



The Superior University

Name:Fatima Nadeem	Roll No:076	Course: PAI Lab
Semester:4rd	Section:4B	Department:SE
Submitted To:Sir Rasikh	Total Marks: 10	Date:

Lab Task 01

Housing Prices

Table of content.

Housing Prices.....	1
Introduction:.....	2
Problem statement:.....	2
Step1:.....	2
Step2:.....	3
Step3:.....	5
Submission on kaggle:.....	6

Introduction:

In this exercise, you will create and submit predictions for a Kaggle competition. You can then improve your model (e.g. by adding features) to apply what you've learned and move up the leaderboard.

Begin by running the code cell below to set up code checkin and the file paths for the dataset.

Problem statement:

It is your job to predict the sales price for each house. For each Id in the test set, you must predict the value of the SalePrice variable.

Step1:

Read the csv file train and test file. By using pandas library.

[1]: import pandas as pd

[2]: train_data=pd.read_csv('train.csv')
train_data

[2]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscV
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	
...
1455	1456	60	RL	62.0	7917	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	
1456	1457	20	RL	85.0	13175	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	
1457	1458	70	RL	66.0	9042	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	GdPrv	Shed	250
1458	1459	20	RL	68.0	9717	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	
1459	1460	20	RL	75.0	9937	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	

1460 rows x 81 columns

[3]: test_data=pd.read_csv('test.csv')
test_data

test_data=pd.read_csv('test.csv')

test_data

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	ScreenPorch	PoolArea	PoolQC	Fence	MiscF
0	1461	20	RH	80.0	11622	Pave	NaN	Reg	Lvl	AllPub	...	120	0	NaN	MnPrv	
1	1462	20	RL	81.0	14267	Pave	NaN	IR1	Lvl	AllPub	...	0	0	NaN	NaN	
2	1463	60	RL	74.0	13830	Pave	NaN	IR1	Lvl	AllPub	...	0	0	NaN	MnPrv	
3	1464	60	RL	78.0	9978	Pave	NaN	IR1	Lvl	AllPub	...	0	0	NaN	NaN	
4	1465	120	RL	43.0	5005	Pave	NaN	IR1	HLS	AllPub	...	144	0	NaN	NaN	
...
1454	2915	160	RM	21.0	1936	Pave	NaN	Reg	Lvl	AllPub	...	0	0	NaN	NaN	
1455	2916	160	RM	21.0	1894	Pave	NaN	Reg	Lvl	AllPub	...	0	0	NaN	NaN	
1456	2917	20	RL	160.0	20000	Pave	NaN	Reg	Lvl	AllPub	...	0	0	NaN	NaN	
1457	2918	85	RL	62.0	10441	Pave	NaN	Reg	Lvl	AllPub	...	0	0	NaN	MnPrv	
1458	2919	60	RL	74.0	9627	Pave	NaN	Reg	Lvl	AllPub	...	0	0	NaN	NaN	

1459 rows x 80 columns

Step2:

EDA Steps ,Preprocessing and Feature engineering.

```
[4]: train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Id                    1460 non-null   int64
1   MSSubClass            1460 non-null   int64
2   MSZoning              1460 non-null   object
3   LotFrontage          1201 non-null   float64
4   LotArea              1460 non-null   int64
5   Street               1460 non-null   object
6   Alley                91 non-null     object
7   LotShape             1460 non-null   object
8   LandContour          1460 non-null   object
9   Utilities            1460 non-null   object
10  LotConfig            1460 non-null   object
11  LandSlope            1460 non-null   object
12  Neighborhood          1460 non-null   object
13  Condition1           1460 non-null   object
14  Condition2           1460 non-null   object
15  BldgType             1460 non-null   object
16  HouseStyle           1460 non-null   object
17  OverallQual          1460 non-null   int64
18  OverallCond          1460 non-null   int64
19  YearBuilt            1460 non-null   int64
20  YearRemodAdd         1460 non-null   int64
21  RoofStyle            1460 non-null   object
22  RoofMatl            1460 non-null   object
23  Exterior1st          1460 non-null   object
24  Exterior2nd          1460 non-null   object
25  MasVnrType           588 non-null    object
26  MasVnrArea           1452 non-null   float64
27  ExterQual            1460 non-null   object
...
70  ScreenPorch          1460 non-null   int64
71  PoolArea             1460 non-null   int64
72  PoolQC               7 non-null      object
73  Fence                281 non-null    object
74  MiscFeature          54 non-null     object
75  MiscVal              1460 non-null   int64
76  MoSold              1460 non-null   int64
77  YrSold              1460 non-null   int64
78  SaleType            1460 non-null   object
79  SaleCondition        1460 non-null   object
80  SalePrice           1460 non-null   int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
```

```
[6]: y=train_data.SalePrice
```

```
[7]: features = ['YearBuilt', 'HouseStyle']
```

```
[8]: import numpy as np
```

```
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
```

```
[9]: def dataEncoder(cols):  
      for i in cols:  
          dataLabelEncoder = LabelEncoder()  
          train_data[i] = dataLabelEncoder.fit_transform(train_data[i])  
  
      columns = ['HouseStyle']  
      dataEncoder(columns)
```

```
[10]: x = train_data[features]
```

```
[11]: x.describe()
```

```
[11]:
```

	YearBuilt	HouseStyle
count	1460.000000	1460.000000
mean	1971.267808	3.038356
std	30.202904	1.911305
min	1872.000000	0.000000
25%	1954.000000	2.000000
50%	1973.000000	2.000000
75%	2000.000000	5.000000
max	2010.000000	7.000000

Step3:

Training the model of svc on two attributes and predicating the prices and completing the Competition.

```
.2]: model_svc = SVC()
model_svc.fit(x, y)

print(model_svc)

SVC()
```

```
.3]: features_test = ['YearBuilt', 'HouseStyle']
```

```
.4]: def dataEncoder(cols):
    for i in cols:
        dataLabelEncoder = LabelEncoder()
        test_data[i] = dataLabelEncoder.fit_transform(test_data[i])

    columns = ['HouseStyle']
    dataEncoder(columns)
```

```
.5]: features_test = ['YearBuilt', 'HouseStyle']
```

```
.6]: o= test_data[features_test]
```

```
.7]: df = pd.DataFrame(o)

# Save the DataFrame as a new CSV file
df.to_csv(r'test88.csv', index=False, header=True)

print("New CSV file created successfully!")

New CSV file created successfully!
```

```
!1]: test10_data=pd.read_csv('test88.csv')
```

```
!]: test10_data=pd.read_csv('test88.csv')
```

```
!]: model_predictions = model_svc.predict(test10_data)
```

```
!]: submission9=pd.DataFrame({'HouseStyle':test10_data['HouseStyle'],'SalePrice':model_predictions})
submission9.to_csv('submission9.csv',index=False)
print("submission successfully")

submission successfully
```

```
!]: submission10=pd.DataFrame({'ID':m['Id'],'SalePrice':model_predictions})
submission10.to_csv('submission10.csv',index=False)
print("submission successfully")



submission successfully
```

Submission on kaggle:

Submissions

- All
- Successful
- Errors

Recent ▼

Submission and Description		Public Score ⓘ
<div></div> <div>submission10.csv Complete · 2mo ago</div>		62283.69135
<div></div> <div>submission9.csv Error · 2mo ago</div>		