Laptop Price Prediction for SmartTech Co.

ML capstone Project

Prashantsagar Uppara

Contents

Project Charter

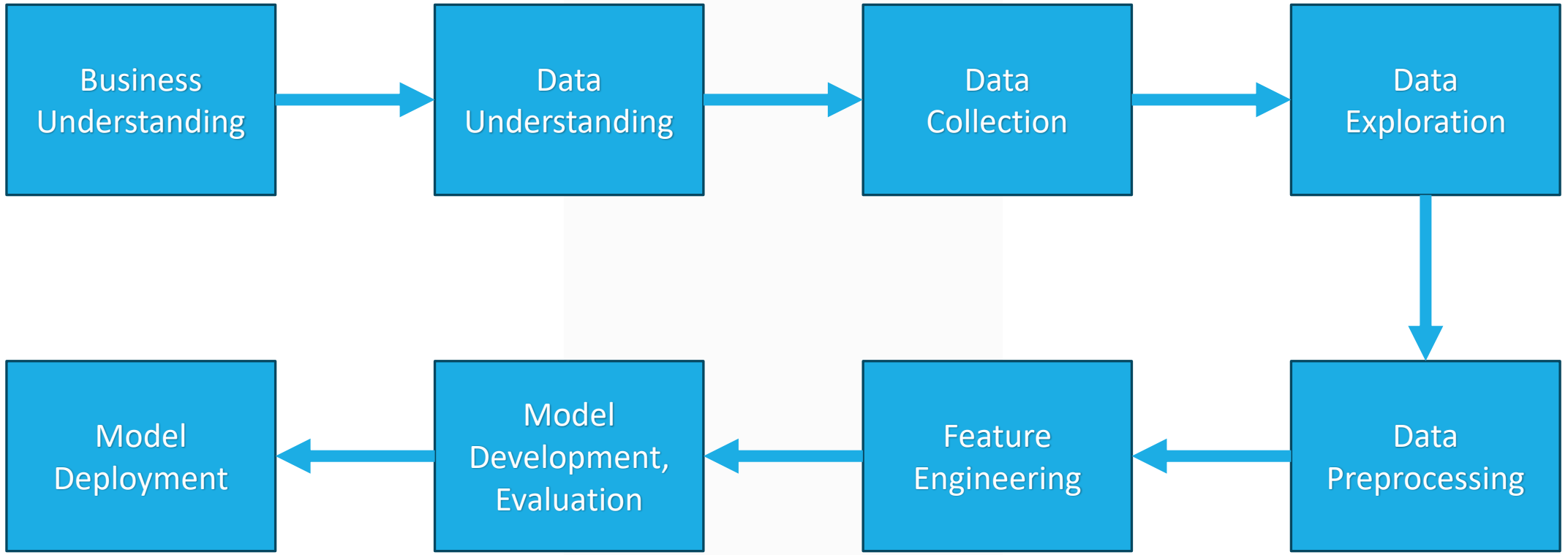
Project Overview:	Project Goal:
❖ SmartTech Co. has partnered with our data science team to develop a robust machine learning model that predicts laptop prices accurately ❖ As the market for laptops continues to expand with a myriad of brands and specifications, having a precise pricing model becomes crucial for both consumers and manufacturers	❖ **Accurate Pricing:** Develop a model that can accurately predict laptop prices based on various features, helping our clients stay competitive in the market ❖ **Market Positioning:** Understand how different features contribute to pricing, enabling SmartTech Co. to strategically position its laptops in the market ❖ **Brand Influence:** Assess the impact of brand reputation on pricing, providing insights into brand perception and market demand

Data Understanding

- Data received in csv file
- Had 13 columns
- Had 1303 records

Sr	Column Labels	Description	Unit of Measurement
1		Sr No without label	No
2	Unnamed: 0	Sr No	No
3	Company	Laptop Brand	-
4	TypeName	Laptop Category (or Family)	-
5	Inches	Screen size	Inches
6	ScreenResolution	Screen resolution	Pixels
7	Cpu	Type of Processor	Generation
8	Ram	Ram Size	GB
9	Memory	Storage Capacity	GB
10	Gpu	Graphics	GB
11	OpSys	Operating System of Laptop	-
12	Weight	Laptop Weight	Kgs
13	Price	Laptop Price	Rs

Process Flow



Technical Stacks

Insights

COMMONLY SOLD LAPTOP SPECIFICATIONS

- **Lenovo** is the most widely sold brand of laptop, which is followed by Dell
- **55%** of laptops are **Notebook**, **8 GB** RAM is the widely used configuration
- **Windows 10** is the widely used operating system
- **Full HD 1920 * 1080** is the widely used screen resolution
- **256 GB SSD** is the widely used memory
- **Intel HD Graphics 620** is widely used Gpu
- **Intel core i5 7200U 2.5 GHz** is widely used Cpu
- **85%** laptops sold were of **non-touch** type [touchscreen]
- **55%** of graphics card (Gpu) are of **intel**, **95%** of processors (Cpu) are of **intel**

Model Evaluation

Metrics	Linear Regression	Ridge Regression	Lasso Regression	Random Forest	XG Boost
R2 Score %	55.8	55.8	55.8	75.4	76.1
RMSE	23227	23223	23227	17314	17048
Training Score %	60	60	60	93	98
Test Score %	56	56	56	75	76

Model Deployment

Click here to access the form

✨ Laptop Price Predictor ✨

Find the Best Price for Your Laptop

Select Brand

Select the Laptop Brand

Select ScreenResolution

Select Laptop ScreenR…

Select Laptop Type

Select the Laptop Type

Select Hybrid Storage (GB)

Select the Hybrid Stora…

Enter Screen Size (inches)

Enter Screen Size (i

Whether Touchscreen

- Yes
- No

Select Ram Size (GB)

Select Ram Size (GB)

Select Hard Disk Size (GB)

Select Hard Disk Size (GB)

Select SSD (GB)

Select SSD (GB)

Select Flash Storage (GB)

Select Flash Storage (GB)

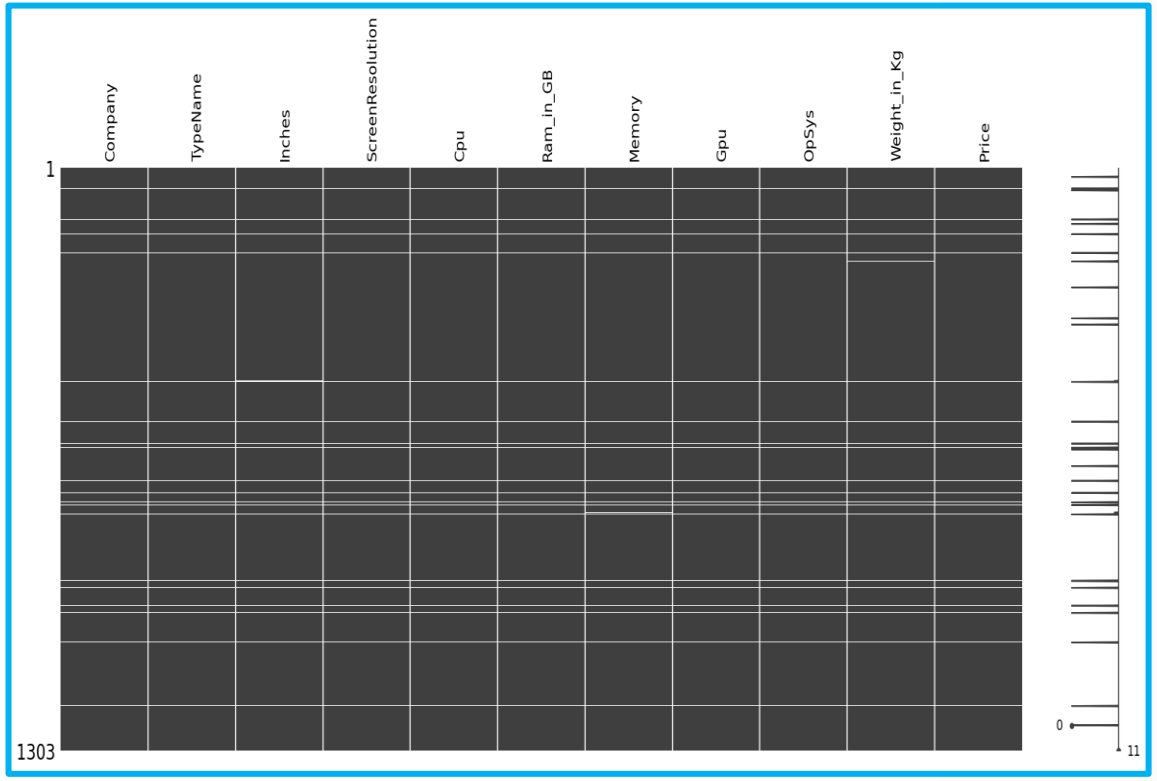
Enter Weight (Kg)

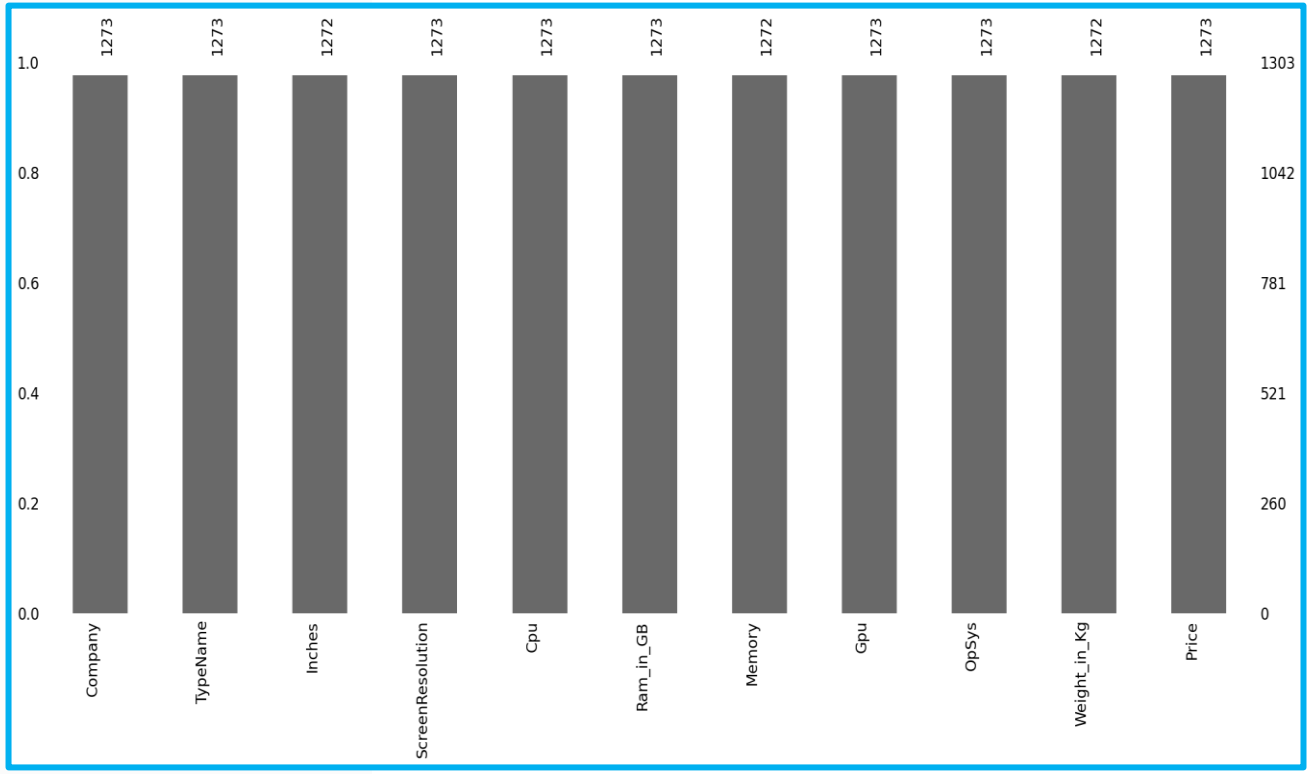
Enter Weight (Kg)

Predict

Appendices

Data Preprocessing





Missing Values & Duplicate Records (29 Nos) Removed:

Company	30	TypeName	30	Inches	31	ScreenResolution	30	Cpu	30
Ram_in_GB	30	Memory	31	Gpu	30	OpSys	30	Weight_in_Kg	31
Price	30								

Data Preprocessing

```
1 df1.isnull().sum()
```

```
Company           30
TypeName          30
Inches            31
ScreenResolution  30
Cpu               30
Ram_in_GB         30
Memory            31
Gpu               30
OpSys             30
Weight_in_Kg      31
Price             30
dtype: int64
```

After removing null values

```
1 df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1270 entries, 0 to 1302
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Company           1270 non-null   object
 1   TypeName          1270 non-null   object
 2   Inches            1270 non-null   float64
 3   ScreenResolution  1270 non-null   object
 4   Cpu               1270 non-null   object
 5   Ram_in_GB         1270 non-null   object
 6   Memory            1270 non-null   object
 7   Gpu               1270 non-null   object
 8   OpSys             1270 non-null   object
 9   Weight_in_Kg      1270 non-null   object
 10  Price             1270 non-null   float64
dtypes: float64(2), object(9)
memory usage: 119.1+ KB
```

After removing duplicates

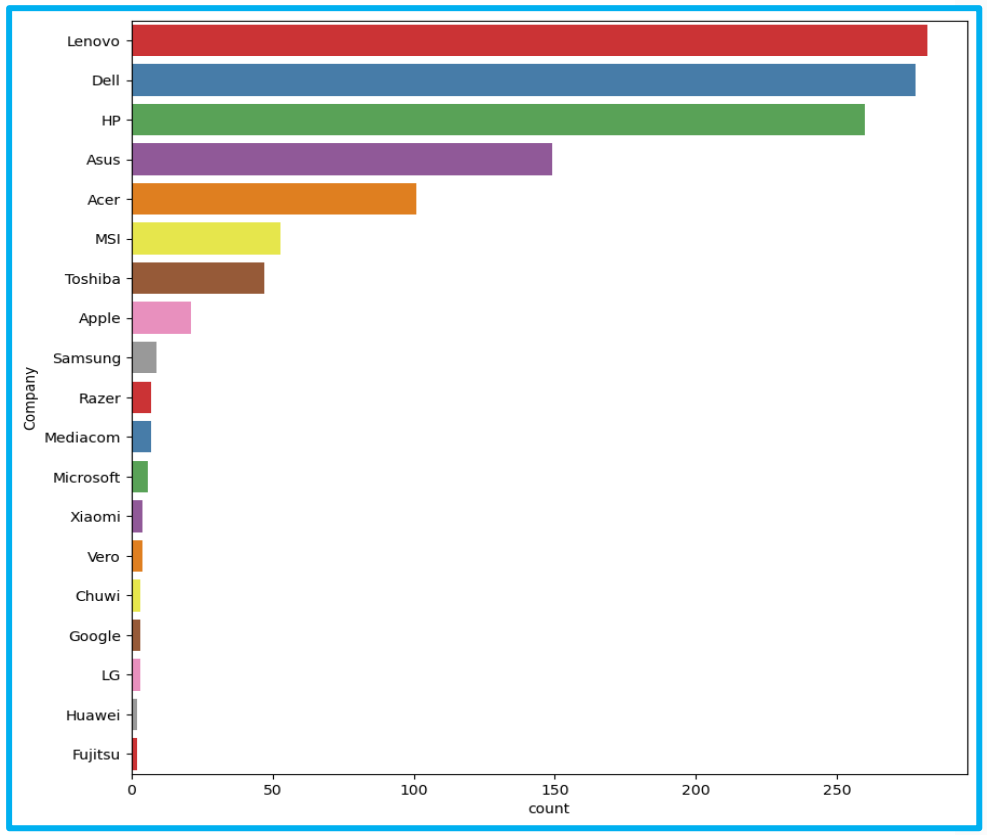
```
1 df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1241 entries, 0 to 1273
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Company           1241 non-null   object
 1   TypeName          1241 non-null   object
 2   Inches            1241 non-null   float64
 3   ScreenResolution  1241 non-null   object
 4   Cpu               1241 non-null   object
 5   Ram_in_GB         1241 non-null   object
 6   Memory            1241 non-null   object
 7   Gpu               1241 non-null   object
 8   OpSys             1241 non-null   object
 9   Weight_in_Kg      1241 non-null   object
 10  Price             1241 non-null   float64
dtypes: float64(2), object(9)
memory usage: 116.3+ KB
```

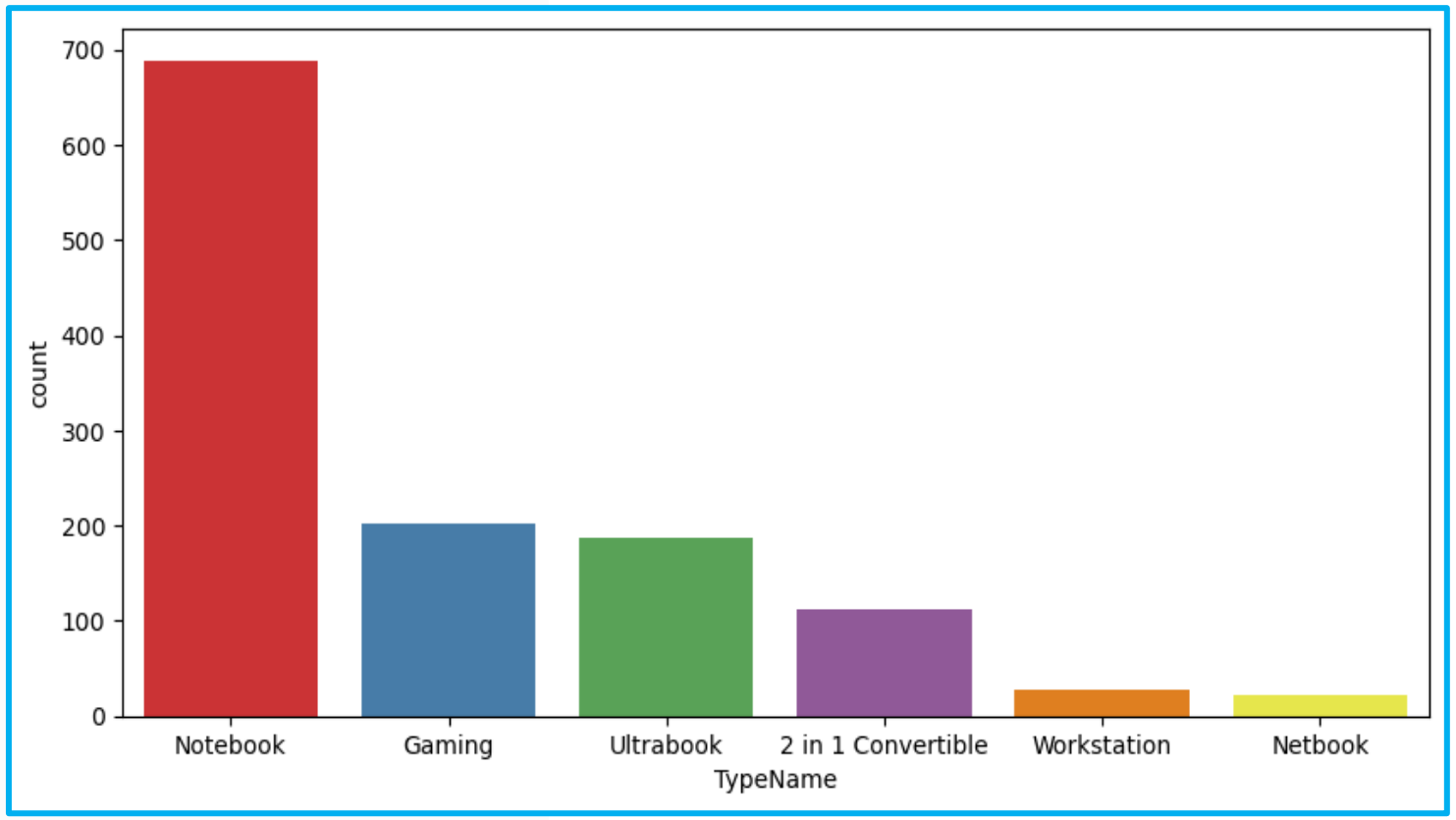
Missing Values & Duplicate Records (29 Nos) Removed:

Company	30	TypeName	30	Inches	31	ScreenResolution	30	Cpu	30
Ram_in_GB	30	Memory	31	Gpu	30	OpSys	30	Weight_in_Kg	31
Price	30								

Data Preprocessing

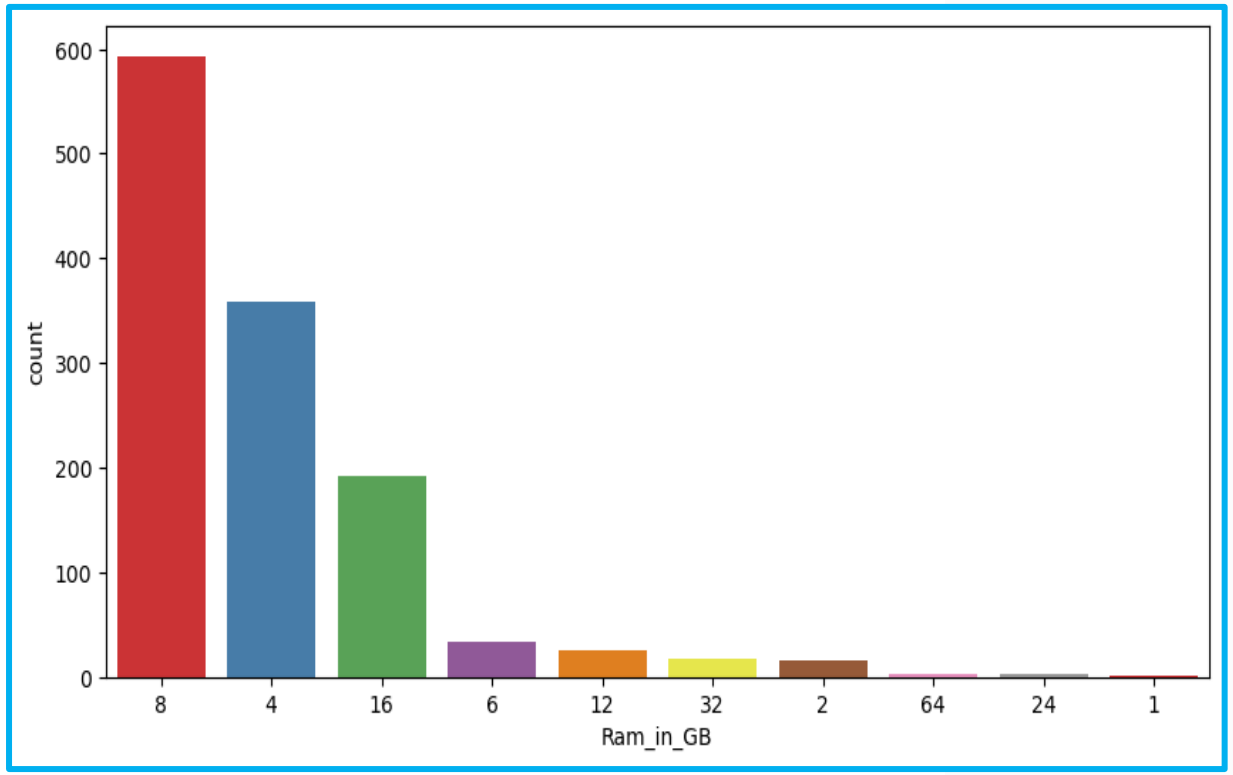


Laptop Brand (Company) Distribution

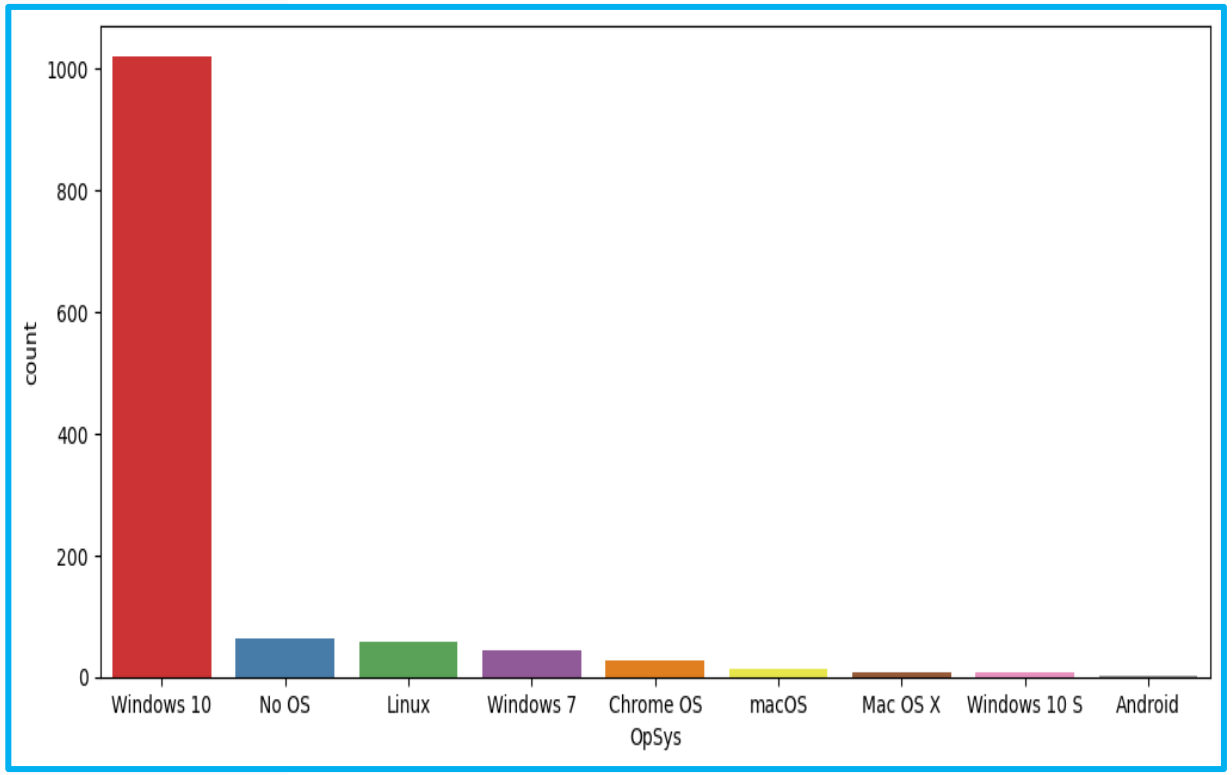


Laptop Type In GB Distribution

Data Preprocessing

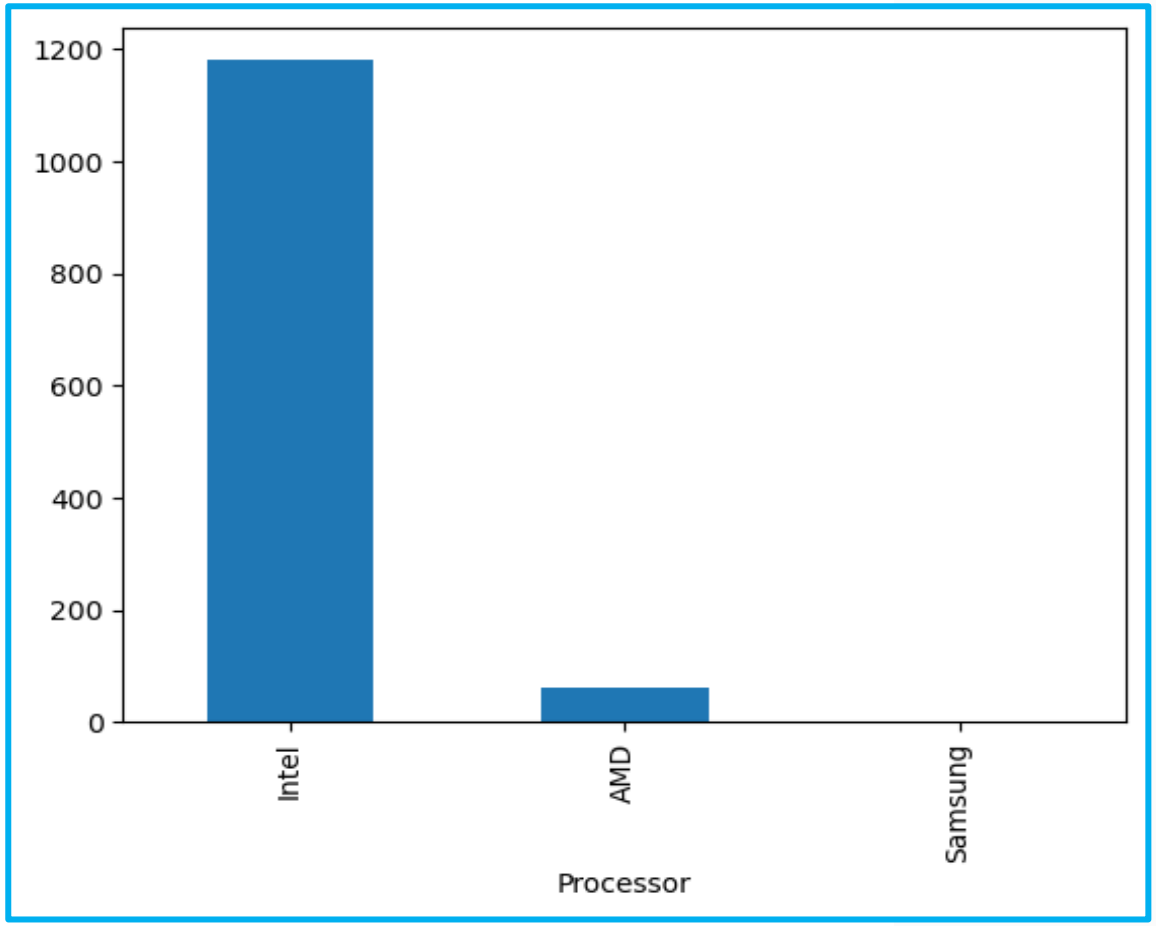


Laptop RAM In GB Distribution

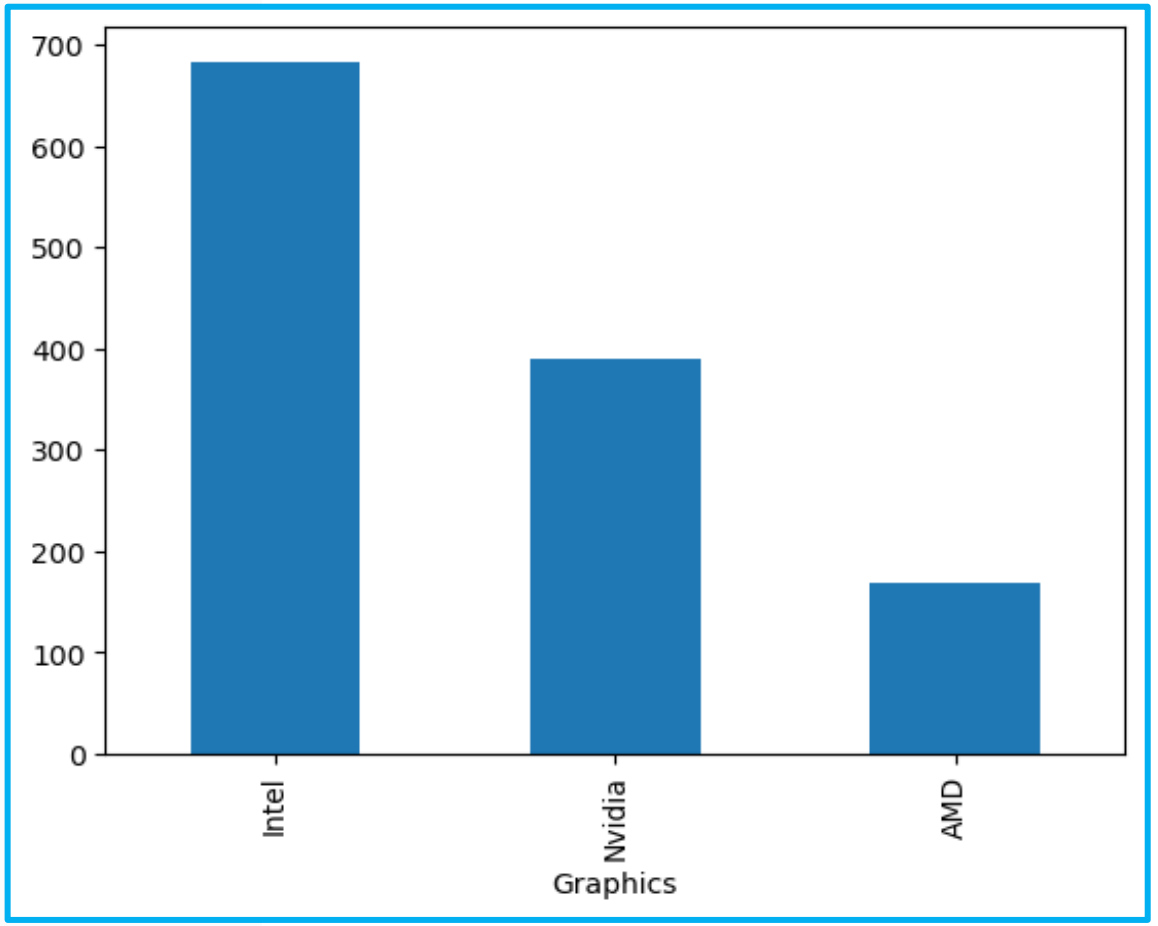


Laptop Operating System Distribution

Feature Engineering

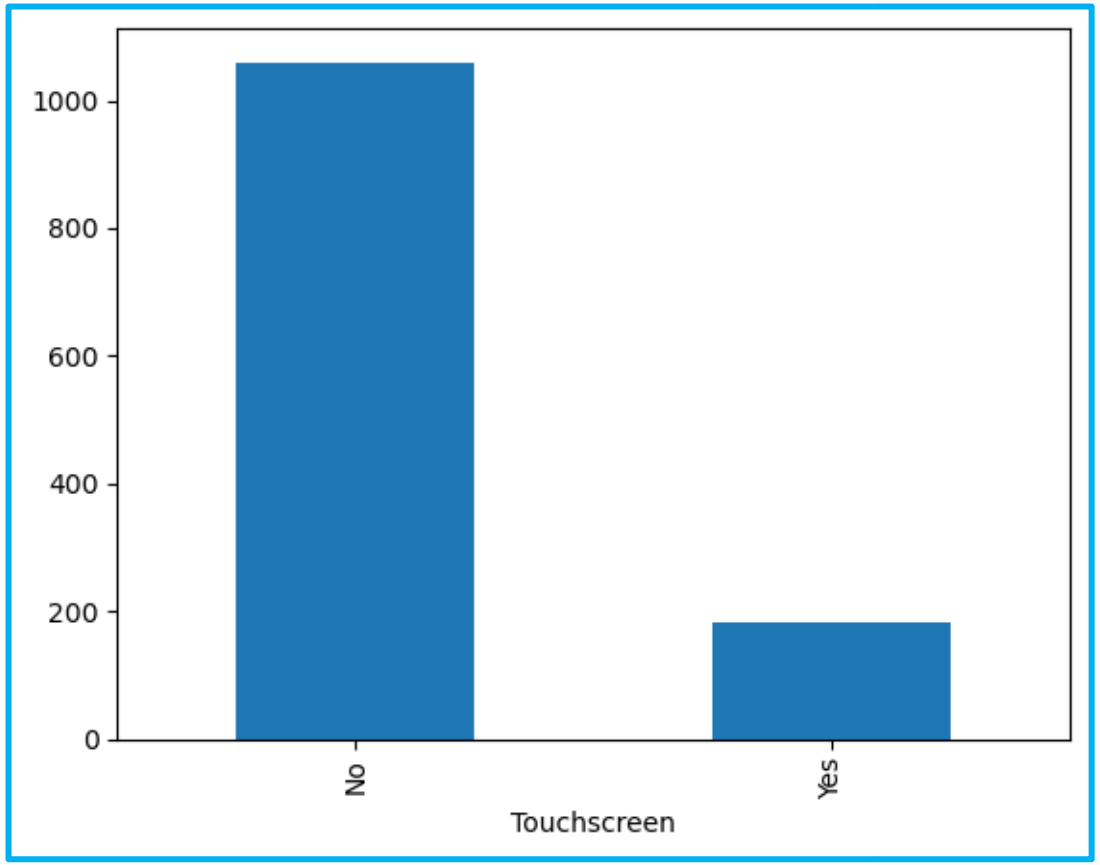


Laptop Processor Brand Distribution



Laptop Graphics Brand Distribution

Feature Engineering



Laptop Touch Distribution

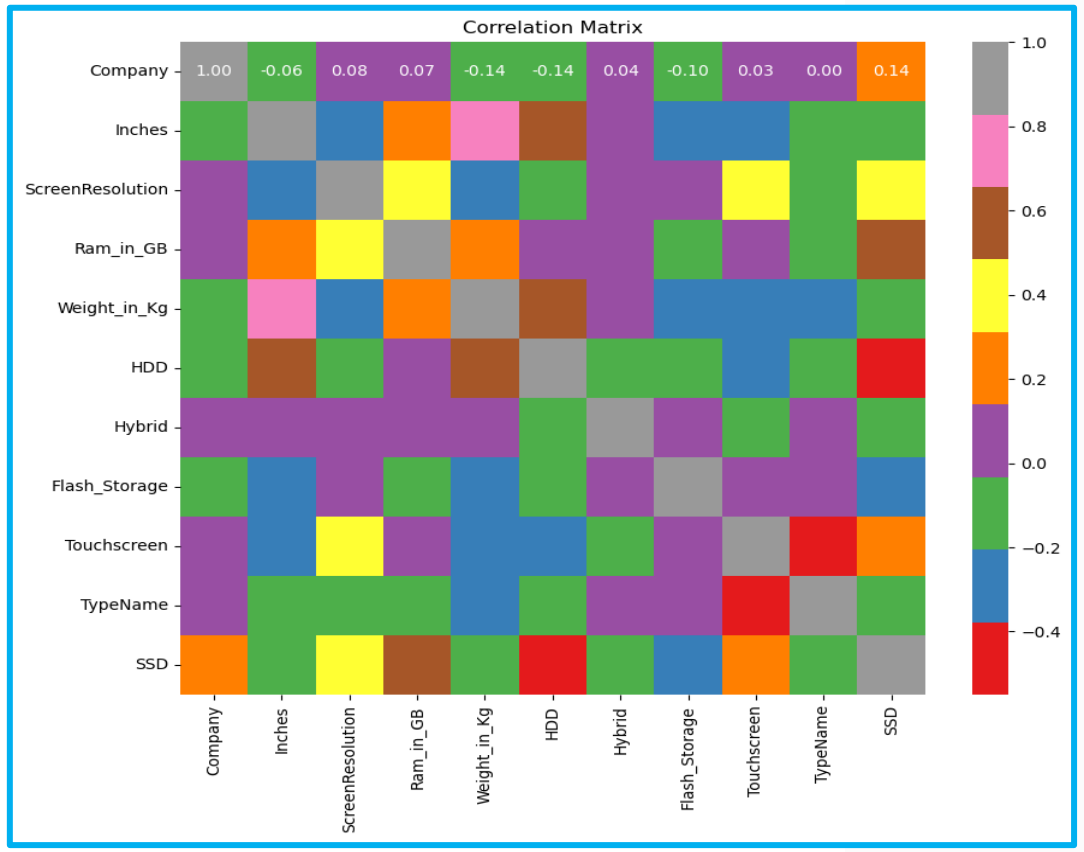
Multicollinearity Check

```python
from statsmodels.stats.outliers_influence import variance_inflation_factor
X = df2
# VIF dataframe
vif_data = pd.DataFrame()
vif_data["feature"] = X.columns
vif_data["VIF"] = [variance_inflation_factor(X.values, i)
                   for i in range(len(X.columns))]
vif_data.sort_values(by='VIF')
```

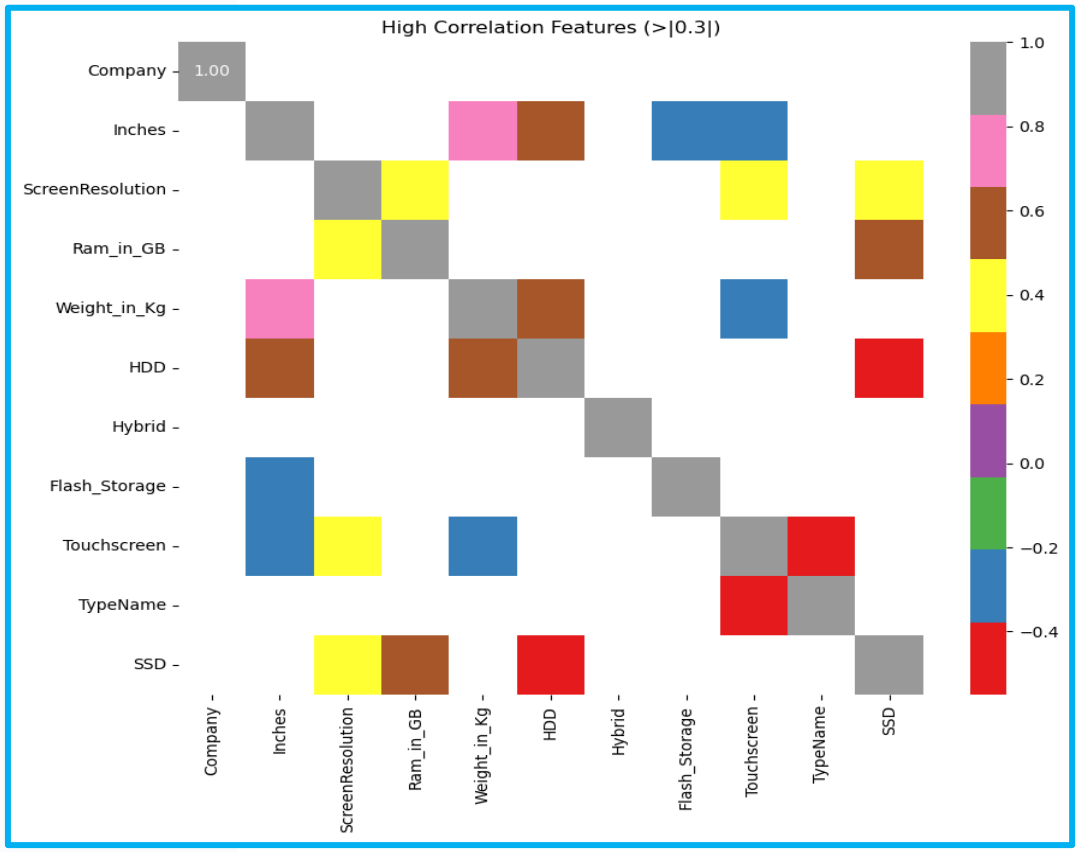
	feature	VIF
11	Hybrid	1.135975
12	Flash_Storage	1.465751
4	Ram_in_GB	1.903230
14	Touchscreen	2.168878
2	Inches	2.749356
7	Weight_in_Kg	3.322219
3	ScreenResolution	3.572620
0	Company	3.676908
9	HDD	4.645731
1	TypeName	7.932320
10	SSD	9.258231
6	OpSys	20.359838
8	Processor	20.615053
13	Graphics	33.966672
5	Gpu	56.719392

Selecting Features with VIF < 10

Correlation Among Features

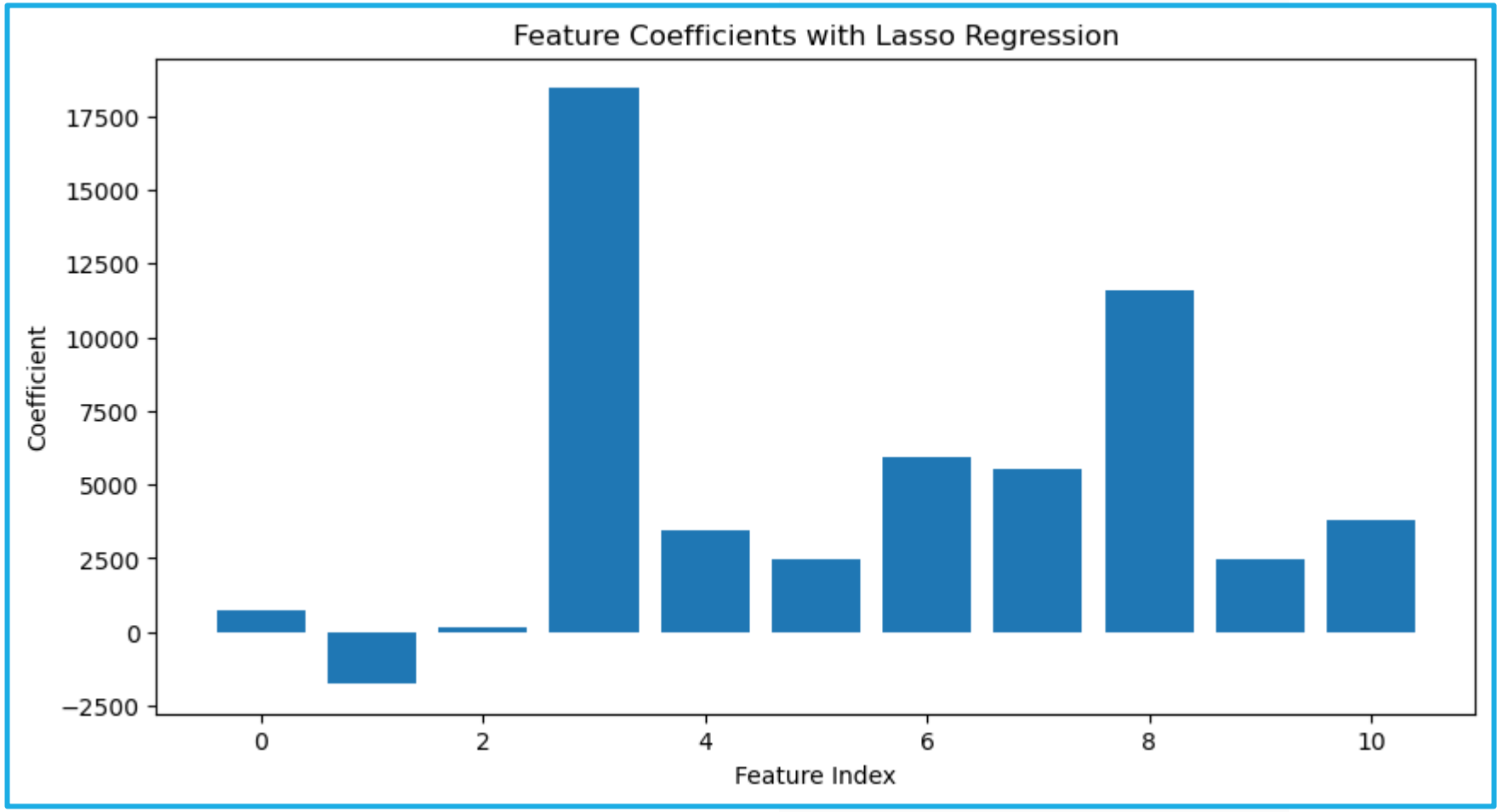


Correlation among features



Correlation among features > 0.30

Feature Importance



Feature Importance

Feature Index	Feature
0	Company
1	Inches
2	ScreenResolution
3	Ram_in_GB
4	Weight_in_Kg
5	HDD
6	Hybrid
7	Flash_Storage
8	Touchscreen
9	TypeName
10	SSD

Hyper Parameter Tuning

For max depth of 40,

100 n_estimators produce optimized results

Hyper Parmeter Tuning - Grid Search

```
from sklearn.model_selection import GridSearchCV

param_grid = {
    'bootstrap': [True],
    'max_depth': [10,20,40,60,80, 90, 100, 110],
    'max_features': [1,2, 3],
    'min_samples_leaf': [1,3, 4, 5],
    'min_samples_split': [1,2,4,8, 10, 12],
    'n_estimators': [20,50,100, 200, 300,500,600]
}

# Creating the base model
rf = RandomForestRegressor(random_state = 42)

grid_search = GridSearchCV(estimator = rf, param_grid = param_grid,
                           cv = 5, n_jobs = -1, verbose = 2, return_train_score=True)
```

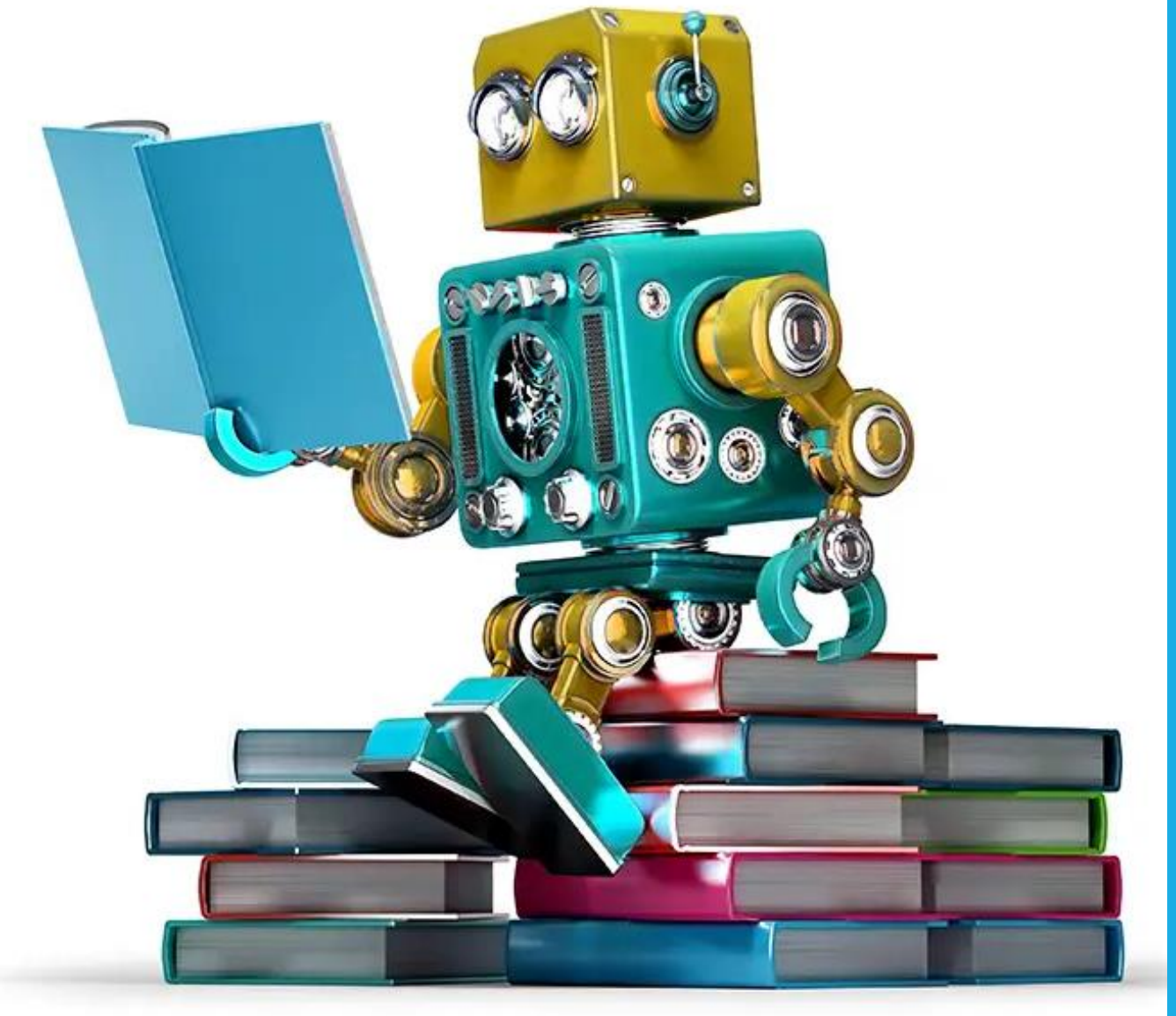
```
# Fit the grid search to the data
grid_search.fit(xtrain, ytrain);
```

```
Fitting 5 folds for each of 4032 candidates, totalling 20160 fits
```

```
grid_search.best_params_
```

```
{'bootstrap': True,
 'max_depth': 40,
 'max_features': 1,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'n_estimators': 100}
```

Questions



Thankyou