

# Laptop Price Prediction

August 3, 2023

```
[48]: import numpy as np
import pandas as pd
```

```
[49]: data=pd.read_csv('laptop_price.csv',encoding='latin-1')
```

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

```
[50]:  laptop_ID Company      Product  TypeName  Inches  \
0         1   Apple  MacBook Pro  Ultrabook   13.3
1         2   Apple  Macbook Air  Ultrabook   13.3
2         3     HP      250 G6    Notebook   15.6
3         4   Apple  MacBook Pro  Ultrabook   15.4
4         5   Apple  MacBook Pro  Ultrabook   13.3

      ScreenResolution      Cpu  Ram  \
0  IPS Panel Retina Display 2560x1600  Intel Core i5 2.3GHz  8GB
1                        1440x900  Intel Core i5 1.8GHz  8GB
2      Full HD 1920x1080  Intel Core i5 7200U 2.5GHz  8GB
3  IPS Panel Retina Display 2880x1800  Intel Core i7 2.7GHz 16GB
4  IPS Panel Retina Display 2560x1600  Intel Core i5 3.1GHz  8GB

      Gpu  OpSys  Weight  Price_euros
0  Intel Iris Plus Graphics 640  macOS  1.37kg  1339.69
1      Intel HD Graphics 6000  macOS  1.34kg  898.94
2      Intel HD Graphics 620  No OS  1.86kg  575.00
3      AMD Radeon Pro 455  macOS  1.83kg  2537.45
4  Intel Iris Plus Graphics 650  macOS  1.37kg  1803.60
```

```
[51]: data.shape
```

```
[51]: (1303, 12)
```

```
[52]: data.isnull().sum()
```

```
[52]: laptop_ID      0
Company          0
Product          0
TypeName         0
```

```
Inches          0
ScreenResolution 0
Cpu             0
Ram            0
Gpu            0
OpSys          0
Weight         0
Price_euros    0
dtype: int64
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1303 entries, 0 to 1302
Data columns (total 12 columns):
#   Column          Non-Null Count  Dtype
---  -
0   laptop_ID       1303 non-null   int64
1   Company         1303 non-null   object
2   Product         1303 non-null   object
3   TypeName        1303 non-null   object
4   Inches          1303 non-null   float64
5   ScreenResolution 1303 non-null   object
6   Cpu             1303 non-null   object
7   Ram             1303 non-null   object
8   Gpu             1303 non-null   object
9   OpSys           1303 non-null   object
10  Weight          1303 non-null   object
11  Price_euros     1303 non-null   float64
dtypes: float64(2), int64(1), object(9)
memory usage: 122.3+ KB
```

```
[54]: data['Ram']=data['Ram'].str.replace('GB','').astype('int32')
```

```
[55]: data['Weight']=data['Weight'].str.replace('kg','').astype('float64')
```

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

```
[56]:  laptop_ID Company      Product  TypeName  Inches  \
0         1   Apple  MacBook Pro  Ultrabook   13.3
1         2   Apple  Macbook Air  Ultrabook   13.3
2         3     HP      250 G6    Notebook   15.6
3         4   Apple  MacBook Pro  Ultrabook   15.4
4         5   Apple  MacBook Pro  Ultrabook   13.3

      ScreenResolution      Cpu  Ram  \
0  IPS Panel Retina Display 2560x1600  Intel Core i5 2.3GHz    8
1                1440x900  Intel Core i5 1.8GHz    8
```

2		Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8
3	IPS Panel Retina Display	2880x1800	Intel Core i7 2.7GHz	16
4	IPS Panel Retina Display	2560x1600	Intel Core i5 3.1GHz	8

		Gpu	OpSys	Weight	Price_euros
0	Intel Iris Plus Graphics	640	macOS	1.37	1339.69
1	Intel HD Graphics	6000	macOS	1.34	898.94
2	Intel HD Graphics	620	No OS	1.86	575.00
3	AMD Radeon Pro	455	macOS	1.83	2537.45
4	Intel Iris Plus Graphics	650	macOS	1.37	1803.60

```
[57]: data.corr()['Price_euros']
```

C:\Users\Nimes\AppData\Local\Temp\ipykernel\_10808\703178330.py:1: FutureWarning:  
The default value of numeric\_only in DataFrame.corr is deprecated. In a future  
version, it will default to False. Select only valid columns or specify the  
value of numeric\_only to silence this warning.

```
data.corr()['Price_euros']
```

```
[57]: laptop_ID      0.067830
      Inches        0.068197
      Ram           0.743007
      Weight        0.210370
      Price_euros    1.000000
      Name: Price_euros, dtype: float64
```

```
[58]: data.Company.value_counts()
```

```
[58]: Dell          297
      Lenovo        297
      HP            274
      Asus          158
      Acer          103
      MSI           54
      Toshiba       48
      Apple         21
      Samsung        9
      Razer          7
      Mediacom       7
      Microsoft      6
      Xiaomi         4
      Vero           4
      Chuwi          3
      Google         3
      Fujitsu        3
      LG             3
      Huawei         2
      Name: Company, dtype: int64
```

```
[59]: def add_company(inpt):
        if inpt=='Samsung' or inpt=='Razer' or inpt=='Mediacom' or
        ↪ inpt=='Microsoft' or inpt=='Xiaomi' or inpt=='Vero' or inpt=='Chuwi' or
        ↪ inpt=='Google' or inpt=='Fujitsu' or inpt=='LG' or inpt=='Huawei':
            return "Other"
        else:
            return inpt
data["Company"]=data['Company'].apply(add_company)
```

```
[60]: data.Company.value_counts()
```

```
[60]: Dell          297
      Lenovo       297
      HP           274
      Asus         158
      Acer         103
      MSI          54
      Other        51
      Toshiba      48
      Apple        21
      Name: Company, dtype: int64
```

```
[61]: len(data.Product.value_counts())
```

```
[61]: 618
```

```
[62]: data.TypeName.value_counts()
```

```
[62]: Notebook          727
      Gaming            205
      Ultrabook         196
      2 in 1 Convertible 121
      Workstation        29
      Netbook            25
      Name: TypeName, dtype: int64
```

```
[63]: data.ScreenResolution.value_counts()
```

```
[63]: Full HD 1920x1080          507
      1366x768                  281
      IPS Panel Full HD 1920x1080 230
      IPS Panel Full HD / Touchscreen 1920x1080 53
      Full HD / Touchscreen 1920x1080 47
      1600x900                  23
      Touchscreen 1366x768       16
      Quad HD+ / Touchscreen 3200x1800 15
      IPS Panel 4K Ultra HD 3840x2160 12
      IPS Panel 4K Ultra HD / Touchscreen 3840x2160 11
```

4K Ultra HD / Touchscreen 3840x2160	10
4K Ultra HD 3840x2160	7
Touchscreen 2560x1440	7
IPS Panel 1366x768	7
IPS Panel Quad HD+ / Touchscreen 3200x1800	6
IPS Panel Retina Display 2560x1600	6
IPS Panel Retina Display 2304x1440	6
Touchscreen 2256x1504	6
IPS Panel Touchscreen 2560x1440	5
IPS Panel Retina Display 2880x1800	4
IPS Panel Touchscreen 1920x1200	4
1440x900	4
IPS Panel 2560x1440	4
IPS Panel Quad HD+ 2560x1440	3
Quad HD+ 3200x1800	3
1920x1080	3
Touchscreen 2400x1600	3
2560x1440	3
IPS Panel Touchscreen 1366x768	3
IPS Panel Touchscreen / 4K Ultra HD 3840x2160	2
IPS Panel Full HD 2160x1440	2
IPS Panel Quad HD+ 3200x1800	2
IPS Panel Retina Display 2736x1824	1
IPS Panel Full HD 1920x1200	1
IPS Panel Full HD 2560x1440	1
IPS Panel Full HD 1366x768	1
Touchscreen / Full HD 1920x1080	1
Touchscreen / Quad HD+ 3200x1800	1
Touchscreen / 4K Ultra HD 3840x2160	1
IPS Panel Touchscreen 2400x1600	1

Name: ScreenResolution, dtype: int64

```
[64]: data["Touchscreen"]=data['ScreenResolution'].apply(lambda x:1 if "Touchscreen" in x else 0 )
data["Ips"]=data['ScreenResolution'].apply(lambda x:1 if "IPS" in x else 0 )
```

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

```
[65]:
```

	laptop_ID	Company	Product	TypeName	Inches	\
0	1	Apple	MacBook Pro	Ultrabook	13.3	
1	2	Apple	Macbook Air	Ultrabook	13.3	
2	3	HP	250 G6	Notebook	15.6	
3	4	Apple	MacBook Pro	Ultrabook	15.4	
4	5	Apple	MacBook Pro	Ultrabook	13.3	

	ScreenResolution	Cpu	Ram	\
0	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	

1		1440x900	Intel Core i5 1.8GHz	8
2	Full HD	1920x1080	Intel Core i5 7200U 2.5GHz	8
3	IPS Panel Retina Display	2880x1800	Intel Core i7 2.7GHz	16
4	IPS Panel Retina Display	2560x1600	Intel Core i5 3.1GHz	8

		Gpu	OpSys	Weight	Price_euros	Touchscreen	Ips
0	Intel Iris Plus Graphics	640	macOS	1.37	1339.69	0	1
1	Intel HD Graphics	6000	macOS	1.34	898.94	0	0
2	Intel HD Graphics	620	No OS	1.86	575.00	0	0
3	AMD Radeon Pro	455	macOS	1.83	2537.45	0	1
4	Intel Iris Plus Graphics	650	macOS	1.37	1803.60	0	1

```
[66]: len(data.Cpu.value_counts())
```

```
[66]: 118
```

```
[67]: data['cpu_name']=data['Cpu'].apply(lambda x:" ".join(x.split()[0:3]))
```

```
[68]: data.cpu_name.value_counts()
```

```
[68]: Intel Core i7          527
      Intel Core i5          423
      Intel Core i3          136
      Intel Celeron Dual      80
      Intel Pentium Quad      27
      Intel Core M            19
      AMD A9-Series 9420      12
      Intel Celeron Quad       8
      AMD A6-Series 9220       8
      AMD A12-Series 9720P     7
      Intel Atom x5-Z8350      5
      AMD A8-Series 7410       4
      Intel Atom x5-Z8550      4
      Intel Pentium Dual       3
      AMD A9-Series 9410       3
      AMD Ryzen 1700           3
      AMD A9-Series A9-9420     2
      AMD A10-Series 9620P     2
      Intel Atom X5-Z8350      2
      AMD E-Series E2-9000e     2
      Intel Xeon E3-1535M       2
      Intel Xeon E3-1505M       2
      AMD E-Series 7110        2
      AMD A10-Series 9600P     2
      AMD A6-Series A6-9220     2
      AMD A10-Series A10-9620P  2
      AMD Ryzen 1600           1
```

```

Intel Atom x5-Z8300      1
AMD E-Series E2-6110    1
AMD FX 9830P            1
AMD E-Series E2-9000    1
AMD A6-Series 7310      1
Intel Atom Z8350        1
AMD A12-Series 9700P    1
AMD A4-Series 7210      1
AMD FX 8800P            1
AMD E-Series 9000e      1
Samsung Cortex A72&A53  1
AMD E-Series 9000       1
AMD E-Series 6110       1
Name: cpu_name, dtype: int64

```

```

[69]: def add_cpu(inpt):
        if inpt=='Intel Core i7' or inpt=='Intel Core i5' or inpt=='Intel Core i3':
            return inpt
        else:
            if inpt.split()[0]=="AMD":
                return "AMD"
            else:
                return 'Other'
data["cpu_name"]=data['cpu_name'].apply(add_cpu)

```

```

[70]: data.cpu_name.value_counts()

```

```

[70]: Intel Core i7      527
      Intel Core i5      423
      Other              155
      Intel Core i3      136
      AMD                62
      Name: cpu_name, dtype: int64

```

```

[72]: len(data.Gpu.value_counts())

```

```

[72]: 110

```

```

[73]: data['gpu_name']=data['Gpu'].apply(lambda x: ' '.join(x.split()[0:1]))

```

```

[75]: data['gpu_name'].value_counts()

```

```

[75]: Intel      722
      Nvidia     400
      AMD        180
      ARM         1
      Name: gpu_name, dtype: int64

```

```
[76]: data=data[data['gpu_name']!="ARM"]
```

```
[77]: data["OpSys"].value_counts()
```

```
[77]: Windows 10      1072
      No OS          66
      Linux          62
      Windows 7      45
      Chrome OS      26
      macOS          13
      Mac OS X        8
      Windows 10 S    8
      Android         2
      Name: OpSys, dtype: int64
```

```
[79]: def add_os(inpt):
      if inpt=='Windows 10' or inpt=='Windows 7' or inpt=='Windows 10 S':
          return "Windows"
      else:
          if inpt=="macOS" or inpt=="Mac OS X":
              return "mac"
          else:
              if inpt=="Linux":
                  return "Linux"
              else:
                  return "Other"

      data["Op_system"]=data['OpSys'].apply(add_os)
```

```
[80]: data["Op_system"].value_counts()
```

```
[80]: Windows      1125
      Other         94
      Linux         62
      mac           21
      Name: Op_system, dtype: int64
```

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

```
[81]:  laptop_ID  Company      Product  TypeName  Inches  \
0         1    Apple  MacBook Pro  Ultrabook   13.3
1         2    Apple  Macbook Air  Ultrabook   13.3
2         3      HP      250 G6    Notebook   15.6
3         4    Apple  MacBook Pro  Ultrabook   15.4
4         5    Apple  MacBook Pro  Ultrabook   13.3

      ScreenResolution      Cpu  Ram  \
0  IPS Panel Retina Display 2560x1600  Intel Core i5 2.3GHz  8
```



1		1440x900		Intel Core i5 1.8GHz	8
2		Full HD 1920x1080		Intel Core i5 7200U 2.5GHz	8
3	IPS Panel Retina Display	2880x1800		Intel Core i7 2.7GHz	16
4	IPS Panel Retina Display	2560x1600		Intel Core i5 3.1GHz	8

		Gpu	OpSys	Weight	Price_euros	Touchscreen	Ips	\
0	Intel Iris Plus Graphics	640	macOS	1.37	1339.69	0	1	
1	Intel HD Graphics	6000	macOS	1.34	898.94	0	0	
2	Intel HD Graphics	620	No OS	1.86	575.00	0	0	
3	AMD Radeon Pro	455	macOS	1.83	2537.45	0	1	
4	Intel Iris Plus Graphics	650	macOS	1.37	1803.60	0	1	

	cpu_name	gpu_name	Op_system
0	Intel Core i5	Intel	mac
1	Intel Core i5	Intel	mac
2	Intel Core i5	Intel	Other
3	Intel Core i7	AMD	mac
4	Intel Core i5	Intel	mac

```
[83]: data=data.  
      ↪drop(columns=['laptop_ID', 'Inches', 'ScreenResolution', "Product", "Cpu", "Gpu", 'OpSys'],axis=1)
```

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

```
[84]:
```

	Company	TypeName	Ram	Weight	Price_euros	Touchscreen	Ips	\
0	Apple	Ultrabook	8	1.37	1339.69	0	1	
1	Apple	Ultrabook	8	1.34	898.94	0	0	
2	HP	Notebook	8	1.86	575.00	0	0	
3	Apple	Ultrabook	16	1.83	2537.45	0	1	
4	Apple	Ultrabook	8	1.37	1803.60	0	1	

	cpu_name	gpu_name	Op_system
0	Intel Core i5	Intel	mac
1	Intel Core i5	Intel	mac
2	Intel Core i5	Intel	Other
3	Intel Core i7	AMD	mac
4	Intel Core i5	Intel	mac

```
[85]: data=pd.get_dummies(data)
```

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

```
[87]:
```

	Ram	Weight	Price_euros	Touchscreen	Ips	Company_Acer	Company_Apple	\
0	8	1.37	1339.69	0	1	0	1	
1	8	1.34	898.94	0	0	0	1	
2	8	1.86	575.00	0	0	0	0	
3	16	1.83	2537.45	0	1	0	1	

4	8	1.37	1803.60	0	1	0	1
---	---	------	---------	---	---	---	---

	Company_Asus	Company_Dell	Company_HP	...	cpu_name_Intel	Core i5	\
0	0	0	0	...		1	
1	0	0	0	...		1	
2	0	0	1	...		1	
3	0	0	0	...		0	
4	0	0	0	...		1	

	cpu_name_Intel	Core i7	cpu_name_Other	gpu_name_AMD	gpu_name_Intel	\
0		0	0	0		1
1		0	0	0		1
2		0	0	0		1
3		1	0	1		0
4		0	0	0		1

	gpu_name_Nvidia	Op_system_Linux	Op_system_Other	Op_system_Windows	\
0	0	0	0		0
1	0	0	0		0
2	0	0	1		0
3	0	0	0		0
4	0	0	0		0

	Op_system_mac
0	1
1	1
2	0
3	1
4	1

[5 rows x 32 columns]

```
[90]: x=data.drop(columns='Price_euros',axis=1)
```

```
[91]: y=data['Price_euros']
```

```
[92]: from sklearn.model_selection import train_test_split
```

```
[93]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

```
[94]: from sklearn.linear_model import LinearRegression
```

```
[97]: lr_model=LinearRegression()
```

```
[98]: lr_model.fit(x_train,y_train)
```

```
[98]: LinearRegression()
```

```
[99]: lr_model.score(x_test,y_test)
```

```
[99]: 0.7090551485936429
```

```
[100]: from sklearn.linear_model import Lasso  
ls_model=Lasso()
```

```
[102]: ls_model.fit(x_train,y_train)
```

```
[102]: Lasso()
```

```
[103]: ls_model.score(x_test,y_test)
```

```
[103]: 0.7093688012383697
```

```
[104]: from sklearn.tree import DecisionTreeRegressor  
dt_model=DecisionTreeRegressor()
```

```
[105]: dt_model.fit(x_train,y_train)
```

```
[105]: DecisionTreeRegressor()
```

```
[106]: dt_model.score(x_test,y_test)
```

```
[106]: 0.7422635361938353
```

```
[107]: from sklearn.ensemble import RandomForestRegressor
```

```
[118]: rf_model=RandomForestRegressor()
```

```
[119]: rf_model.fit(x_train,y_train)
```

```
[119]: RandomForestRegressor()
```

```
[120]: rf_model.score(x_test,y_test)
```

```
[120]: 0.7908010112911014
```

```
[121]: from sklearn.model_selection import GridSearchCV  
parameters={'n_estimators':[10,50,100],'criterion':  
    ↳['squared_error','absolute_error','poison']}  
grid_obj=GridSearchCV(estimator=rf_model,param_grid=parameters)  
grid_fit=grid_obj.fit(x_train,y_train)  
best_model=grid_fit.best_estimator_  
best_model
```

```
C:\ProgramData\anaconda3\lib\site-  
packages\sklearn\model_selection\_validation.py:378: FitFailedWarning:  
15 fits failed out of a total of 45.
```

The score on these train-test partitions for these parameters will be set to nan.

If these failures are not expected, you can try to debug them by setting `error_score='raise'`.

Below are more details about the failures:

```
-----
15 fits failed with the following error:
Traceback (most recent call last):
  File "C:\ProgramData\anaconda3\lib\site-
packages\sklearn\model_selection\_validation.py", line 686, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\ProgramData\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py",
line 340, in fit
    self._validate_params()
  File "C:\ProgramData\anaconda3\lib\site-packages\sklearn\base.py", line 581,
in _validate_params
    validate_parameter_constraints(
  File "C:\ProgramData\anaconda3\lib\site-
packages\sklearn\utils\_param_validation.py", line 97, in
validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'criterion' parameter
of RandomForestRegressor must be a str among {'poisson', 'friedman_mse',
'squared_error', 'absolute_error'}. Got 'poison' instead.

    warnings.warn(some_fits_failed_message, FitFailedWarning)
C:\ProgramData\anaconda3\lib\site-
packages\sklearn\model_selection\_search.py:952: UserWarning: One or more of the
test scores are non-finite: [0.73622223 0.74618136 0.74924123 0.71424087
0.74874965 0.75215125
      nan      nan      nan]
    warnings.warn(
```

```
[121]: RandomForestRegressor(criterion='absolute_error')
```

```
[122]: best_model.score(x_test,y_test)
```

```
[122]: 0.8001006095130103
```

```
[ ]:
```