



# LAPTOP PRICE ANALYSIS

A Machine Learning Approach to Price Prediction

Presentation by Himanshu Meshram



# PROBLEM STATEMENT

The goal of this project is to predict the price of laptops based on their specifications. This involves analyzing various features such as brand, processor, RAM, storage, and display type.

By analyzing various laptop features, we can build a predictive model that helps consumers make informed purchasing decisions.

# PROJECT WORKFLOW



1. Load and preprocess the dataset.
2. Convert categorical data to numerical format using One-Hot Encoding.
1. Train a Linear Regression model to predict laptop prices.
2. Evaluate the model using MSE and R-squared.
3. Visualize the actual vs predicted prices.
4. Following this structured approach ensures a systematic and efficient way to develop an accurate pricing model.
5. Here is the [Data](#).

# DATA PREPROCESSING

**Preprocessing is a crucial step where we:**

- Handle missing values
- Convert categorical data into numerical format using One-Hot Encoding
- Normalize/scale numerical features if necessary
- One-Hot Encoding helps convert categorical variables (e.g., brand, processor) into a machine-readable format.

## ▼ Data Cleaning

```
[ ] data.info()
```

```
data.isnull().sum()
```

```
data.duplicated().sum()
```

```
...
```

```
[ ] data.drop(columns=["Unnamed: 0"], inplace=True)
```

```
[ ] data.head()
```

```
[ ] data["Ram"] = data["Ram"].astype(str).str.replace("GB", "")
```

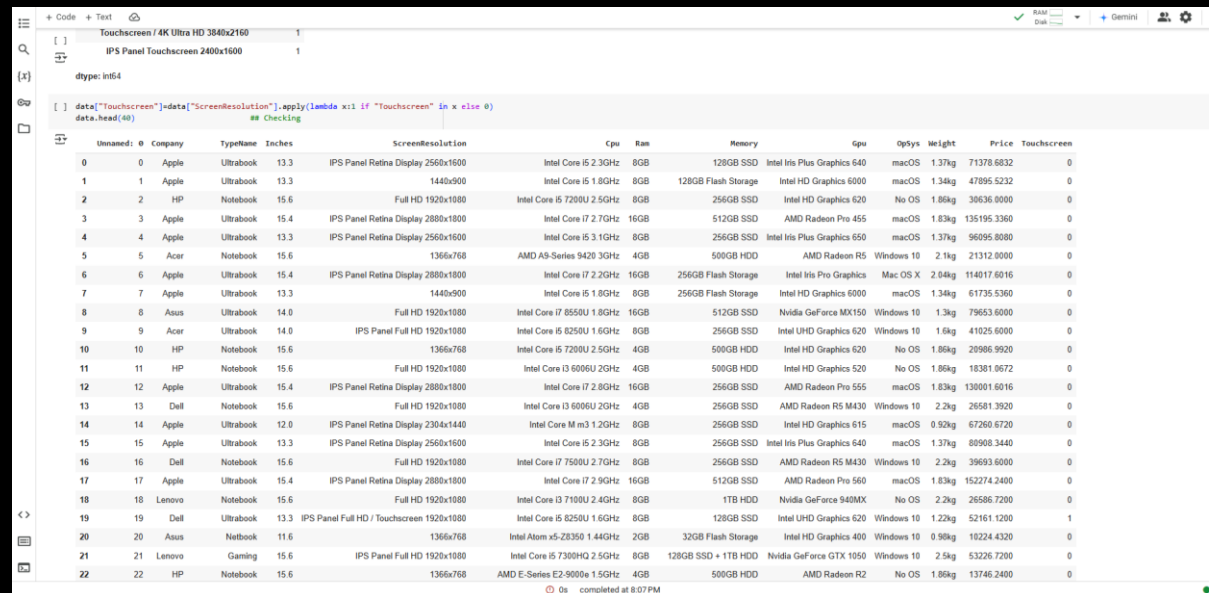
```
[ ] data["Weight"] = data["Weight"].str.replace("kg", "")
```

```
[ ] data["Ram"] = data["Ram"].astype('int')  
data["Weight"] = data["Weight"].astype('float')
```

```
[ ] data.info()
```

# TRAIN-TEST SPLIT

- To ensure our model generalizes well, we split the dataset into training and testing sets. This allows us to evaluate the model's performance on unseen data.
- A well-balanced train-test split prevents overfitting and ensures the model generalizes well to new data.



The screenshot shows a Jupyter Notebook interface. The top part displays two variables: 'Touchscreen' (a Series of 1s) and 'ScreenResolution' (a Series of strings). Below these, a code cell contains the following Python code:

```
[ ] data["Touchscreen"] = data["ScreenResolution"].apply(lambda x: 1 if "Touchscreen" in x else 0)
data.head(40)
```

The code cell is followed by a table of 40 rows of laptop specifications. The table has the following columns: Unnamed: 0, Company, TypeName, Inches, ScreenResolution, Cpu, Ram, Memory, Gpu, OpSys, Weight, Price, and Touchscreen. The 'Touchscreen' column contains 0s and 1s, indicating whether the laptop has a touchscreen.

| Unnamed: 0 | Company | TypeName  | Inches | ScreenResolution                          | Cpu                          | Ram  | Memory              | Gpu                          | OpSys      | Weight | Price       | Touchscreen |
|------------|---------|-----------|--------|-------------------------------------------|------------------------------|------|---------------------|------------------------------|------------|--------|-------------|-------------|
| 0          | Apple   | Ultrabook | 13.3   | IPS Panel Retina Display 2560x1600        | Intel Core i5 2.3GHz         | 8GB  | 128GB SSD           | Intel Iris Plus Graphics 640 | macOS      | 1.37kg | 71378.6832  | 0           |
| 1          | Apple   | Ultrabook | 13.3   | 1440x900                                  | Intel Core i5 1.8GHz         | 8GB  | 128GB Flash Storage | Intel HD Graphics 6000       | macOS      | 1.34kg | 47895.5232  | 0           |
| 2          | HP      | Notebook  | 15.6   | Full HD 1920x1080                         | Intel Core i5 7200U 2.5GHz   | 8GB  | 256GB SSD           | Intel HD Graphics 620        | No OS      | 1.86kg | 30636.0000  | 0           |
| 3          | Apple   | Ultrabook | 15.4   | IPS Panel Retina Display 2880x1800        | Intel Core i7 2.7GHz         | 16GB | 512GB SSD           | AMD Radeon Pro 455           | macOS      | 1.83kg | 135195.3360 | 0           |
| 4          | Apple   | Ultrabook | 13.3   | IPS Panel Retina Display 2560x1600        | Intel Core i5 3.1GHz         | 8GB  | 256GB SSD           | Intel Iris Plus Graphics 650 | macOS      | 1.37kg | 96095.8080  | 0           |
| 5          | Acer    | Notebook  | 15.6   | 1366x768                                  | AMD A9-Series 9420 3GHz      | 4GB  | 500GB HDD           | AMD Radeon R5                | Windows 10 | 2.1kg  | 21312.0000  | 0           |
| 6          | Apple   | Ultrabook | 15.4   | IPS Panel Retina Display 2880x1800        | Intel Core i7 2.2GHz         | 16GB | 256GB Flash Storage | Intel Iris Pro Graphics      | Mac OS X   | 2.04kg | 114017.6016 | 0           |
| 7          | Apple   | Ultrabook | 13.3   | 1440x900                                  | Intel Core i5 1.8GHz         | 8GB  | 256GB Flash Storage | Intel HD Graphics 6000       | macOS      | 1.34kg | 61735.5360  | 0           |
| 8          | Asus    | Ultrabook | 14.0   | Full HD 1920x1080                         | Intel Core i7 8550U 1.8GHz   | 16GB | 512GB SSD           | Nvidia GeForce MX150         | Windows 10 | 1.3kg  | 79653.6000  | 0           |
| 9          | Acer    | Ultrabook | 14.0   | IPS Panel Full HD 1920x1080               | Intel Core i5 8250U 1.6GHz   | 8GB  | 256GB SSD           | Intel UHD Graphics 620       | Windows 10 | 1.6kg  | 41025.6000  | 0           |
| 10         | HP      | Notebook  | 15.6   | 1366x768                                  | Intel Core i5 7200U 2.5GHz   | 4GB  | 500GB HDD           | Intel HD Graphics 620        | No OS      | 1.86kg | 29986.9920  | 0           |
| 11         | HP      | Notebook  | 15.6   | Full HD 1920x1080                         | Intel Core i3 6006U 2GHz     | 4GB  | 500GB HDD           | Intel HD Graphics 520        | No OS      | 1.86kg | 18381.0672  | 0           |
| 12         | Apple   | Ultrabook | 15.4   | IPS Panel Retina Display 2880x1800        | Intel Core i7 2.8GHz         | 16GB | 256GB SSD           | AMD Radeon Pro 555           | macOS      | 1.83kg | 130001.6016 | 0           |
| 13         | Dell    | Notebook  | 15.6   | Full HD 1920x1080                         | Intel Core i3 6006U 2GHz     | 4GB  | 256GB SSD           | AMD Radeon R5 M430           | Windows 10 | 2.2kg  | 26581.3920  | 0           |
| 14         | Apple   | Ultrabook | 12.0   | IPS Panel Retina Display 2304x1440        | Intel Core M m3 1.2GHz       | 8GB  | 256GB SSD           | Intel HD Graphics 615        | macOS      | 0.92kg | 67260.6720  | 0           |
| 15         | Apple   | Ultrabook | 13.3   | IPS Panel Retina Display 2560x1600        | Intel Core i5 2.3GHz         | 8GB  | 256GB SSD           | Intel Iris Plus Graphics 640 | macOS      | 1.37kg | 80900.3440  | 0           |
| 16         | Dell    | Notebook  | 15.6   | Full HD 1920x1080                         | Intel Core i7 7500U 2.7GHz   | 8GB  | 256GB SSD           | AMD Radeon R5 M430           | Windows 10 | 2.2kg  | 39693.6000  | 0           |
| 17         | Apple   | Ultrabook | 15.4   | IPS Panel Retina Display 2880x1800        | Intel Core i7 2.9GHz         | 16GB | 512GB SSD           | AMD Radeon Pro 560           | macOS      | 1.83kg | 152274.2400 | 0           |
| 18         | Lenovo  | Notebook  | 15.6   | Full HD 1920x1080                         | Intel Core i3 7100U 2.4GHz   | 8GB  | 1TB HDD             | Nvidia GeForce 940MX         | No OS      | 2.2kg  | 26586.7200  | 0           |
| 19         | Dell    | Ultrabook | 13.3   | IPS Panel Full HD / Touchscreen 1920x1080 | Intel Core i5 8250U 1.6GHz   | 8GB  | 128GB SSD           | Intel UHD Graphics 620       | Windows 10 | 1.22kg | 52161.1200  | 1           |
| 20         | Asus    | Netbook   | 11.6   | 1366x768                                  | Intel Atom x5-Z8350 1.44GHz  | 2GB  | 32GB Flash Storage  | Intel HD Graphics 400        | Windows 10 | 0.98kg | 10224.4320  | 0           |
| 21         | Lenovo  | Gaming    | 15.6   | IPS Panel Full HD 1920x1080               | Intel Core i5 7300HQ 2.5GHz  | 8GB  | 128GB SSD + 1TB HDD | Nvidia GeForce GTX 1050      | Windows 10 | 2.5kg  | 53226.7200  | 0           |
| 22         | HP      | Notebook  | 15.6   | 1366x768                                  | AMD E-Series E2-9000e 1.5GHz | 4GB  | 500GB HDD           | AMD Radeon R2                | No OS      | 1.86kg | 13746.2400  | 0           |



# MODEL TRAINING - LINEAR REGRESSION

- Linear Regression is a simple yet effective model to start with. It finds a relationship between input features and the target variable (price) by minimizing error.

## LINEAR REGRESSION

```
[ ] step1 = ColumnTransformer(transformers=[
    ('col_tnf', OneHotEncoder(sparse_output=False, drop='first'), [0,1,7,10,11])
], remainder='passthrough')

step2 = Lasso(alpha=0.001)

pipe = Pipeline([
    ('step1', step1),
    ('step2', step2)
])

pipe.fit(X_train, y_train)

y_pred = pipe.predict(X_test)

print('R2 score', r2_score(y_test, y_pred))
print('MAE', mean_absolute_error(y_test, y_pred))
```

```
step1 = ColumnTransformer(transformers=[
    ('col_tnf', OneHotEncoder(sparse_output=False, drop='first'), [0,1,7,10,11])
], remainder='passthrough')

step2 = KNeighborsRegressor(n_neighbors=3)

pipe = Pipeline([
    ('step1', step1),
    ('step2', step2)
])

pipe.fit(X_train, y_train)

y_pred = pipe.predict(X_test)


print('R2 score', r2_score(y_test, y_pred))
print('MAE', mean_absolute_error(y_test, y_pred))
```

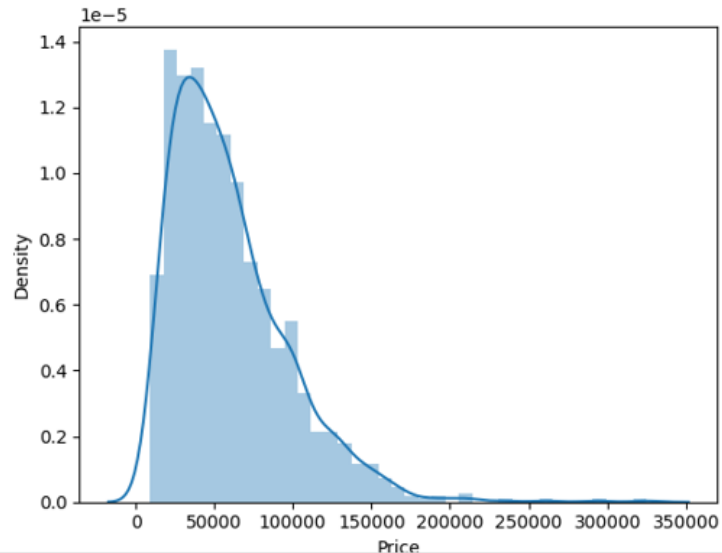
- Linear Regression assumes a linear relationship between input features and the target variable. While simple, it provides valuable baseline results.

# MODEL EVALUATION

## Model

```
[ ] sns.distplot(data["Price"]) ## tiled LEFT side
```

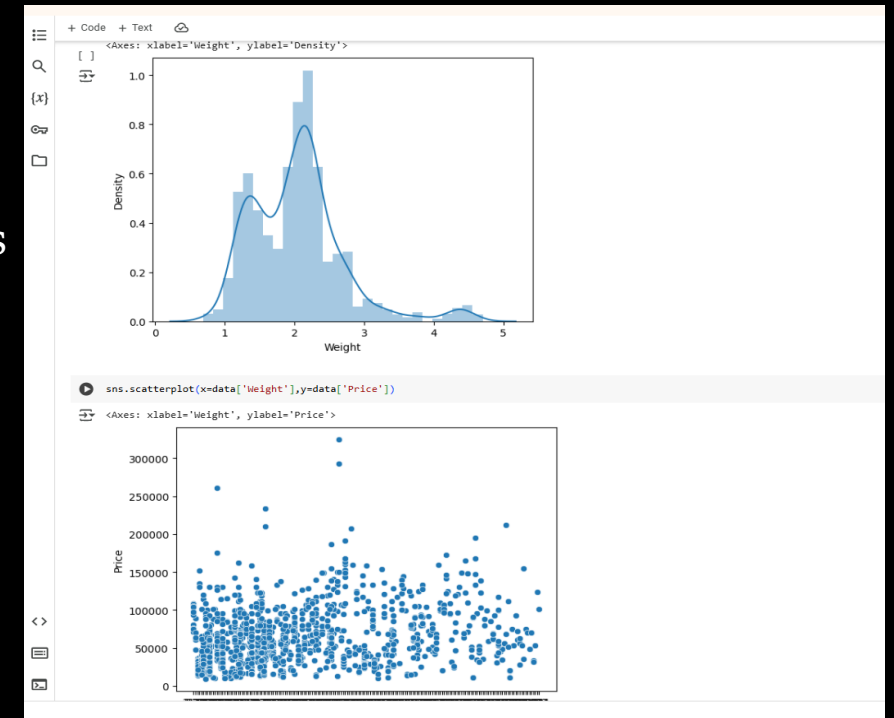
 <Axes: xlabel='Price', ylabel='Density'>



- We assess the model using:
- Mean Squared Error (MSE): Measures the average squared error.
- R-squared: Explains how well the independent variables predict the dependent variable.
- Low MSE and high R-squared values indicate a well-performing model, while high MSE suggests room for improvement.

# VISUALIZING PREDICTIONS

- A scatter plot of actual vs predicted prices helps us understand the model's accuracy. Ideally, points should lie close to the  $y = x$  line.
- Data visualization helps interpret model accuracy and spot patterns in pricing trends.





# FUTURE IMPROVEMENTS

- Experiment with more advanced models like Random Forest or XGBoost.
- Perform feature engineering to extract meaningful information.
- Tune hyperparameters for better accuracy.
- Hyperparameter tuning and ensemble models like XGBoost can significantly improve accuracy.

# PROJECT PHOTOS

|    | A  | B           | C         | D    | E         | F       | G     | H    | I     | J      | K      | L      | M       | N        |         |
|----|----|-------------|-----------|------|-----------|---------|-------|------|-------|--------|--------|--------|---------|----------|---------|
| 1  |    | Company     | Type      | Name | Inches    | Screen  | Res   | Cpu  | Ram   | Memory | Gpu    | OpSys  | Weight  | Price    |         |
| 2  | 0  | Apple       | Ultrabook | 13.3 | IPS Panel | F Intel | Core  | 8GB  | 128GB | SSC    | Intel  | Iris   | PI      | macOS    |         |
| 3  | 1  | Apple       | Ultrabook | 13.3 | 1440x900  | Intel   | Core  | 8GB  | 128GB | Flar   | Intel  | HD     | Q       | macOS    |         |
| 4  | 2  | HP          | Notebook  | 15.6 | Full HD   | 19      | Intel | Core | 8GB   | 256GB  | SSC    | Intel  | HD      | Q        | No OS   |
| 5  | 3  | Apple       | Ultrabook | 15.4 | IPS Panel | F Intel | Core  | 16GB | 512GB | SSC    | AMD    | Rade   | macOS   | 1.83kg   |         |
| 6  | 4  | Apple       | Ultrabook | 13.3 | IPS Panel | F Intel | Core  | 8GB  | 256GB | SSC    | Intel  | Iris   | PI      | macOS    |         |
| 7  | 5  | Acer        | Notebook  | 15.6 | 1366x768  | AMD     | A9-Si | 4GB  | 500GB | HD     | AMD    | Rade   | Windows | 2.1kg    |         |
| 8  | 6  | Apple       | Ultrabook | 15.4 | IPS Panel | F Intel | Core  | 16GB | 256GB | Flar   | Intel  | HD     | Q       | Mac OS X |         |
| 9  | 7  | Apple       | Ultrabook | 13.3 | 1440x900  | Intel   | Core  | 8GB  | 256GB | Flar   | Intel  | HD     | Q       | macOS    |         |
| 10 | 8  | Asus        | Ultrabook | 14   | Full HD   | 19      | Intel | Core | 16GB  | 512GB  | SSC    | Nvidia | Ge      | Windows  |         |
| 11 | 9  | Acer        | Ultrabook | 14   | IPS Panel | F Intel | Core  | 8GB  | 256GB | SSC    | Intel  | UHD    | Windows | 1.6kg    |         |
| 12 | 10 | HP          | Notebook  | 15.6 | 1366x768  | Intel   | Core  | 4GB  | 500GB | HD     | Intel  | HD     | Q       | No OS    |         |
| 13 | 11 | HP          | Notebook  | 15.6 | Full HD   | 19      | Intel | Core | 4GB   | 500GB  | HD     | Intel  | HD      | Q        | No OS   |
| 14 | 12 | Apple       | Ultrabook | 15.4 | IPS Panel | F Intel | Core  | 16GB | 256GB | SSC    | AMD    | Rade   | macOS   | 1.83kg   |         |
| 15 | 13 | Dell        | Notebook  | 15.6 | Full HD   | 19      | Intel | Core | 4GB   | 256GB  | SSC    | AMD    | Rade    | Windows  |         |
| 16 | 14 | Apple       | Ultrabook | 12   | IPS Panel | F Intel | Core  | 8GB  | 256GB | SSC    | Intel  | HD     | Q       | macOS    |         |
| 17 | 15 | Apple       | Ultrabook | 13.3 | IPS Panel | F Intel | Core  | 8GB  | 256GB | SSC    | Intel  | Iris   | PI      | macOS    |         |
| 18 | 16 | Dell        | Notebook  | 15.6 | Full HD   | 19      | Intel | Core | 8GB   | 256GB  | SSC    | AMD    | Rade    | Windows  |         |
| 19 | 17 | Apple       | Ultrabook | 15.4 | IPS Panel | F Intel | Core  | 16GB | 512GB | SSC    | AMD    | Rade   | macOS   | 1.83kg   |         |
| 20 | 18 | Lenovo      | Notebook  | 15.6 | Full HD   | 19      | Intel | Core | 8GB   | 1TB    | HDD    | Nvidia | Ge      | No OS    |         |
| 21 | 19 | Dell        | Ultrabook | 13.3 | IPS Panel | F Intel | Core  | 8GB  | 128GB | SSC    | Intel  | UHD    | Windows | 1.22kg   |         |
| 22 | 20 | Asus        | Netbook   | 11.6 | 1366x768  | Intel   | Atom  | 2GB  | 32GB  | Flas   | Intel  | HD     | Q       | Windows  |         |
| 23 | 21 | Lenovo      | Gaming    | 15.6 | IPS Panel | F Intel | Core  | 8GB  | 128GB | SSC    | Nvidia | Ge     | Windows | 2.5kg    |         |
| 24 | 22 | HP          | Notebook  | 15.6 | 1366x768  | AMD     | E-Ser | 4GB  | 500GB | HD     | AMD    | Rade   | No OS   | 1.86kg   |         |
| 25 | 23 | Dell        | 2 in 1    | 13.3 | Full HD   | 19      | Intel | Core | 8GB   | 256GB  | SSC    | Intel  | UHD     | Windows  |         |
| 26 | 24 | HP          | Ultrabook | 15.6 | Full HD   | 19      | Intel | Core | 8GB   | 256GB  | SSC    | Intel  | HD      | Q        | Windows |
| 27 | 25 | Dell        | Notebook  | 15.6 | 1366x768  | Intel   | Core  | 4GB  | 1TB   | HDD    | Intel  | HD     | Q       | Windows  |         |
| 28 | 26 | Apple       | Ultrabook | 13.3 | 1440x900  | Intel   | Core  | 8GB  | 128GB | Flar   | Intel  | HD     | Q       | Mac OS X |         |
| 29 | 27 | Dell        | Notebook  | 15.6 | Full HD   | 19      | Intel | Core | 8GB   | 256GB  | SSC    | AMD    | Rade    | Windows  |         |
| 30 | 28 | Dell        | Ultrabook | 15.6 | Full HD   | 19      | Intel | Core | 8GB   | 256GB  | SSC    | Intel  | UHD     | Windows  |         |
| 31 | 29 | HP          | Notebook  | 17.3 | Full HD   | 19      | Intel | Core | 8GB   | 1TB    | HDD    | Nvidia | Ge      | Windows  |         |
| 32 | 30 | Chuwi       | Notebook  | 15.6 | Full HD   | 19      | Intel | Atom | 4GB   | 64GB   | Flas   | Intel  | HD      | Q        | Windows |
|    |    | laptop_data |           |      |           |         |       |      |       |        |        |        |         |          |         |



```
+ Code + Text
data["layer1HDD"] = data["first"].apply(lambda x: 1 if "HDD" in x else 0)
data["layer1SSD"] = data["first"].apply(lambda x: 1 if "SSD" in x else 0)
data["layer1Hybrid"] = data["first"].apply(lambda x: 1 if "Hybrid" in x else 0)
data["layer1Flash_Storage"] = data["first"].apply(lambda x: 1 if "Flash Storage" in x else 0)

data["first"] = data["first"].str.replace(r'\D', '')

data["second"].fillna("0", inplace = True)

data["layer2HDD"] = data["second"].apply(lambda x: 1 if "HDD" in x else 0)
data["layer2SSD"] = data["second"].apply(lambda x: 1 if "SSD" in x else 0)
data["layer2Hybrid"] = data["second"].apply(lambda x: 1 if "Hybrid" in x else 0)
data["layer2Flash_Storage"] = data["second"].apply(lambda x: 1 if "Flash Storage" in x else 0)

data["second"] = data["second"].str.replace(r'\D', '')

data["first"] = pd.to_numeric(data["first"], errors='coerce').astype('Int64') # changed this line
data["second"] = pd.to_numeric(data["second"], errors='coerce').astype('Int64') # changed this line as well

data["HDD"] = (data["first"]*data["layer1HDD"]+data["second"]*data["layer2HDD"])
data["SSD"] = (data["first"]*data["layer1SSD"]+data["second"]*data["layer2SSD"])
data["Hybrid"] = (data["first"]*data["layer1Hybrid"]+data["second"]*data["layer2Hybrid"])
data["Flash_Storage"] = (data["first"]*data["layer1Flash_Storage"]+data["second"]*data["layer2Flash_Storage"])

data.drop(columns=['first', 'second', 'layer1HDD', 'layer1SSD', 'layer1Hybrid',
                    'layer1Flash_Storage', 'layer2HDD', 'layer2SSD', 'layer2Hybrid',
                    'layer2Flash_Storage'],inplace=True)

data.head()
```

| Unnamed: 0 | Company | Type  | Name      | Ram  | Memory            | Gpu                          | OpSys | Weight | Price       | Touchscreen | Ips | ppl        | Cpu brand     | HDD  | SSD  | Hybrid | Flash_Storage |
|------------|---------|-------|-----------|------|-------------------|------------------------------|-------|--------|-------------|-------------|-----|------------|---------------|------|------|--------|---------------|
| 0          | 0       | Apple | Ultrabook | 8GB  | 128 SSD           | Intel Iris Plus Graphics 640 | macOS | 1.37kg | 71378.6532  | 0           | 1   | 226.963005 | Intel Core i5 | <NA> | <NA> | <NA>   | <NA>          |
| 1          | 1       | Apple | Ultrabook | 8GB  | 128 Flash Storage | Intel HD Graphics 6000       | macOS | 1.34kg | 47895.5232  | 0           | 0   | 127.677940 | Intel Core i5 | <NA> | <NA> | <NA>   | <NA>          |
| 2          | 2       | HP    | Notebook  | 8GB  | 256 SSD           | Intel HD Graphics 620        | No OS | 1.96kg | 30636.0000  | 0           | 0   | 141.211998 | Intel Core i5 | <NA> | <NA> | <NA>   | <NA>          |
| 3          | 3       | Apple | Ultrabook | 16GB | 512 SSD           | AMD Radeon Pro 455           | macOS | 1.83kg | 135195.3360 | 0           | 1   | 220.534624 | Intel Core i7 | <NA> | <NA> | <NA>   | <NA>          |
| 4          | 4       | Apple | Ultrabook | 8GB  | 256 SSD           | Intel Iris Plus Graphics 650 | macOS | 1.37kg | 96095.8080  | 0           | 1   | 226.963005 | Intel Core i5 | <NA> | <NA> | <NA>   | <NA>          |

```
LINEAR REGRESSION

[ ] step1 = ColumnTransformer(transformers=[
    ('col_tmf', OneHotEncoder(sparse_output=False, drop='first'), [0,1,7,10,11])
], remainder='passthrough')

step2 = Lasso(alpha=0.001)

pipe = Pipeline([
    ('step1', step1),
    ('step2', step2)
])

pipe.fit(X_train, y_train)

y_pred = pipe.predict(X_test)

print('R2 score', r2_score(y_test, y_pred))
print('MAE', mean_absolute_error(y_test, y_pred))

[ ] step1 = ColumnTransformer(transformers=[
    ('col_tmf', OneHotEncoder(sparse=False, drop='first'), [0,1,7,10,11])
], remainder='passthrough')

step2 = KNeighborsRegressor(n_neighbors=3)

pipe = Pipeline([
    ('step1', step1),
    ('step2', step2)
])

pipe.fit(X_train, y_train)
```

# CONCLUSION

- This project demonstrated how machine learning can be used to predict laptop prices based on specifications. Key takeaways include:
  - Data preprocessing is crucial for model performance.
  - Linear Regression provides a baseline, but advanced models can improve accuracy.
  - Feature selection and hyperparameter tuning can further optimize predictions.
- Next steps include experimenting with other models like Random Forest or XGBoost for better results.
- A well-optimized model can assist businesses and consumers in making better pricing decisions.

# REFERENCES & LINKS

- [Reference Link](#)
- [Data](#)
- [HimanshuMeshram Github](#)
- [HimanshuMeshram LinkedIn](#)