Artificial Intelligence

Project Phase 0 Report

Mohammad Mahdi Islami
Student No. 810195548

a. آشنایی اولیه با داده ها

قطعه کدی که به منظور رسم نمودار ها نوشته شده به صورت زیر است:

```python
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np


def solve_equation(A,B,C,E,F,G):
    #Eq1 => Aw + Bb = C
    Eq1 = np.array([[A,B], [E,F]])
    Eq2 = np.array([C,G])
    Ans = np.linalg.solve(Eq1, Eq2)
    return Ans


df = pd.read_csv("houses.csv")

#Drop the categorical datas.
df = df.drop(columns=['LotConfig','Neighborhood'])
df = df.fillna(df.mean())


df.plot.scatter(x='MSSubClass', y='SalePrice',c='Blue',title='Year Built')
df.plot.scatter(x='LotArea', y='SalePrice',c='Blue',title='Lot Area')
df.plot.scatter(x='OverallQual', y='SalePrice',c='Blue',title='Overall Qual')
df.plot.scatter(x='LotFrontage', y='SalePrice',c='Blue',title='Lot Frontage')
df.plot.scatter(x='OverallCond', y='SalePrice',c='Blue',title='Overall Cond')
df.plot.scatter(x='BedroomAbvGr', y='SalePrice',c='Blue',title='Bedroom AbvGr')
df.plot.scatter(x='TotRmsAbvGrd', y='SalePrice',c='Blue',title='Tot RmsAbvGrd')
df.plot.scatter(x='TotalBsmtSF', y='SalePrice',c='Blue',title='Total BsmtSF')
df.plot.scatter(x='YearBuilt', y='SalePrice',c='Blue',title='Year Built')


Ans = solve_equation(np.sum(np.square(df['OverallQual'])), np.sum(df['OverallQual']),
                     np.sum(df['OverallQual']*df['SalePrice']), np.sum(df['OverallQual']),
                     1133, np.sum(df['SalePrice']))

print(Ans)

x = df['OverallQual']
y = Ans[0]*x + Ans[1]

df.plot.scatter(x='OverallQual', y='SalePrice',c='Blue',title='Overall Qual')
plt.plot(x, y, '-r')
plt.grid()
plt.show()

Rmse = np.sqrt(np.square(np.subtract(Ans[0]*df['OverallQual'] + Ans[1],df['SalePrice'])).mean())
print(Rmse)
```

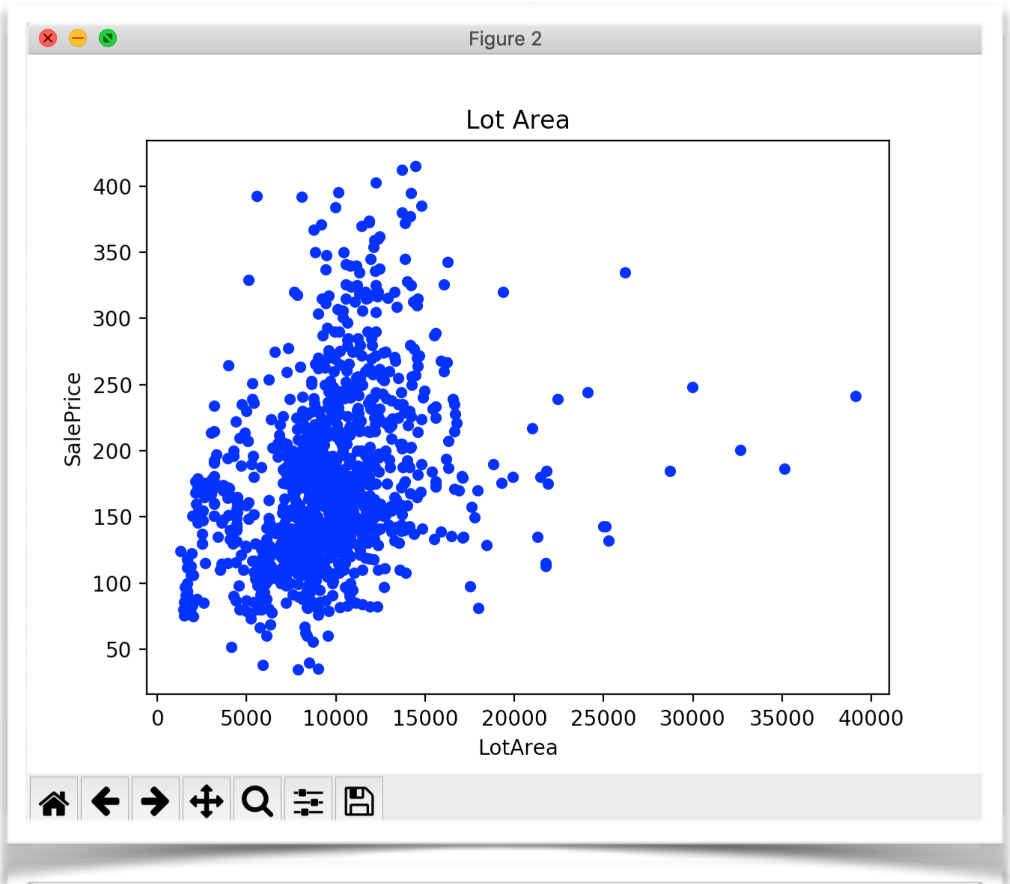

Fig1. Plot **SalePrice vs LotArea**

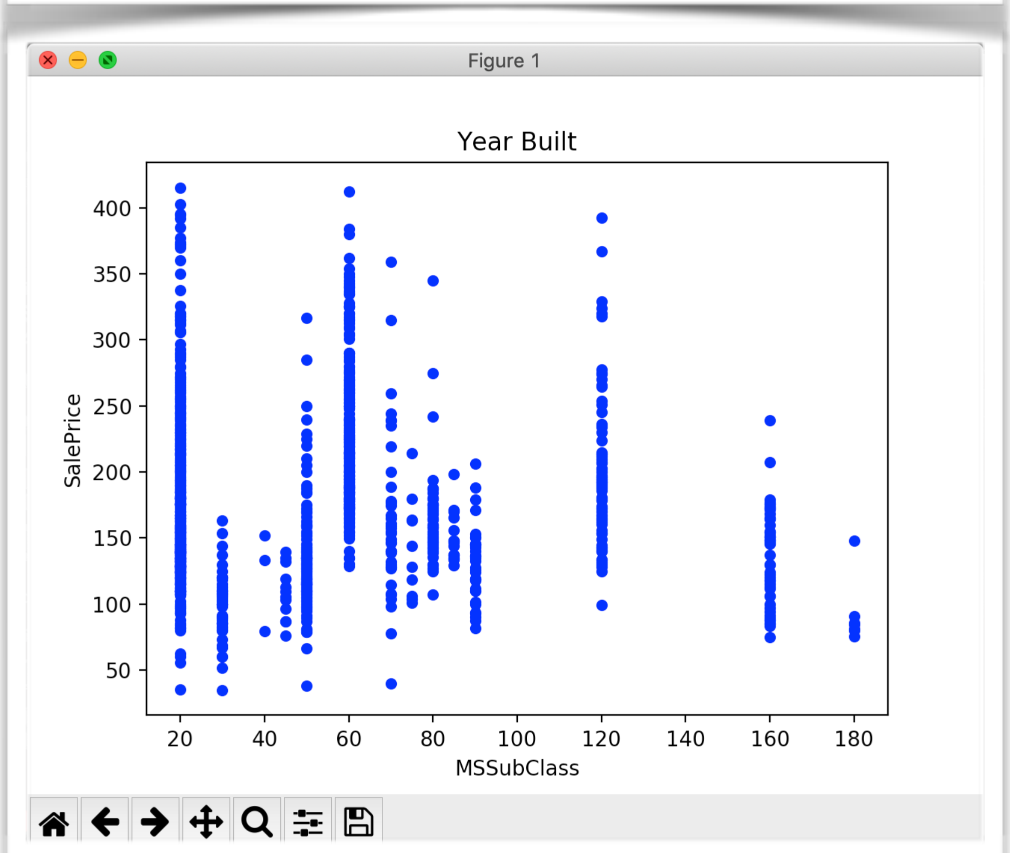

Fig2 Plot **SalePrice vs YearBuilt**

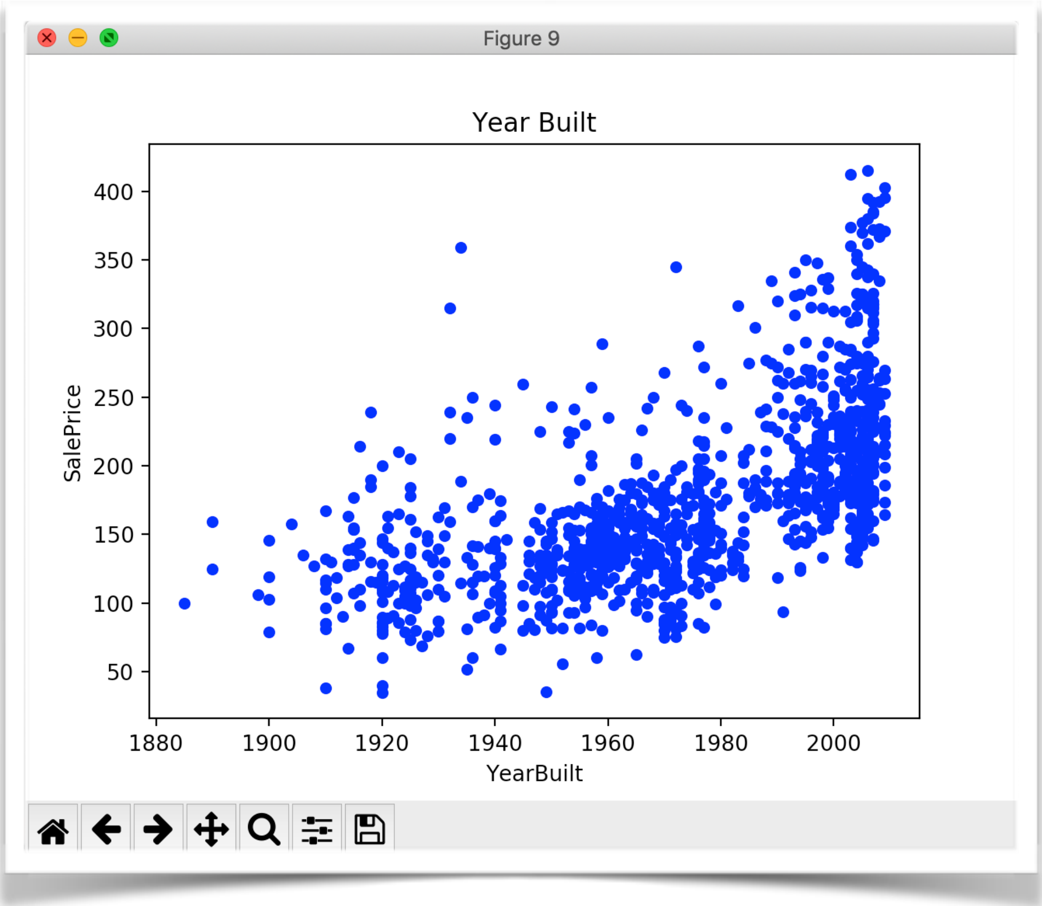

Fig3. Plot **SalePrice vs Year Built**

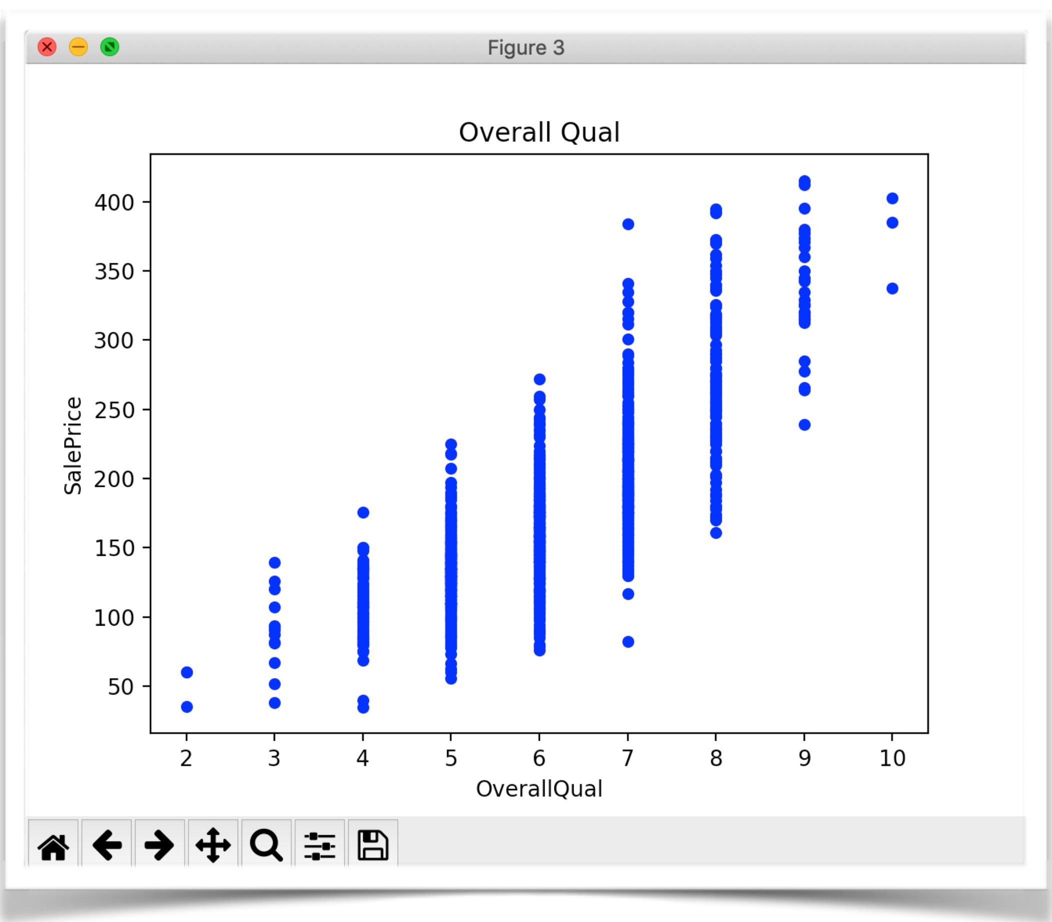

Fig4. Plot **SalePrice vs OverallQual**

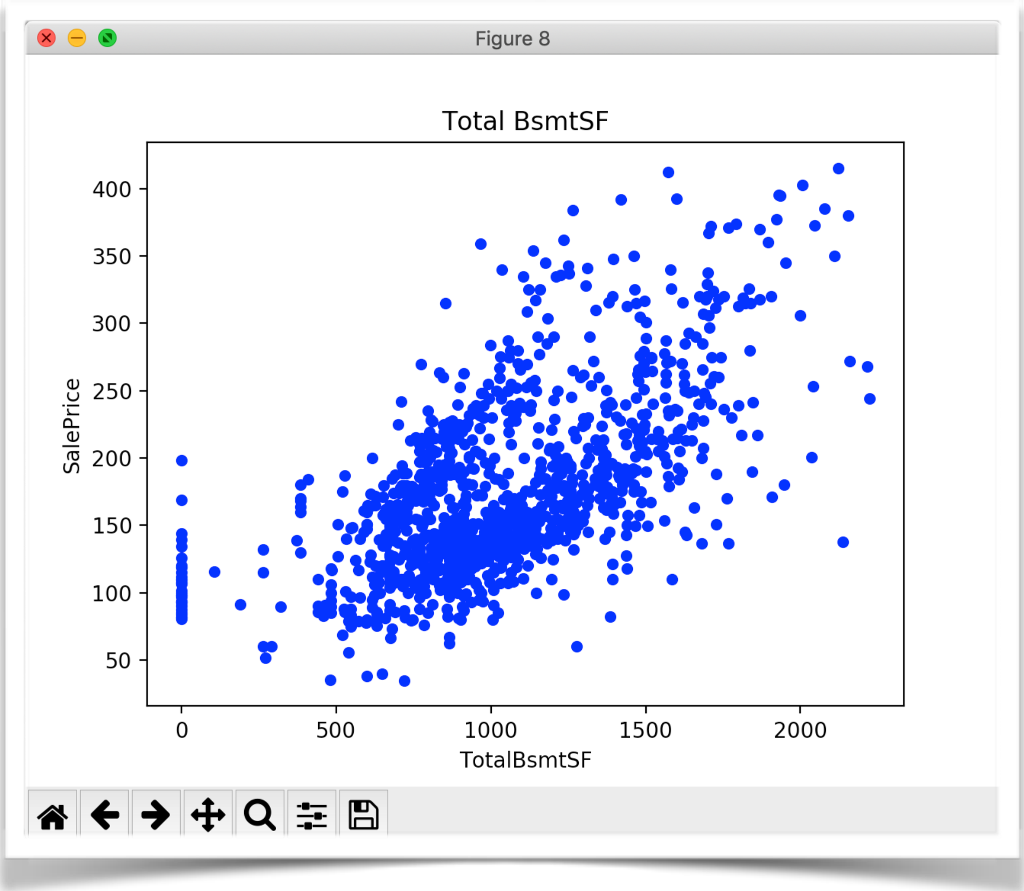

Fig5. Plot **SalePrice vs BsmSF**

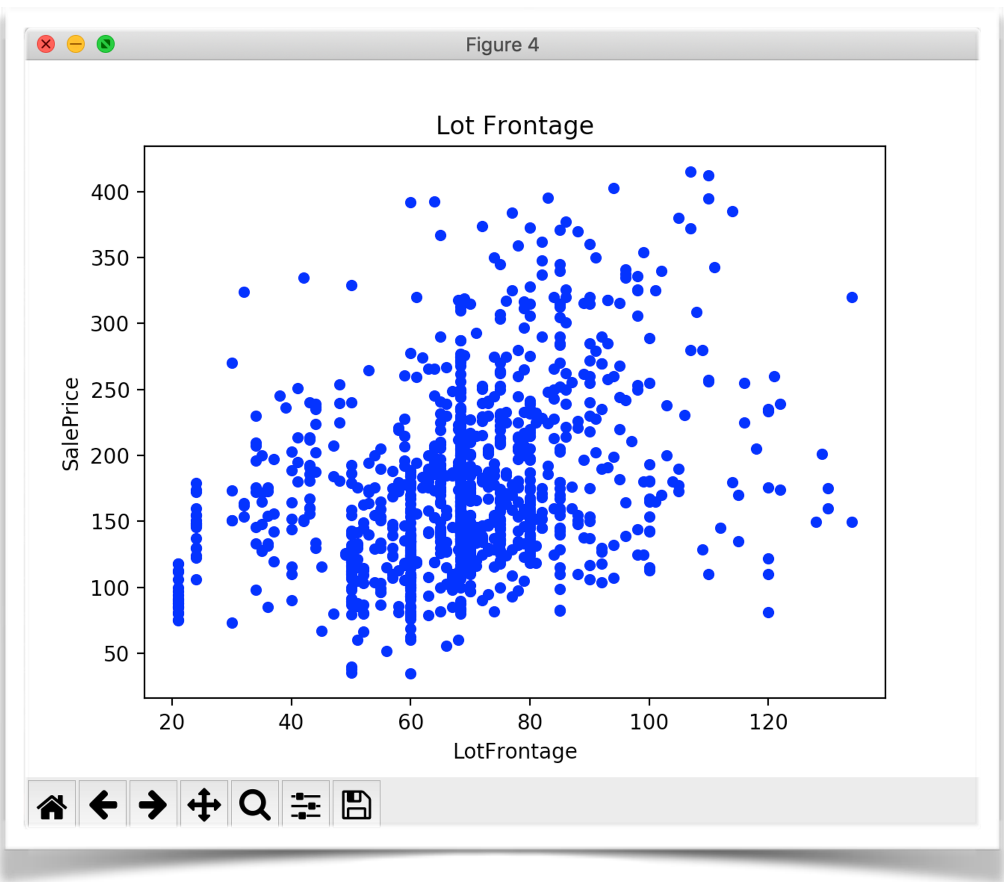

Fig6. Plot **SalePrice vs LotFrontage**

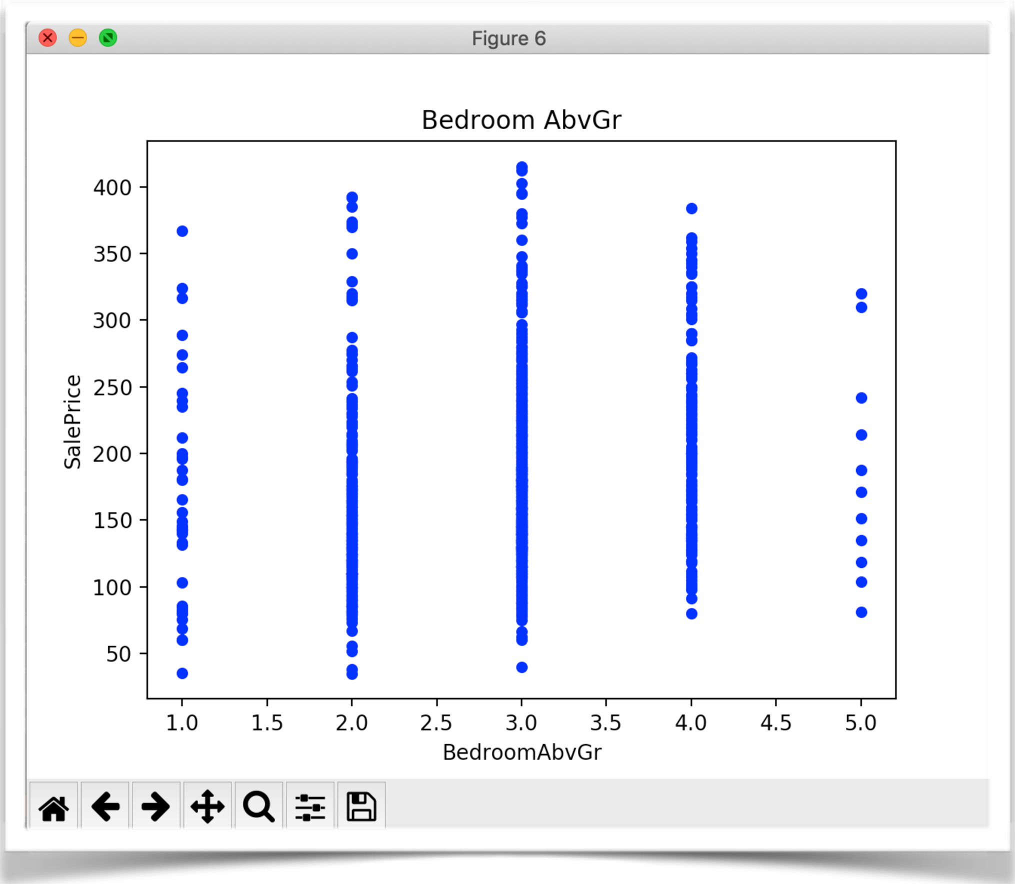

Fig7. Plot **SalePrice vs Bedroom AbvGrd**

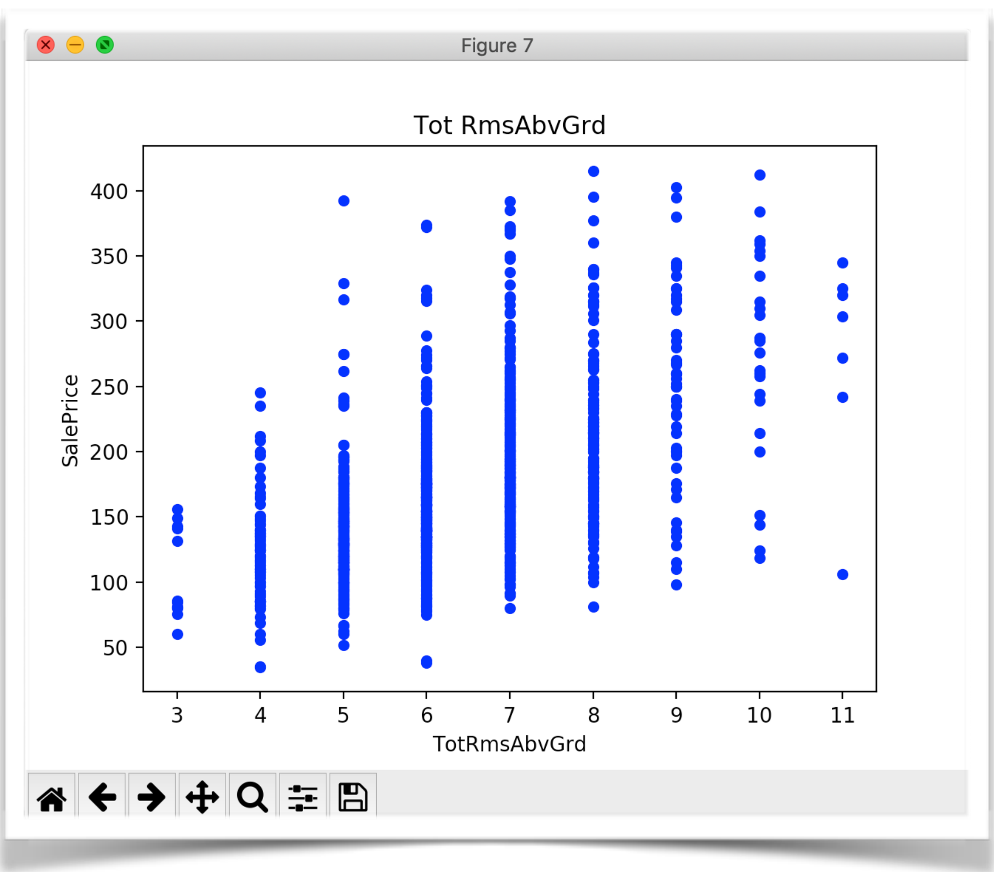

Fig8. Plot **SalePrice vs Tot RmsAbvGrd**

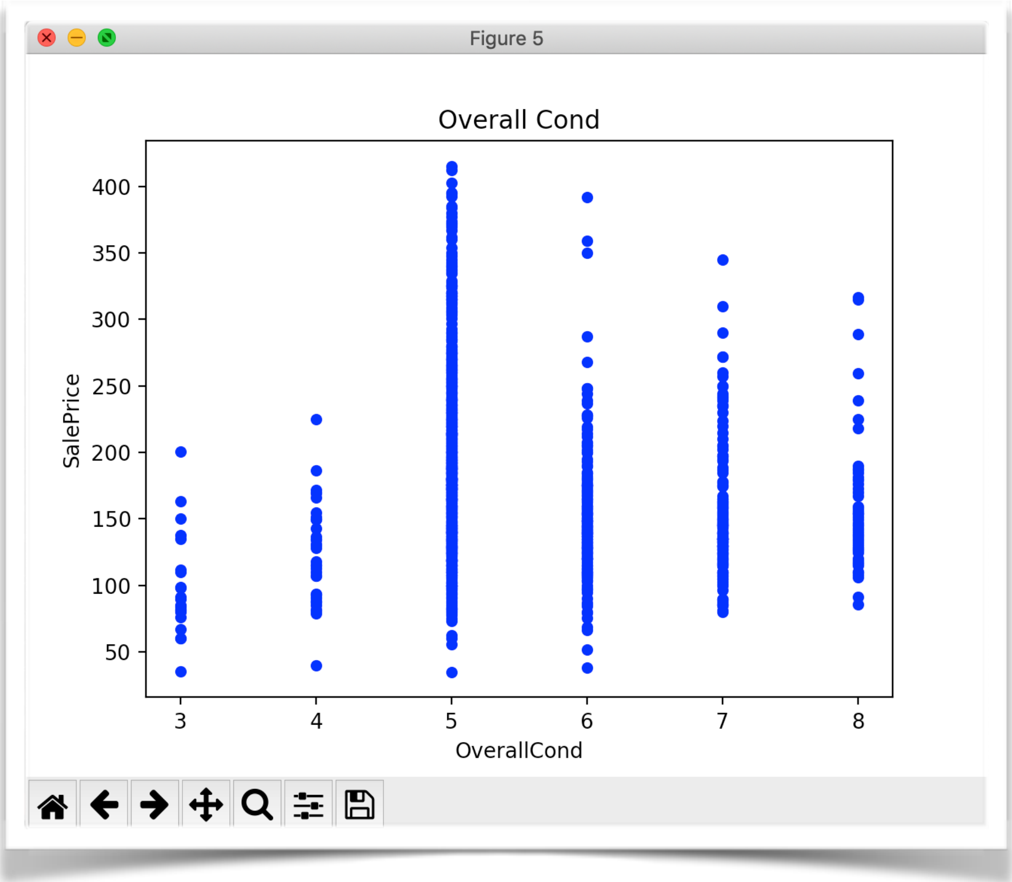

Fig9. Plot **SalePrice vs Overall Cond**

مقادیر به دست آمده برای L rmse با مقادیر ستون های مختلف دیتافریم

MSSubClass
[02+1.77271990e 02-4.31348617e-]
65.38453639003524

LotArea
[5.58012398e-03 1.21951289e+02]
61.650314655270805

Lot Frontage
[1.30982549 85.25980132]
60.74348767764437

Overall Cond
[-9.50029829 227.72230868]
64.84900887423757

BedroomAbvGr
[18.29119825 123.14813241]
63.91979790611292

TotRmsAbvGrd
[24.81374092 17.12019498]
54.72251470597509',

TotalBsmtSF
[0.104487   67.00858358]
51.43029609703544

YearBuilt
[-3.16005452e-01 7.98961406e+02]
71.2429241935347

Overall Qual
[W = **41.08498196** , b = **-74.47264601**]
L rmse = **38.54794652807224**

Minimum L(rmse)

طبق محاسبات انجام شده نموداری که مولفه ی افقی آن را Overall Qual تشکیل میدهد داری مقدار کمینه L rmse است بنابر این نموداری که بیشتر رفتار خطی دارد را این نمودار در نظر میگیریم.

فی الواقع برای به دست آوردن W و b در معادله خطی از فرمول Lrmse مشتق ضمنی نسبت به W و b گرفته برابر صفر قرار میدهیم تا مقدار کمینه خط را به ما بدهد:

$$\hat{y} = wx + \quad b$$

$$L_{RMSE}(\hat{y}, y) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}$$

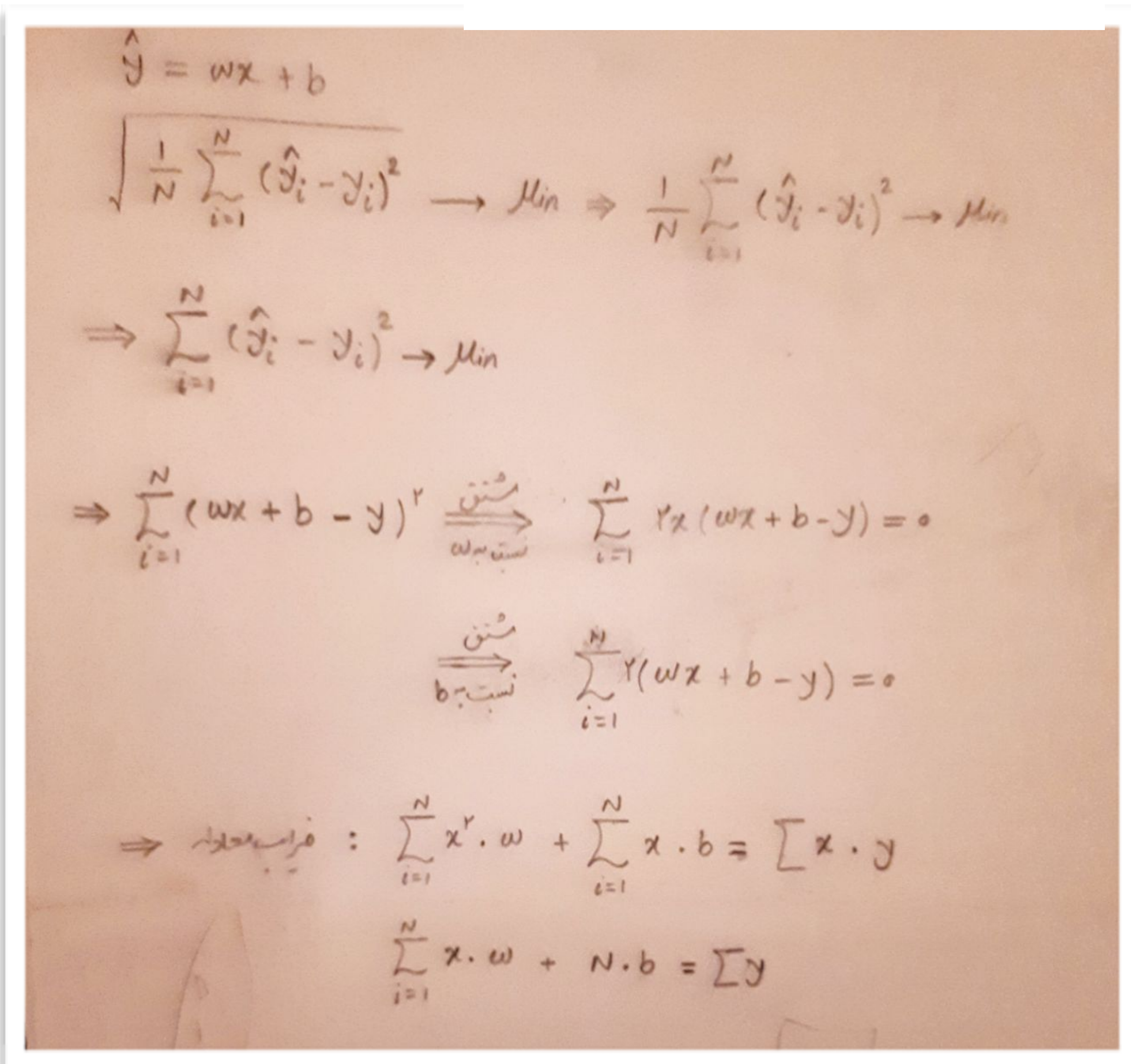

Fig10. مشتق تابع برای به دست آوردن مینیمم

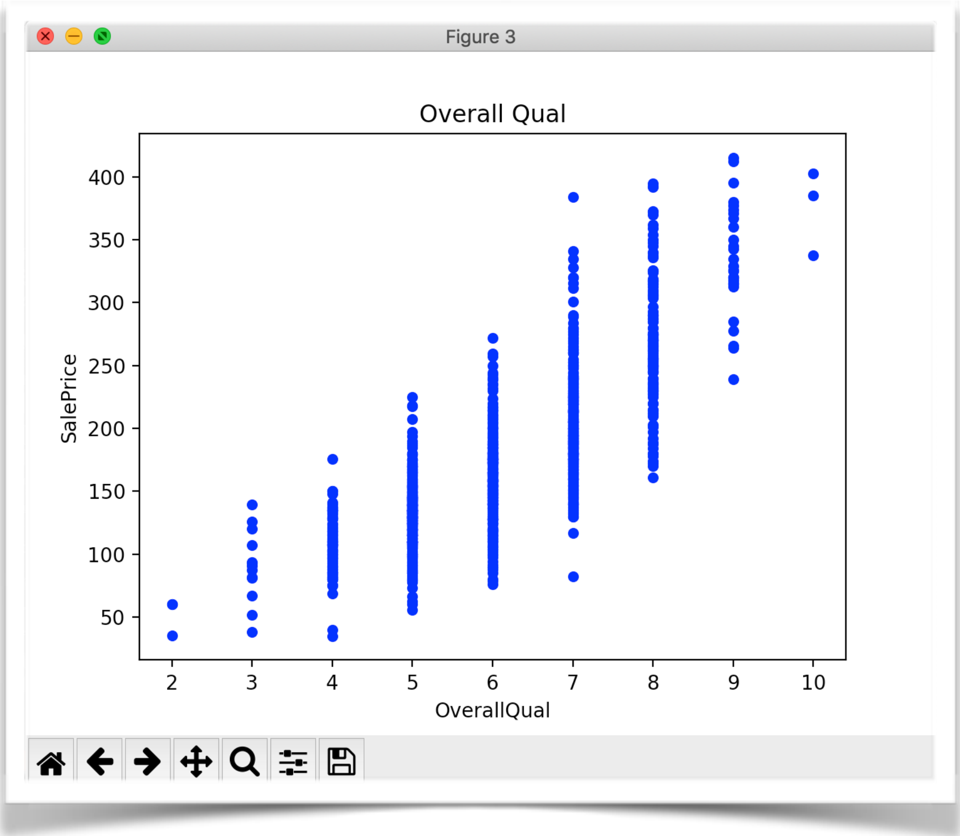

Fig10. Overall Qual Figure

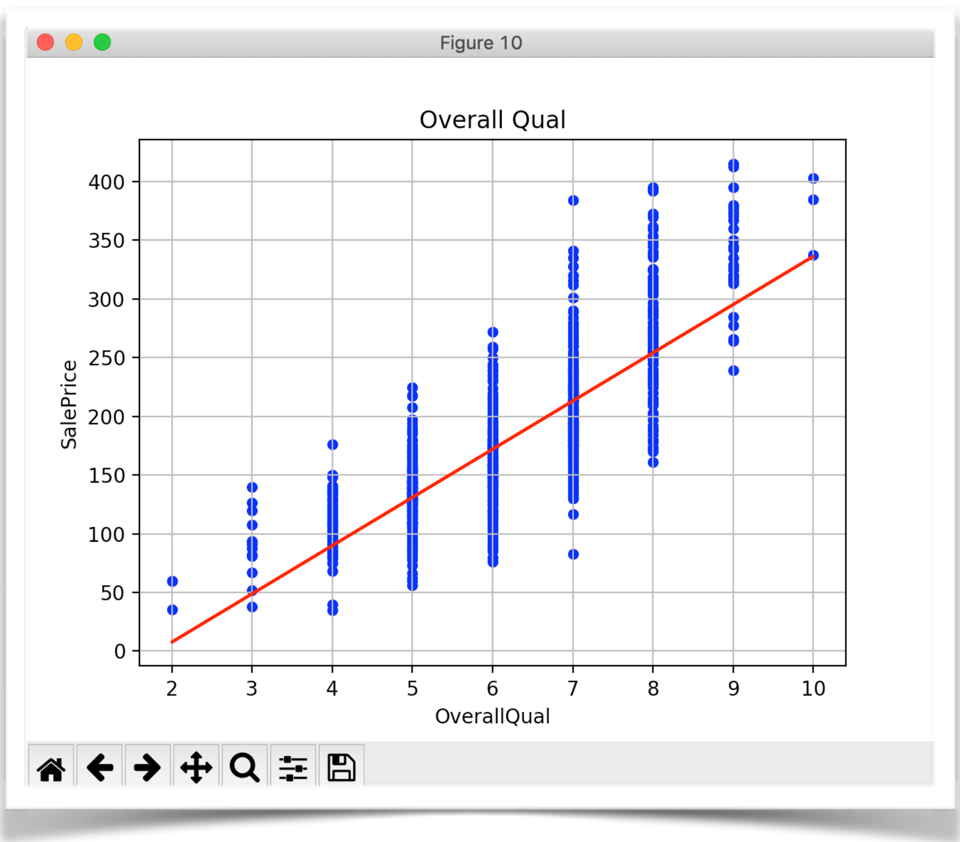

Fig10. Overall Qual Figure With predicted Line

برای محاسبه Knn طبق فرمول گفته شده در لینک راهنما ابتدا داده ها را استاندارد میکنیم تا تاثیر داده های با مقیاس بزرگ اثر داده ها با مقیاس کوچک را از بین نبرد.

$$X_s = \frac{X - Min}{Max - Min}$$

سپس با استفاده از فرمول اقلیدسی فاصله دو نقطه فاصله ی خانه ی سفارشی را با خانه های دیتافریم اندازه گیری کرده و با توابع موجود در numpy از آنها ۱۰ خانه نزدیک به خانه سفارشی را انتخاب کرده و قیمت آن ها را میانگین میگیریم. باید توجه داشته باشیم که به جای دو بعد از ۹ بعد برخورداریم و مجموع تفاضلات این ۹ بعد متناظر با هم ملاک اندازه گیری ماست.

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

```python
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

def read_dataframe(filename):
    idf = pd.read_csv(filename)
    idf = idf.drop(columns=['LotConfig','Neighborhood'])
    idf = idf.fillna(idf.mean())
    return idf

def standardalize_dataframe(df):
    sdf = df - df.min()
    sdf = sdf / (sdf.max() - sdf.min())
    return sdf

data = [70,11435,8,67.66037735849056,7,3,7,792,1929]
idf = pd.DataFrame([data],columns=['MSSubClass', 'LotArea', 'OverallQual', 'LotFrontage', 'OverallCond',
            'BedroomAbvGr', 'TotRmsAbvGrd', 'TotalBsmtSF', 'YearBuilt'] )

df = read_dataframe("houses.csv")

SalePrice = df['SalePrice']
Id = df['Id']

sdf = standardalize_dataframe(df)
sdf = sdf.drop(columns=['Id','SalePrice'])
ddf = df.drop(columns=['Id','SalePrice'])

idf = idf - ddf.min()
idf = idf / (ddf.max() - ddf.min())

def predict(dataframe):
    distance = np.square(np.subtract(sdf, dataframe.iloc[0]))
    Ans = distance.sum(axis=1)
    Ans = np.sqrt(Ans)
    Min = idx = np.argpartition(Ans, 10)
    Ans1 = SalePrice[idx[:10]]
    avg = Ans1.mean()
    return avg

Price = predict(idf)
print(Price)
```