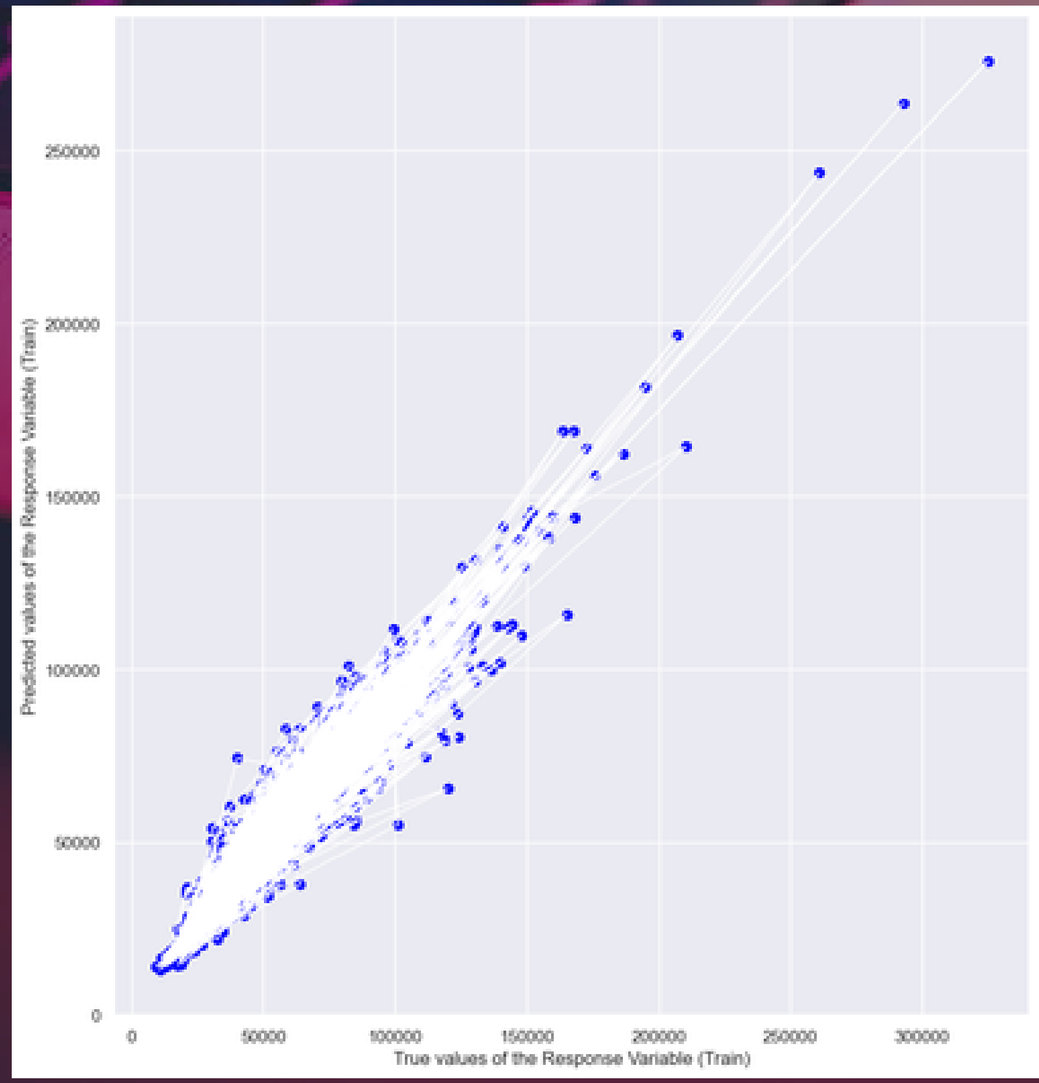
train and verify iterative efficiency

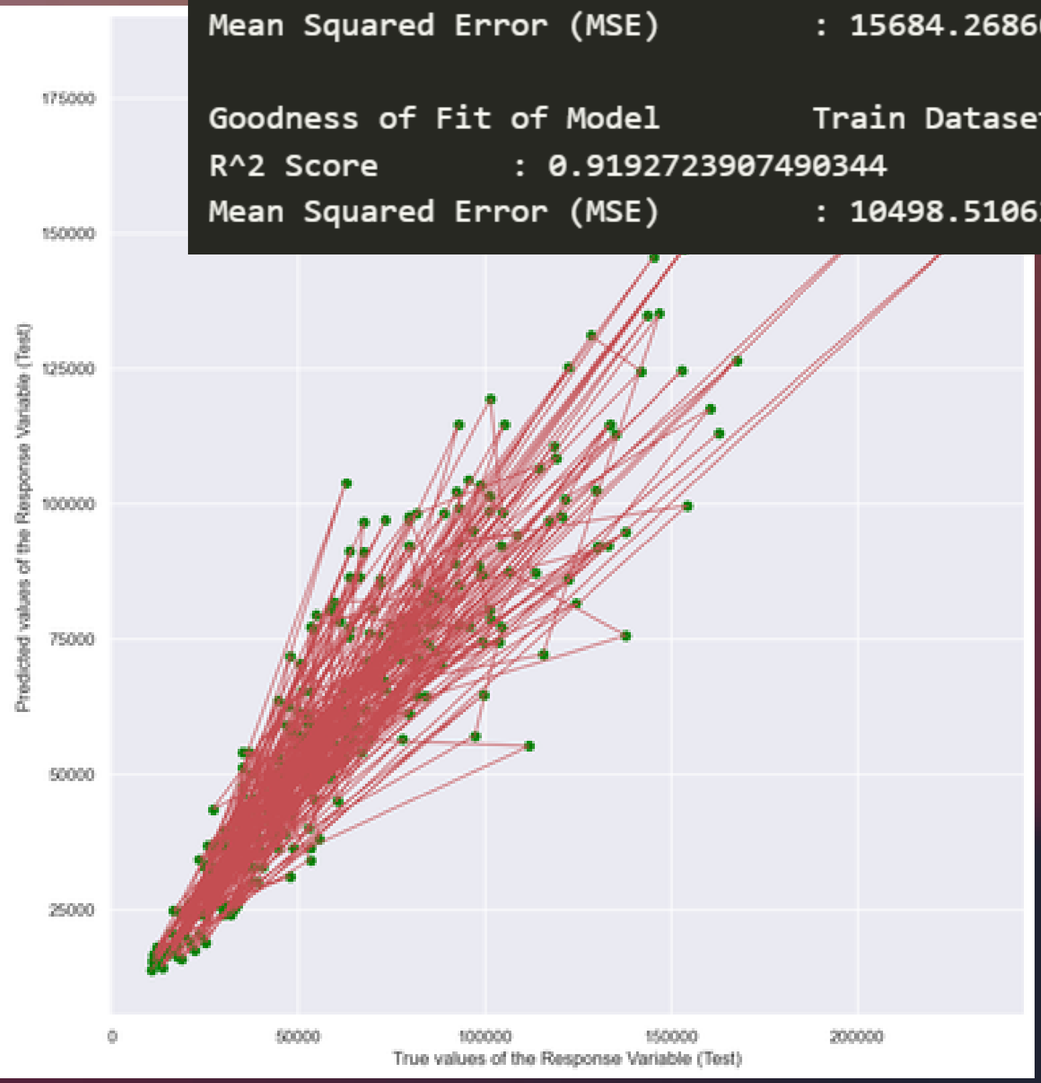
https://towardsdatascience.com/a-beginners-guide-to-xgboost-87f5d4c30ed7

XGBOOST

ANALYSIS

```
RMSE of the base model: 15684.269

Goodness of Fit of Model        Test Dataset
R^2 Score       : 0.8280172253211489
Mean Squared Error (MSE)        : 15684.268667416705

Goodness of Fit of Model        Train Dataset
R^2 Score       : 0.9192723907490344
Mean Squared Error (MSE)        : 10498.51063841628
```

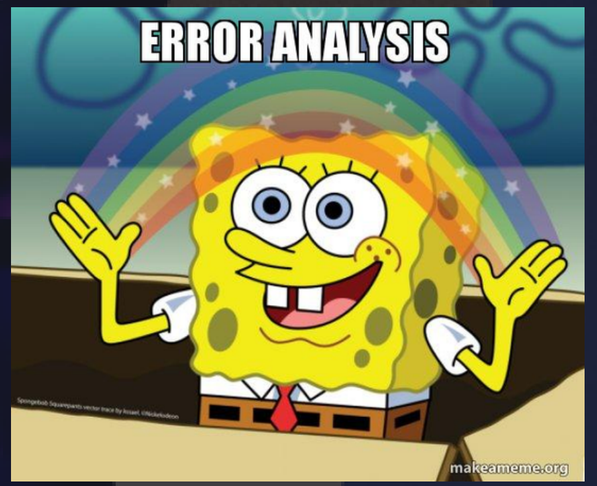




TRAIN

TEST

OUTCOME & CONCLUSIONS



COST DRIVERS: RAM & SSD

PREDICTION ACCURACY: 77-82%

MEAN ERROR IN PREDICTION: 15600 - 17500 INR(250 SGD)

MODEL GOOD?

References

- https://www.datacamp.com/tutorial/xgboost-in-python
- https://www.simplilearn.com/tutorials/data-science-tutorial/svm-in-r
- https://www.analyticsvidhya.com/blog/2020/03/support-vector-regression-tutorial-for-machine-learning/