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.

Touchscreen / 4K Ultra HD 3840x2160	1
IPS Panel Touchscreen 2400x1600	1

dtype: int64

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

	Unnamed: 0	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price	Touchscreen
0	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	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	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	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	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	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	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	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	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	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	10	HP	Notebook	15.6	1366x768	Intel Core i5 7200U 2.5GHz	4GB	500GB HDD	Intel HD Graphics 620	No OS	1.86kg	20986.9920	0
11	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	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	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	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	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	80908.3440	0
16	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	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	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	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	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	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	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

0s completed at 8:07 PM

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.

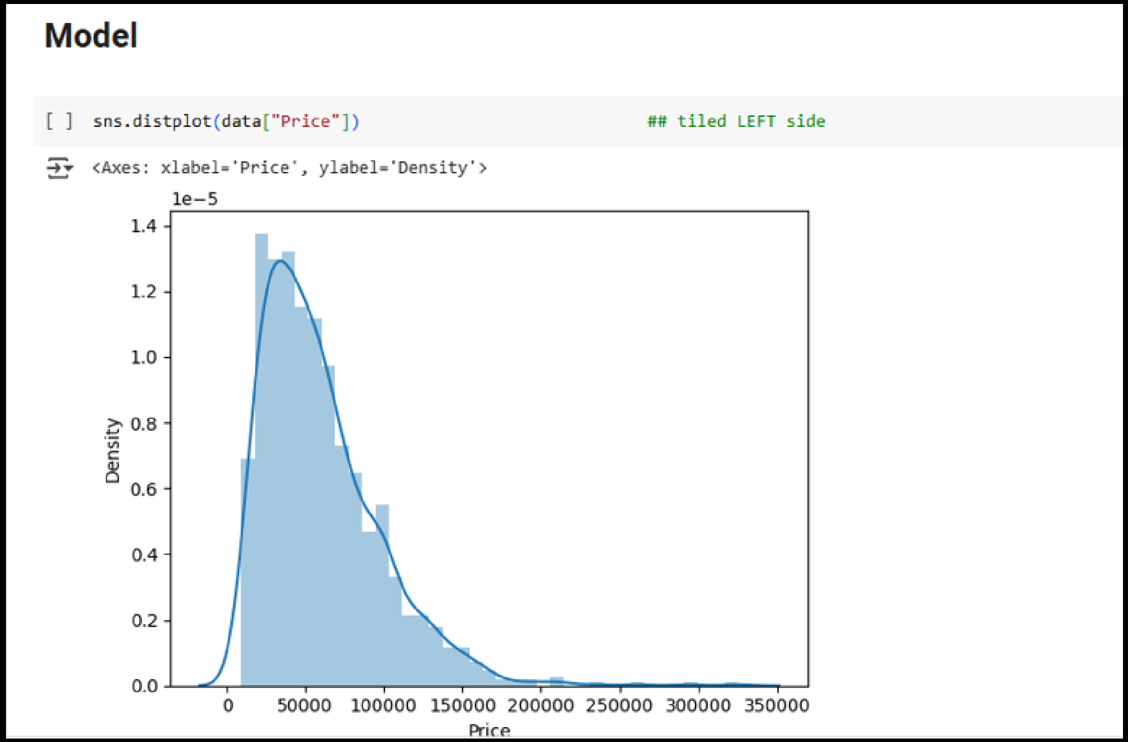
LINEAR REGRESSION

```
[ ] 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=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.

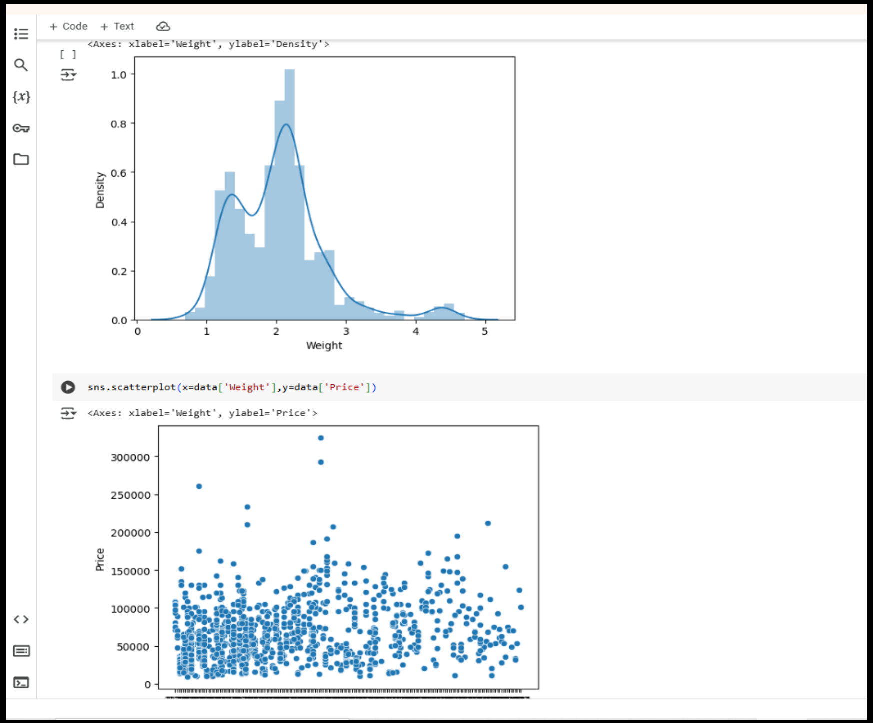
MODEL EVALUATION



- 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.

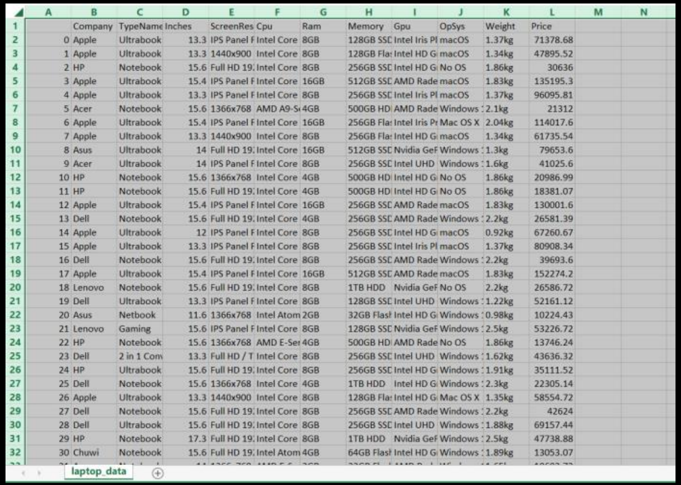


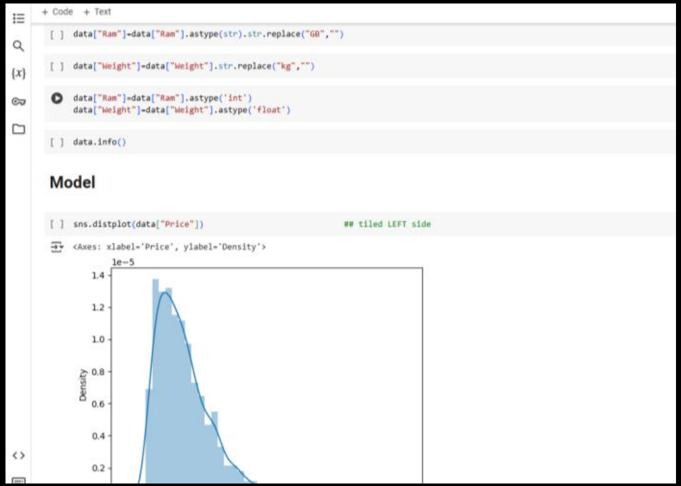
- Data visualization helps interpret model accuracy and spot patterns in pricing trends.

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





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