CAPSTONE PROJECT 1: HOUSE PRICE PREDICTIONS

Mehmet KETENCI

WHAT IS TARGET OF PROJE?

In this project, our aim is to predict house prices in certain places of Washington area to give very clear perspective to both sellers and buyers.

WHAT DOES IT INCLUDE?

Data Wrangling: Checking the data to clean with preprocessing and converting raw data to desired format to be used for future processes.

EDA: Deeping into behind to reasons by visualizations.(Seaborn, matplotlib)

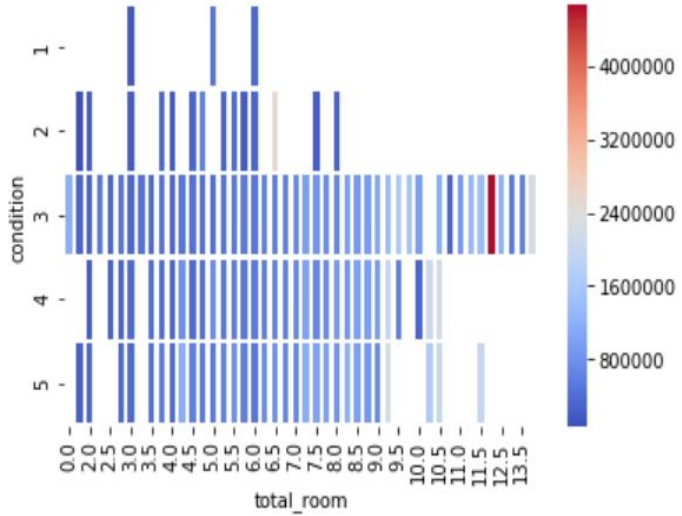
Statistical Inferences: Featuring especially things will be searched in previous steps (Data Wrangling and EDA) using Z-T table with OLS chart

Machine Learning: Finalize ‘House Price Prediction’ with Machine Learning algorithms by using supervised learning models to predict house prices in this chapter since the target data is a numerical column.

TOTAL ROOM-CONDITION EFFECT ON PRICE

```
df = h.pivot_table(index='condition',columns='total_room',values='price')
sns.heatmap(df,cmap='coolwarm',linewidths=1.5)
```

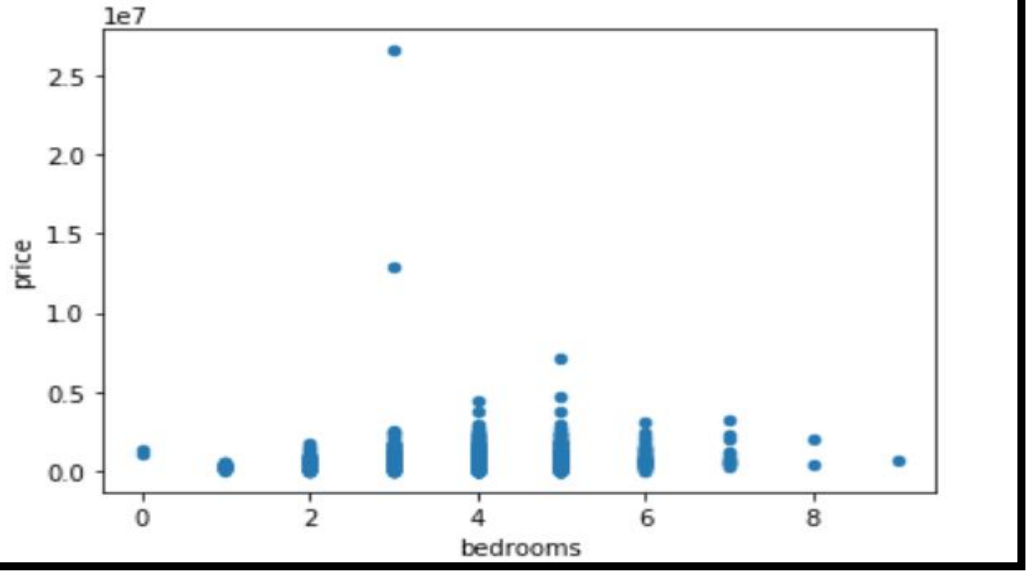
```
<matplotlib.axes._subplots.AxesSubplot at 0x22f03764dd8>
```



In this graph below, we can see the triple relationships among price-#total room-condition.There is a home with dark orange color which condition is 3 and its price is pretty bigger than average.This home price needs to be reviewed in detail. We see the same issue for the home with condition is 2.It is also way above the average and required what the reason behind is.

OUTLIERS DEPEND ON BEDROOM NUMBERS

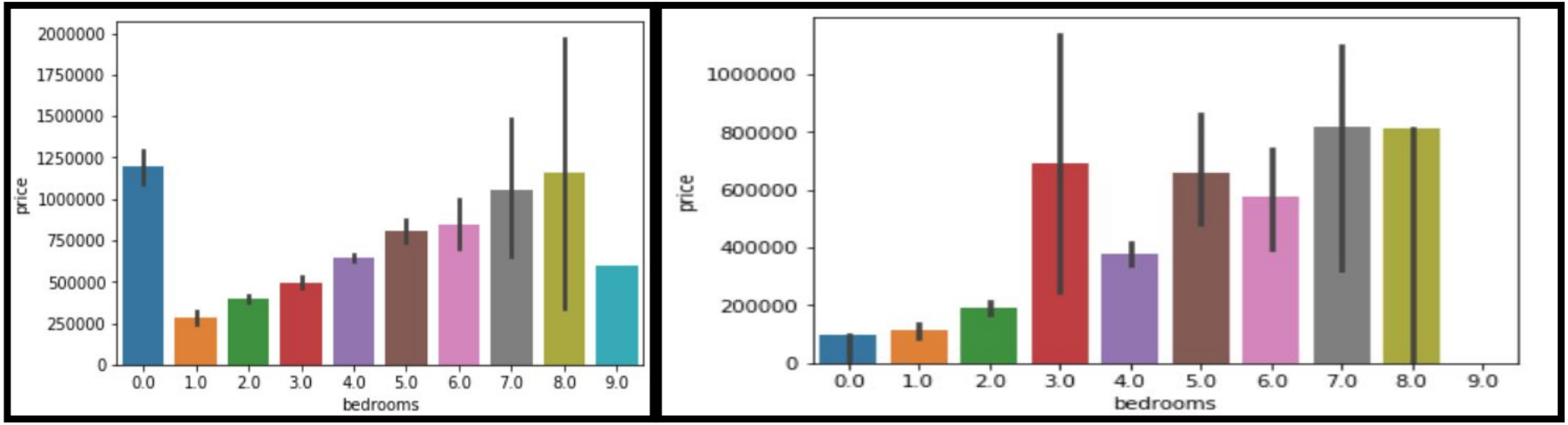
```
house.plot(kind='scatter',x='bedrooms',y='price')
plt.show()
```



There is a normality between bedroom and price relationship graph. It also has some outliers. (bedroom number:3)

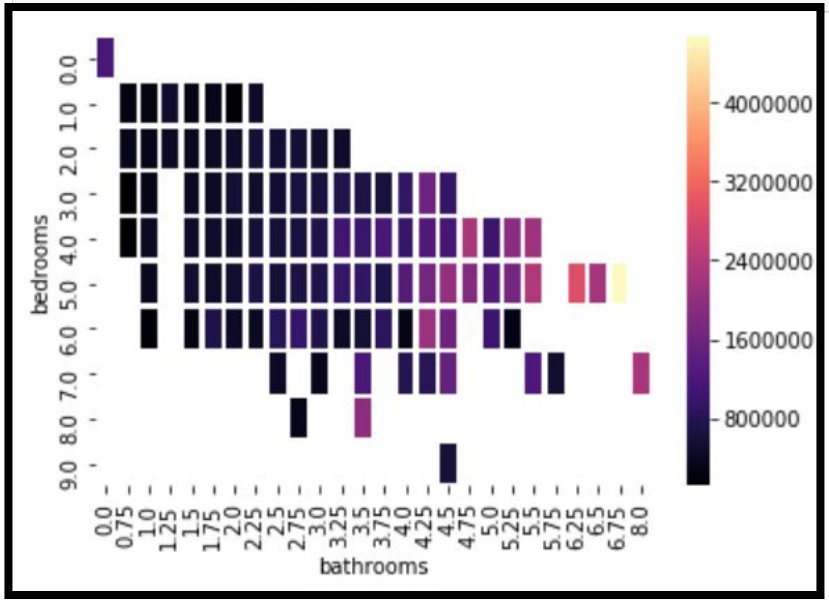
Bedroom-Price RELATIONSHIP

```
h=new_house
h[h.price >= new_house['price'].mean()]
h
sns.barplot(x='bedrooms',y='price',data=h,estimator = np.std)
```



Bedroom and Bathroom effect on price with heatmap

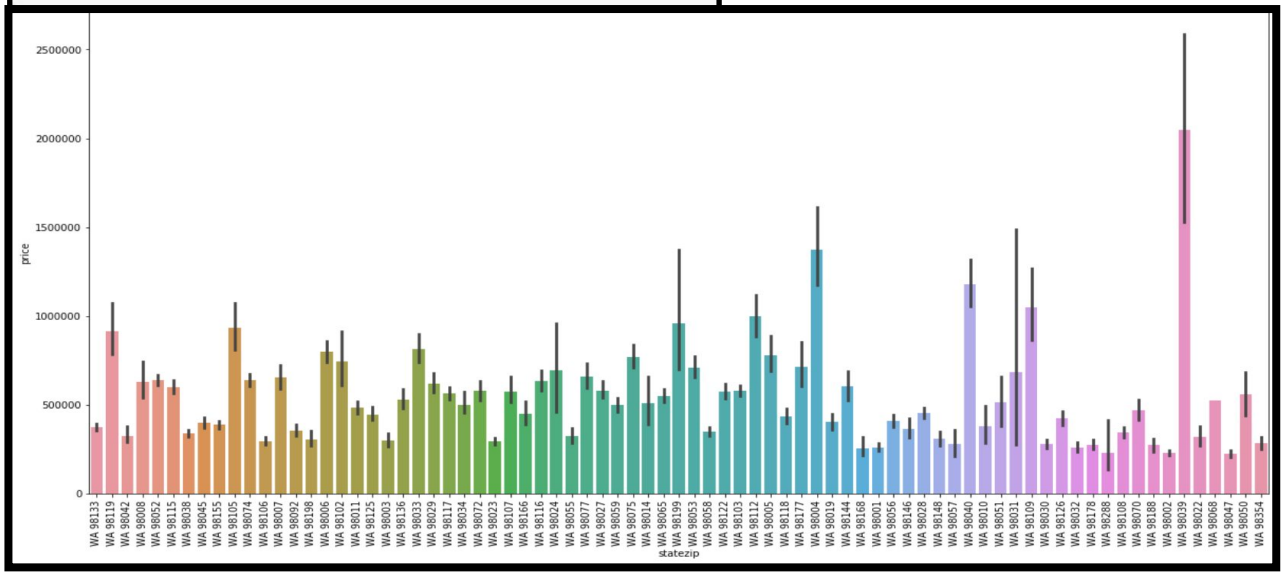
```
df = h.pivot_table(index='bedrooms',columns='bathrooms',values='price')
sns.heatmap(df,cmap='magma',linewidths=1.5)
```



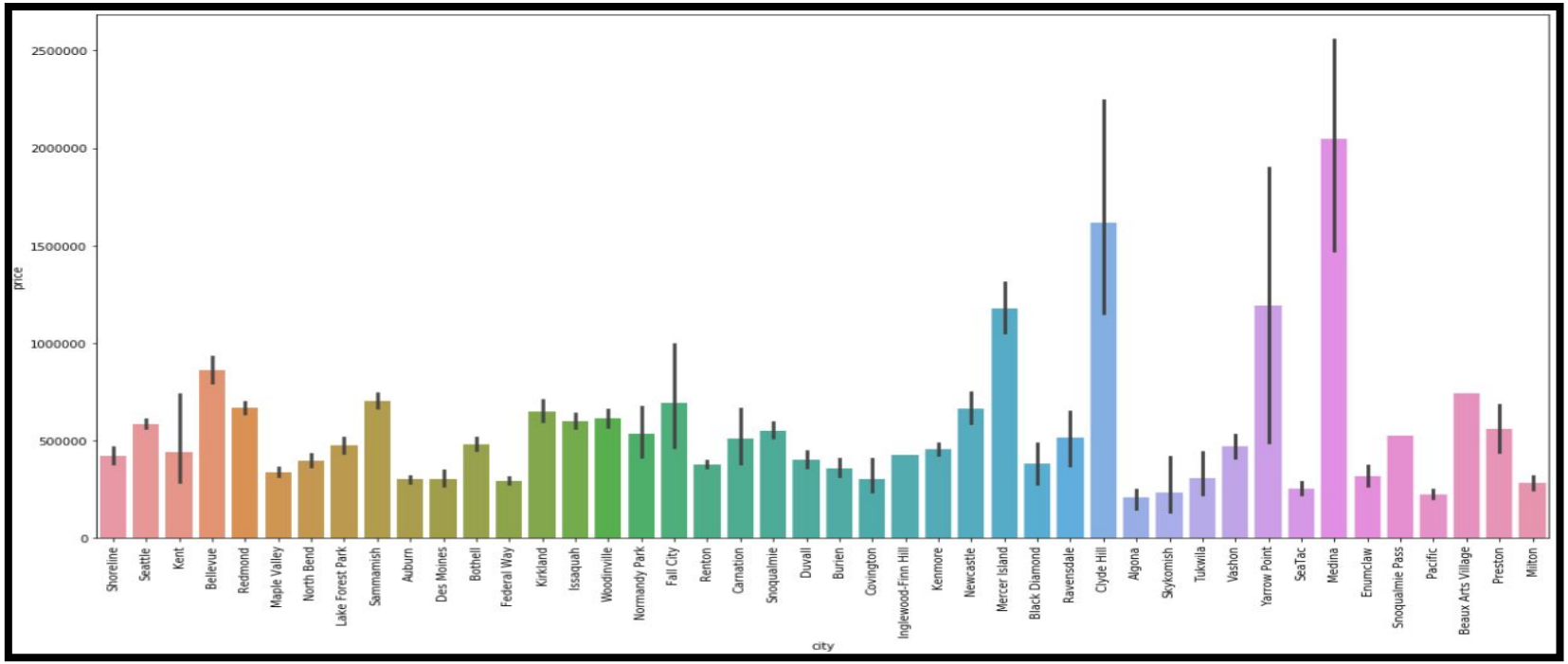
We observe low prices with lower number of bedroom and bathroom.Price is directly proportional with combine bedroom and bathroom numbers. There are outliers but probably they might be impacted by other items as well such as condition,view,location etc...

Prices by locations(Zip codes)

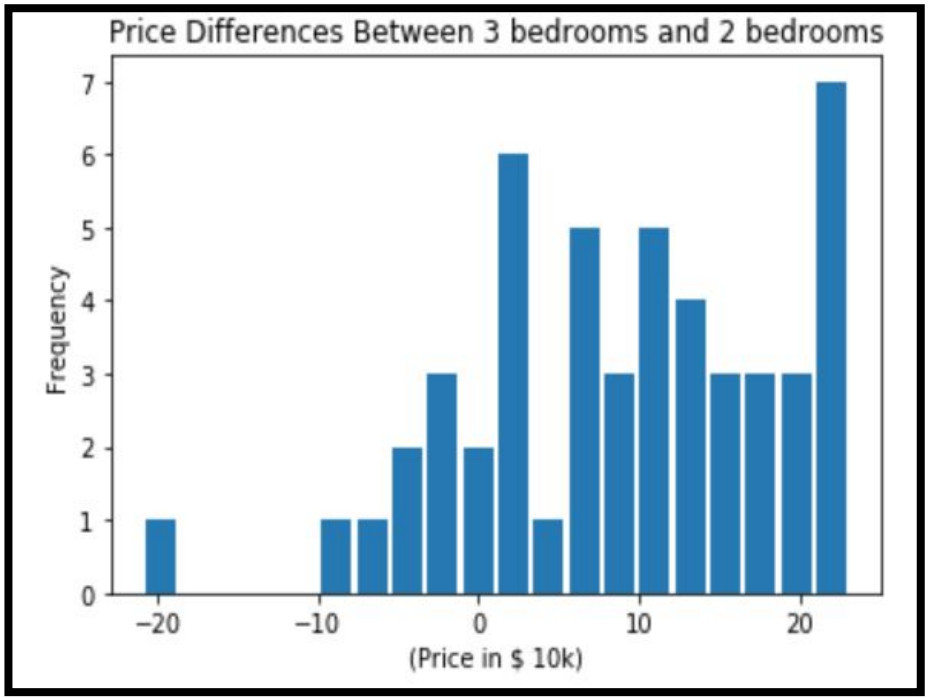
```
plt.figure(figsize=(20,10))
sns.barplot(x='city',y='price',data = h)
plt.xticks(rotation = 'vertical')
plt.show()
```



Prices by cities

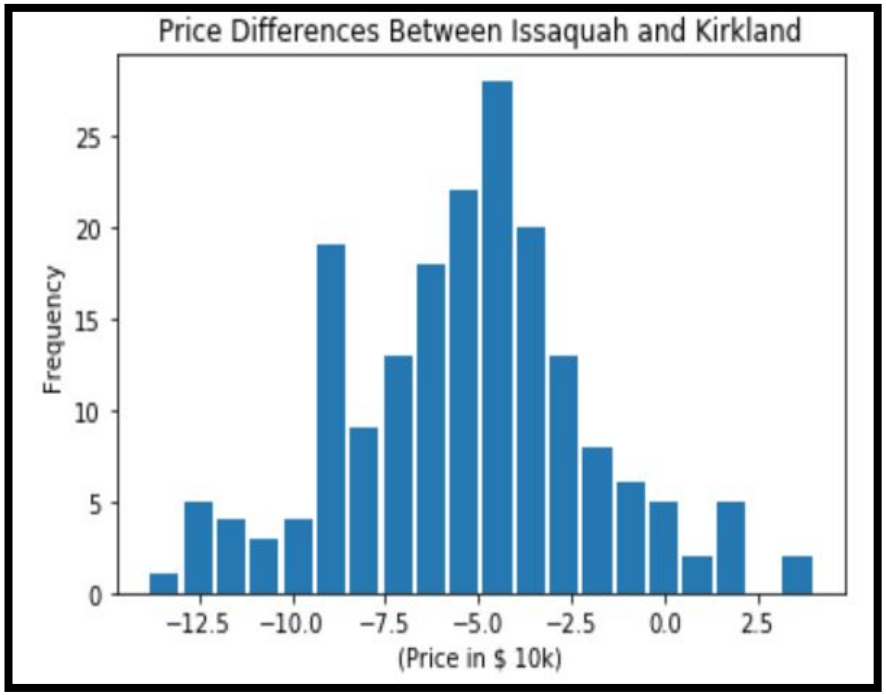


STATISTICAL APPROACH



There is a significant difference between mean price of homes with 2 bedrooms and mean price of homes with 3 bedrooms.

Comparing Prices Between Two Cities



As you can see on the graph,major data gathered below 0 which mean of home prices in Issaquah is lower than mean of home prices in Kirkland.

Coming A Family To The Region

Wederson Family moves from California to Washington. They search for a home that has 2 bathrooms and 3 or 4 bedrooms in new area.
What is the probability of they rent a home with condition is not lower than 4 (with Ms. Wederson`s demand) of given options above?(Solve by using Bayes Theorem)

PROBABILITY OF GETTING HOME THEY WISH IS...

```
X = house[((house.bedrooms==3)|(house.bedrooms==4))&(house.bathrooms==2)]

print('P(X):{}{}{}'.format(len(X),'/',4600))
```

```
P(X):355/4600
```

Question: How many home conditions does have greater than or equal to 4?

```
A = house[house.condition>=4]

print('P(A):{}{}{}'.format(len(A ),'/',4600))
```

```
P(A):1687/4600
```

```
#Let`s find out of up_4 how many homes are there with 3 or 4 bedrooms and 2 bathrooms.
#P(X|A)=?
X_A = A[((A.bedrooms==3)|(A.bedrooms==4))&(A.bathrooms==2)]
print('P(X|A):{}{}{}'.format(len(X_A),'/',1687))
```

```
P(X|A):177/1687
```

```
   Now, I am plug in values into Bayes formula

   P(A|X)=(P(X|A)*P(A))/P(X)     (BAYES T.)
```

```
(((177/1687)*(1687/4600))/(355/4600))*100
```

```
49.859154929577464
```

Probability of Wederson Family choosing homes they wish is 49.9%.

It seems they are lucky, 1 out of 2 homes is the home they are looking for and they can easily access or find those homes by filtering method in any home selling websites.

SETTING BASELINE MODEL

LINEAR REGRESSION

```
#Normalizing features

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()
X_scaled = sc.fit_transform(X_train)
y_scaled = y_train
X_tested = sc.transform(X_test)
y_tested = y_test
```

```
#Building a baseline model - Linear Regression

from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_scaled,y_scaled)
y_pred = lr.predict(X_tested)
```

```
r2_score(y_tested,y_pred)
```

```
0.5473440758579227
```

FEATURE SELECTION

UNIVARIATE SELECTION

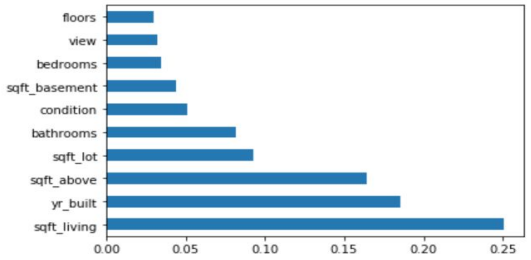
```
from sklearn.ensemble import ExtraTreesRegressor
model = ExtraTreesRegressor(n_estimators=100)
model.fit(X,y)
```

```
ExtraTreesRegressor(bootstrap=False, criterion='mse', max_depth=None,
                    max_features='auto', max_leaf_nodes=None,
                    min_impurity_decrease=0.0, min_impurity_split=None,
                    min_samples_leaf=1, min_samples_split=2,
                    min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=None,
                    oob_score=False, random_state=None, verbose=0,
                    warm_start=False)
```

```
model.feature_importances_
```

```
array([0.03421333, 0.08164022, 0.25061447, 0.09248207, 0.0296628 ,
       0.01333475, 0.03190069, 0.0510895 , 0.16417404, 0.04413165,
       0.18532957, 0.02142691])
```

```
feat_importances = pd.Series(model.feature_importances_,index = X.columns)
feat_importances.nlargest(10).plot(kind='barh')
plt.show()
```



We searched for effect of other columns on price by feature selection.

CREATING NEW FEATURES

PREPROCESSING POLYNOMIAL FEATURES

```
from sklearn.preprocessing import PolynomialFeatures
```

```
poly_features = PolynomialFeatures(interaction_only=True)
X_train_poly = poly_features.fit_transform(X_train)
X_test_poly = poly_features.fit_transform(X_test)
poly_model = LinearRegression()
poly_model.fit(X_train_poly,y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
y_test_predict = poly_model.predict(poly_features.fit_transform(X_test))
r2_score(y_test,y_test_predict)
```

```
0.5835449417335691
```

```
sc2 = StandardScaler()
X_poly_scaled = sc2.fit_transform(X_train_poly)
y_scaled = y_train
X_poly_tested = sc2.transform(X_test_poly)
y_tested = y_test
```

```
lin_reg = LinearRegression()
lin_reg.fit(X_poly_scaled,y_scaled)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
y_poly_pred = lin_reg.predict(X_poly_tested)
r2_score(y_tested,y_poly_pred)
```

```
0.5921726562728207
```

Model was pretty improved by preprocessing with polynomial features compared to started.

We then enrich the model with Ridge Regression.

RIDGE REGRESSION

RIDGE REGRESSION

```
from sklearn.linear_model import Ridge
ridge = Ridge(alpha=3.7,random_state=38)
ridge.fit(X_poly_scaled,y_train)
```

```
Ridge(alpha=3.7, copy_X=True, fit_intercept=True, max_iter=None,
      normalize=False, random_state=38, solver='auto', tol=0.001)
```

```
y_ridge_pred = ridge.predict(X_poly_tested)
r2_score(y_test,y_ridge_pred)
```

```
0.599859165492652
```

In general, with more columns for each observation, we'll get more information and the model will be able to learn better from the dataset and therefore, make better predictions. We have gotten much better results with this small dataset which includes 4600 rows. We actually started with 0.25 R2 score and pulled up to 0.60 with ML algorithms.Ridge regression is able to explain 60% of the variability in predicting house prices based on the input features.

THANK YOU