Ziyao Guo and HaoMin Wang

Professor : Alan Burstein

CS521 Information Structures with Python

12/16/2021

Forecast Analysis Report of House Prices

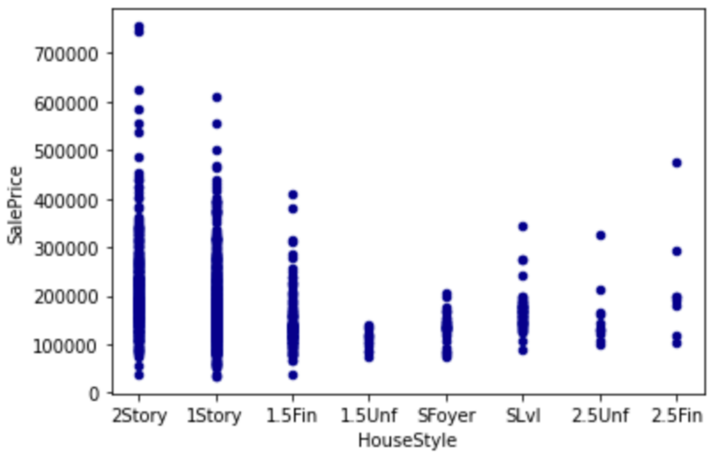
Introduction:

For the final project we choose the topic is house price forecasting. In the whole process of this model construction, we preprocessed the data and extracted key features to analyze the house price through visualization and pre-training of the basic model. It is of great significance for building an effective housing price prediction model for the financial market and people's livelihood.

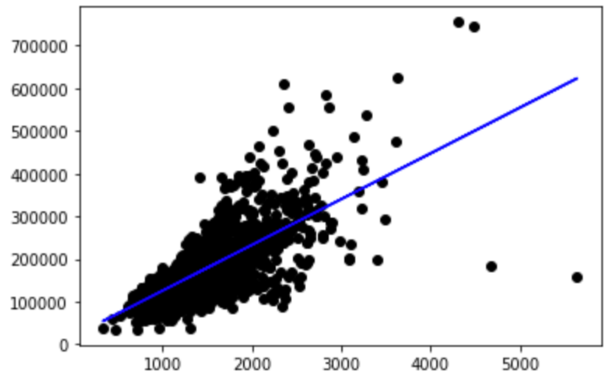
Data and Model Analysis:

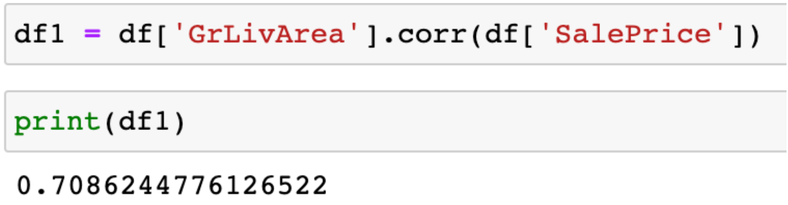
First, we quickly extract some basic housing information from the folder. Through tables, charts and models show the sales price and housing type of comparative data information, which are the key information that customers are concerned about buying a house. I'll focus on a few graphs for the following analysis.

The following scatterplot illustrates the relationship between the type of dwelling style and the sales price. We can see that 2story and 1story are among the best prices, and it is also a potential illustration of the popularity of 2story-style houses, with prices reaching up to $700,000.



The following linear regression is the relationship between the two variables of ground living area square feet and sales price. All the points in this scatter plot are basically fitted very closely, indicating that the ground living area is positively correlated with the price of the house. Through the two sets of data that exist, we can see that the value of the correlation coefficient reaches 0.71, indicating that the two sets of data have a relatively high correlation.



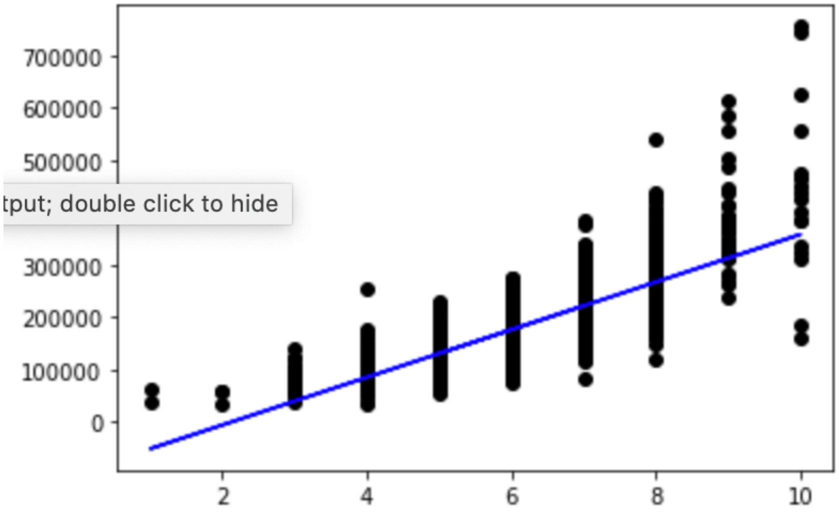


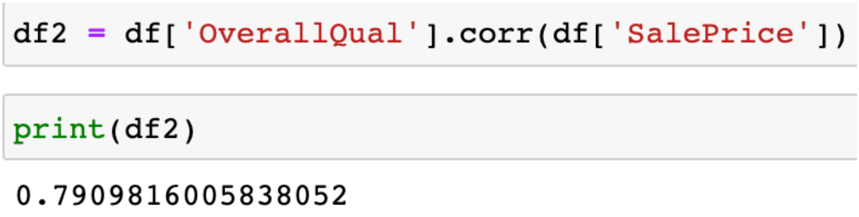
```
df1 = df['GrLivArea'].corr(df['SalePrice'])
```

```
print(df1)
```

```
0.7086244776126522
```

The following linear regression plot reflects the relationship between the total quality of the house and the sales price variable. The higher the numbers, the better, and the sales price will go up. It shows that the quality of the house is positively correlated with the sales price, and the correlation is about 0.80.



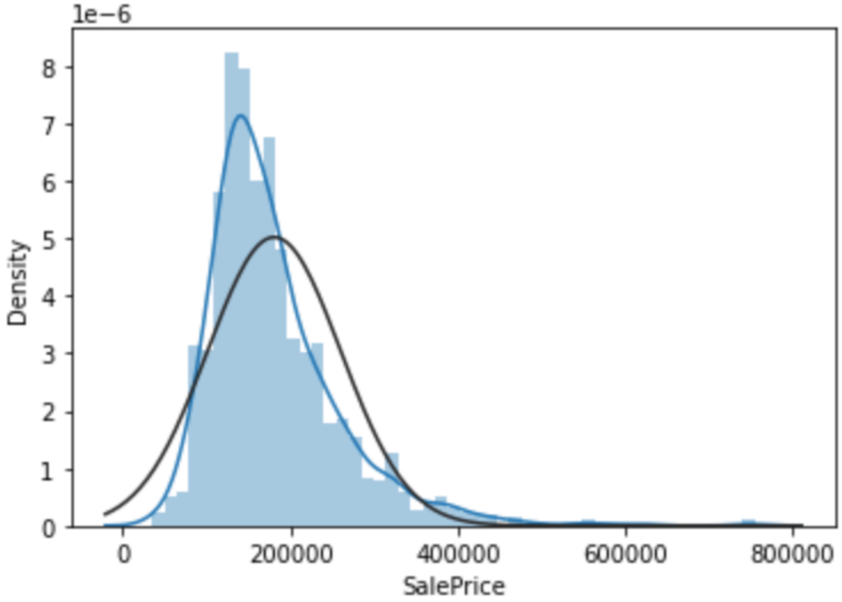


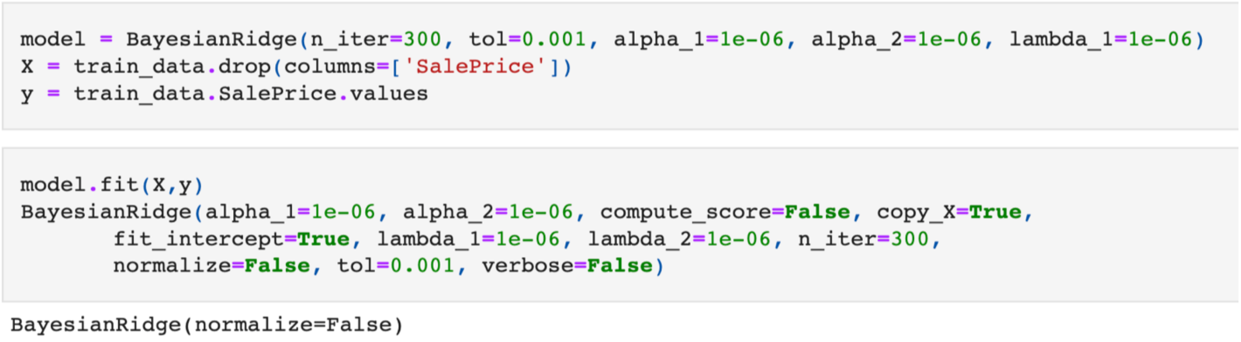
```
df2 = df['OverallQual'].corr(df['SalePrice'])
```

```
print(df2)
```

```
0.7909816005838052
```

We plot the distribution by using the histogram and the maximum likelihood gaussian distribution fit as follows. As we can see the density of home sales prices is mainly concentrated between $100,000 and $300,000. From the side, it reflects the rise of the low and middle classes in the United States. Due to low land prices, cheap housing costs, preferential tax policies, convenient transportation and other factors, the suburbanization of the United States has become an unrepeatable model. In addition, due to the impact of the epidemic, people have gradually chosen the suburbs for investment or settlement. As the process of suburbanization slows down, a new spatial balance is being reached between cities and suburbs.





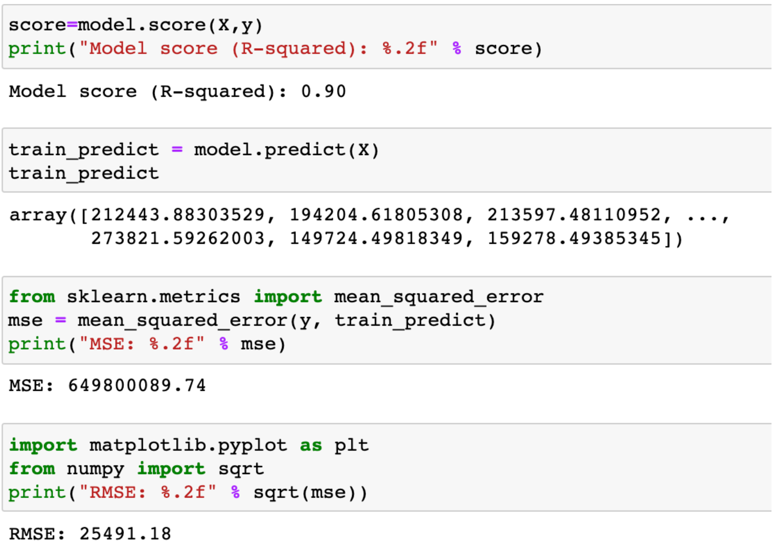
```
model = BayesianRidge(n_iter=300, tol=0.001, alpha_1=1e-06, alpha_2=1e-06, lambda_1=1e-06)
X = train_data.drop(columns=['SalePrice'])
y = train_data.SalePrice.values
```

```
model.fit(X,y)
BayesianRidge(alpha_1=1e-06, alpha_2=1e-06, compute_score=False, copy_X=True,
       fit_intercept=True, lambda_1=1e-06, lambda_2=1e-06, n_iter=300,
       normalize=False, tol=0.001, verbose=False)
```

```
BayesianRidge(normalize=False)
```

The above image fits the Bayesian Ridge model and optimizes the regularization parameters lambda (weight accuracy) and alpha (noise accuracy). Before that, we loaded the dataset and split it into training and testing sections; We then define the model with default parameters and fit it to the training data.

We also can check the model score that is R-squared metrics. We can draw from the following results. R-squared 0.90 means 90% of the variation in the output variable is explained by the input variables. Finally, we predicted the test data and checked the accuracy level and created graphs to visualize the results and raw data.



```
score=model.score(X,y)
print("Model score (R-squared): %.2f" % score)
```

```
Model score (R-squared): 0.90
```

```
train_predict = model.predict(X)
train_predict
```

```
array([212443.88303529, 194204.61805308, 213597.48110952, ...,
       273821.59262003, 149724.49818349, 159278.49385345])
```

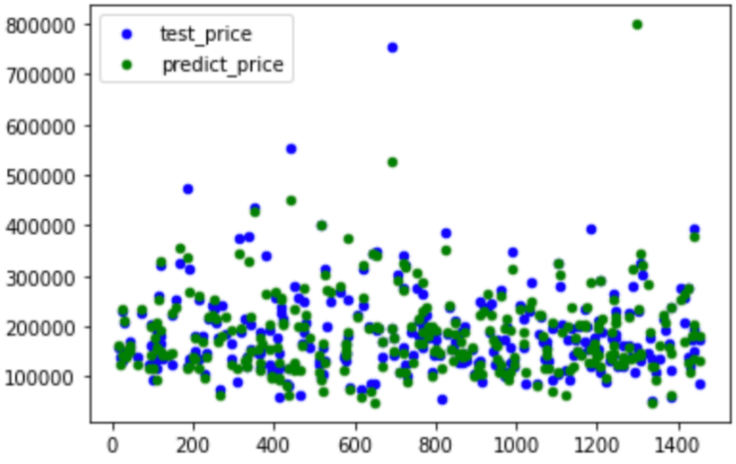
```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y, train_predict)
print("MSE: %.2f" % mse)
```

```
MSE: 649800089.74
```

```
import matplotlib.pyplot as plt
from numpy import sqrt
print("RMSE: %.2f" % sqrt(mse))
```

```
RMSE: 25491.18
```

We also took advantage of the model and got the difference between the price and the original price. The image shows that the test price and the predicted price roughly coincide, indicating the accuracy of our model forecast.



Conclusion

Based on our model analysis above, the larger the R-squared value, the more accurately the predictors are able to predict the value of the response variable, and other models fit closely. Through this semester of programming and python learning, overall, we did a good job.