Regression Project: Boston House Price Prediction

Marks: 60

↳ 20 cells hidden

Bivariate Analysis

[ ] ↳ 5 cells hidden

Model Building - Linear Regression

```
#Linear regression model libraries
from statsmodels.formula.api import ols
import statsmodels.api as sm
from sklearn.preprocessing import MinMaxScaler
from statsmodels.stats.outliers_influence import variance_inflation_factor

# Library to split data
from sklearn.model_selection import train_test_split
```

Steps to follow

- Split dependent and independent variables
- Split data in test and train sets
- Build OLS model
  - Standardise numerical features
  - Test assumptions of model & perform feature selection if needed
- Test model

```
# Split dependent and independent variables
# Independent variables
X = df.drop(['MEDV'],axis=1)
# Dependent variable
y = medlog
```

```
# Split data into test and train data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state = 1)
```

```
# Scale variables
X_train_scaled = pd.DataFrame(MinMaxScaler().fit_transform(X_train),index=X_train.index,columns=X_train.columns)
```

```
# Build OLS model
# Add intercept term in regression
X_train_scaled=sm.add_constant(X_train_scaled)
# Create model
model1=sm.OLS(y_train, X_train_scaled).fit()
```

Model Performance Check

1. How does the model is performing? Check using Rsquared, RSME, MAE, MAPE
2. Is there multicollinearity? Check using VIF
3. How does the model is performing after cross validation?

```
print(model1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   MEDV   R-squared:                       0.774
Model:                            OLS   Adj. R-squared:                  0.766
Method:                 Least Squares   F-statistic:                     97.19
Date:                Fri, 13 Oct 2023   Prob (F-statistic):          3.27e-102
```

```
Time:                        18:32:36   Log-Likelihood:                 80.575
No. Observations:                 354   AIC:                            -135.1
Df Residuals:                     341   BIC:                            -84.85
Df Model:                          12
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          3.7435      0.099     37.969      0.000       3.550       3.937
CRIM          -0.9476      0.125     -7.579      0.000      -1.194      -0.702
ZN             0.1638      0.074      2.226      0.027       0.019       0.308
INDUS          0.0876      0.083      1.053      0.293      -0.076       0.251
CHAS           0.1028      0.039      2.644      0.009       0.026       0.179
NOX           -0.4929      0.090     -5.462      0.000      -0.670      -0.315
RM             0.2654      0.107      2.471      0.014       0.054       0.477
AGE            0.0320      0.062      0.519      0.604      -0.089       0.153
DIS           -0.5753      0.112     -5.126      0.000      -0.796      -0.355
RAD            0.3517      0.079      4.458      0.000       0.197       0.507
TAX           -0.2860      0.103     -2.784      0.006      -0.488      -0.084
PTRATIO       -0.4018      0.064     -6.299      0.000      -0.527      -0.276
LSTAT         -1.0842      0.088    -12.316      0.000      -1.257      -0.911
==============================================================================
Omnibus:                       34.291   Durbin-Watson:                   1.980
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               88.992
Skew:                           0.449   Prob(JB):                     4.74e-20
Kurtosis:                       5.286   Cond. No.                         26.1
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Observations

- We see an R^2 value of 0.774 which is pretty decent
- The 'INDUS' and 'AGE' features have p-values above 0.05, meaning we have reasonable evidence to fail to reject the null hypothesis. Before we move on to checking the 4 linear regression assumptions, let's check the model for multicollinearity by evaluating the features' VIF scores. We will perform iterative feature selection through this.

```
# Create function that returns a dataframe with VIF scores
def vif(X_train_scaled):
  vif_scores = pd.Series([variance_inflation_factor(X_train_scaled.values, i) for i in range(len(X_train_scaled.columns))], index=X_trai
  print('VIF scores')
  print(vif_scores)
```

```
vif(X_train_scaled)
```

```
VIF scores
const      89.257836
CRIM        1.924114
ZN          2.743574
INDUS       3.999538
CHAS        1.076564
NOX         4.396157
RM          1.860950
AGE         3.150170
DIS         4.355469
RAD         8.345247
TAX        10.191941
PTRATIO     1.943409
LSTAT       2.861881
dtype: float64
```

Observations

- Two fetaures have high VIF scores. These are 'RAD' and 'TAX'
- Since TAX's VIF score is higher, let's remove it from our training data and check the data's collinearity again

```
# Collinearity v2
# Remove 'TAX' feature
X_train_scaled2=X_train_scaled.drop('TAX',axis=1)
# Check collinearity
vif(X_train_scaled2)
```

```
VIF scores
const      89.256101
CRIM        1.923159
ZN          2.483399
INDUS       3.270983
CHAS        1.050708
NOX         4.361847
RM          1.857918
AGE         3.149005
```

```
DIS        4.333734
RAD        2.942862
PTRATIO    1.909750
LSTAT      2.860251
dtype: float64
```

Observations

- The high collinearity of RAD is gone! We can now update our model on this new training data

```
# Linear regression model v2
model2=sm.OLS(y_train, X_train_scaled2).fit()
print(model2.summary())
```

```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:                   MEDV   R-squared:                       0.769
Model:                            OLS   Adj. R-squared:                  0.761
Method:                 Least Squares   F-statistic:                     103.3
Date:                Fri, 13 Oct 2023   Prob (F-statistic):          1.40e-101
Time:                        18:32:54   Log-Likelihood:                 76.596
No. Observations:                 354   AIC:                            -129.2
Df Residuals:                     342   BIC:                            -82.76
Df Model:                          11                                         
Covariance Type:            nonrobust                                         
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          3.7447      0.100     37.612      0.000       3.549       3.941
CRIM          -0.9399      0.126     -7.445      0.000      -1.188      -0.692
ZN             0.1007      0.071      1.425      0.155      -0.038       0.240
INDUS         -0.0112      0.076     -0.148      0.883      -0.161       0.138
CHAS           0.1196      0.039      3.082      0.002       0.043       0.196
NOX           -0.5151      0.091     -5.675      0.000      -0.694      -0.337
RM             0.2775      0.108      2.560      0.011       0.064       0.491
AGE            0.0287      0.062      0.461      0.645      -0.094       0.151
DIS           -0.5532      0.113     -4.894      0.000      -0.776      -0.331
RAD            0.1750      0.047      3.699      0.000       0.082       0.268
PTRATIO       -0.4252      0.064     -6.659      0.000      -0.551      -0.300
LSTAT         -1.0783      0.089    -12.134      0.000      -1.253      -0.904
==============================================================================
Omnibus:                       30.699   Durbin-Watson:                   1.923
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               83.718
Skew:                           0.372   Prob(JB):                     6.62e-19
Kurtosis:                       5.263   Cond. No.                         24.9
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Observations

- VIF scores for the features are all now acceptable
- ZN, INDUS, and AGE all have p-values above 0.05. This means that for these features we fail to reject the null hypothesis, and as such the features are deemed insignificant in our model. Therefore, we can go ahead and remove them, then update our model

```
# Linear regression model v3
X_train_scaled3=X_train_scaled2.drop(['ZN','INDUS','AGE'],axis=1)
model3=sm.OLS(y_train, X_train_scaled3).fit()
print(model3.summary())
print('-'*100)
print(vif(X_train_scaled3))
```

```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:                   MEDV   R-squared:                       0.767
Model:                            OLS   Adj. R-squared:                  0.762
Method:                 Least Squares   F-statistic:                     142.1
Date:                Fri, 13 Oct 2023   Prob (F-statistic):          2.61e-104
Time:                        18:33:20   Log-Likelihood:                 75.486
No. Observations:                 354   AIC:                            -133.0
Df Residuals:                     345   BIC:                            -98.15
Df Model:                           8                                         
Covariance Type:            nonrobust                                         
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          3.7487      0.096     39.211      0.000       3.561       3.937
CRIM          -0.9191      0.125     -7.349      0.000      -1.165      -0.673
CHAS           0.1198      0.039      3.093      0.002       0.044       0.196
NOX           -0.5133      0.082     -6.296      0.000      -0.674      -0.353
RM             0.3074      0.105      2.928      0.004       0.101       0.514
DIS           -0.4846      0.087     -5.561      0.000      -0.656      -0.313
RAD            0.1805      0.046      3.890      0.000       0.089       0.272
```

```
PTRATIO       -0.4559      0.058     -7.832      0.000      -0.570      -0.341
LSTAT         -1.0610      0.082    -12.949      0.000      -1.222      -0.900
==============================================================================
Omnibus:                       32.514   Durbin-Watson:                   1.925
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               87.354
Skew:                           0.408   Prob(JB):                     1.07e-19
Kurtosis:                       5.293   Cond. No.                         21.0
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
----------------------------------------------------------------------------------------------------
VIF scores
const      82.501860
CRIM        1.892679
CHAS        1.049602
NOX         3.528194
RM          1.748438
DIS         2.582254
RAD         2.838523
PTRATIO     1.591527
LSTAT       2.437311
dtype: float64
None
```

Observations

- We now see that every feature is significant to our model (p-value<=0.05) and all VIF scores are sufficiently low (<5.0)
- model 3 is therefore our final model. We can ahead and make that clear in our saved variables

```
# model3 is our final model
ols_model=model3
```

Checking Linear Regression Assumptions

- In order to make statistical inferences from a linear regression model, it is important to ensure that the assumptions of linear regression are satisfied. These assumptions are:
  - **Constant zero mean of residuals**
  - **No heteroscedasticity**
  - **Linearity of variables**
  - **Residuals are normally distributed**

```
# Constant zero mean of residuals
res=ols_model.resid
print(res.mean())
```

```
4.193130463061679e-15
```

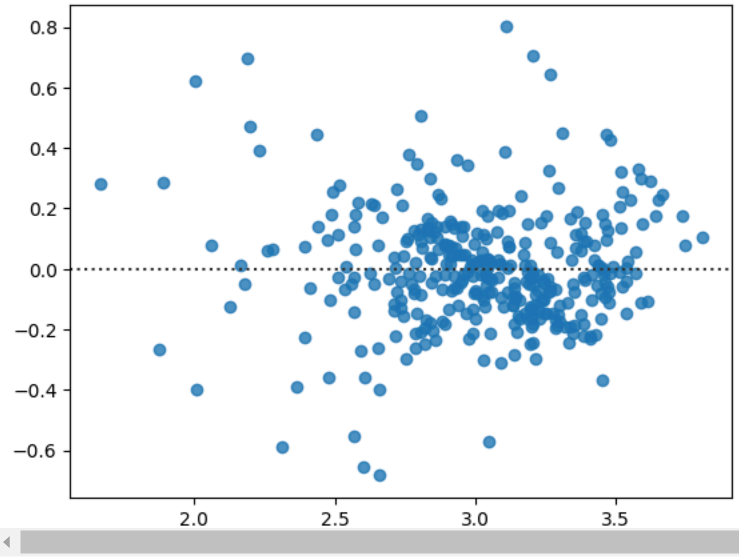
```
# No heteroscedasticity
# Proceed with Goldfeld-Quandt hypothesis test
import statsmodels.stats.api as sms
from statsmodels.compat import lzip

s = ["F stat", "p-value"]
t = sms.het_goldfeldquandt(y_train, X_train_scaled3)
lzip(s, t)
```

```
[('F stat', 1.0835082923425288), ('p-value', 0.30190120067668275)]
```

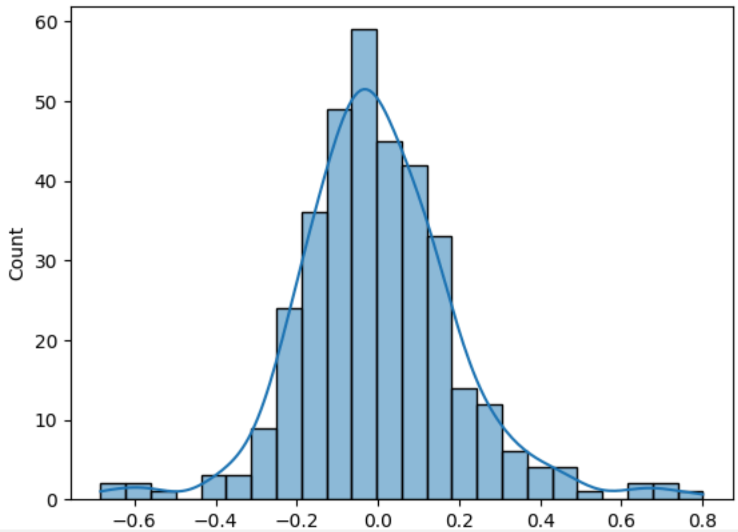
```
# Linearity of Variables
f=ols_model.fittedvalues
sns.residplot(x=f,y=res)
```

```
<Axes: >
```



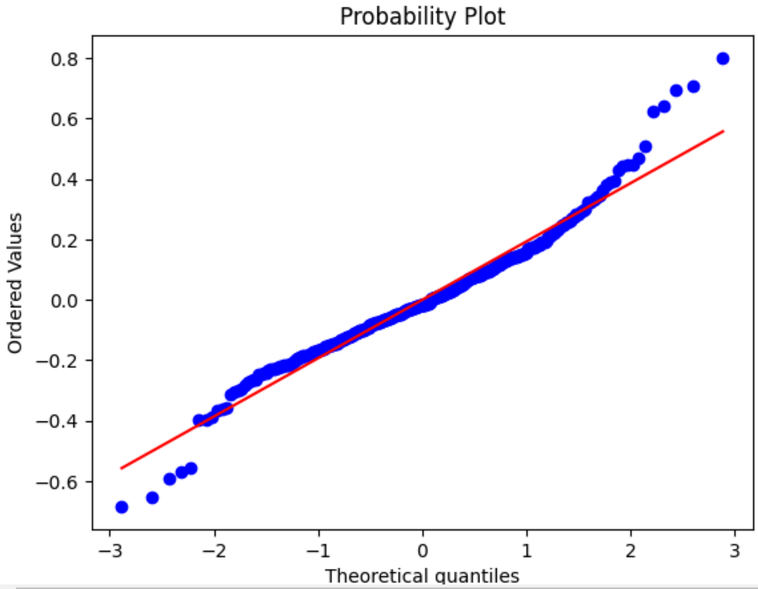

```
# Normality of error terms
sns.histplot(res,kde=True)
print('The skew of the residuals is',res.skew(),'with median',res.median(),'and mean',res.mean())
plt.show()
```

```
The skew of the residuals is 0.40987469081129696 with median -0.018686816844734944 and mean 4.193130463061679e-15
```




```
import pylab
import scipy.stats as stats

stats.probplot(res, plot = pylab)
plt.show()
```




```
print(ols_model.rsquared)
print(ols_model.mse_resid)
print(np.sqrt(ols_model.mse_resid))
```

```
0.7671737057912822
0.0392187883349037
0.19803734075901872
```

Performance metrics

Metrics used:

- **MSE**
- **RMSE**
- **MAE**
- **MAPE**

```
# Prediction on training data
y_pred_train = ols_model.predict(X_train_scaled3)

# Prediction on test data
X_test_scaled=sm.add_constant(pd.DataFrame(MinMaxScaler().fit_transform(X_test),index=X_test.index,columns=X_test.columns)).drop(['TAX',
y_pred_test = ols_model.predict(X_test_scaled)
```

```
# MSE
mse_train=((y_pred_train-y_train)**2).mean()
mse_test=((y_pred_test-y_test)**2).mean()
# RMSE
rmse_train=np.sqrt(mse_train)
rmse_test=np.sqrt(mse_test)
# MAE
mae_train=(np.abs(y_pred_train-y_train)).mean()
mae_test=(np.abs(y_pred_test-y_test)).mean()
# MAPE
mape_train=((np.abs(y_pred_train-y_train))*100/y_train).mean()
mape_test=((np.abs(y_pred_test-y_test))*100/y_test).mean()
```

```
met=[mse_train,mse_test,rmse_train,rmse_test,mae_train,mae_test,mape_train,mape_test]
metname=['mse_train','mse_test','rmse_train','rmse_test','mae_train','mae_test','mape_train','mape_test']
```

```
for i,j in zip(met,metname):
  print(j,': ',i,sep='')
```

```
mse_train: 0.03822170049588073
mse_test: 0.04033136829216085
rmse_train: 0.19550370967293876
rmse_test: 0.200826712098169
mae_train: 0.14368596074360607
mae_test: 0.15573407255418578
mape_train: 4.981812504754256
mape_test: 5.237965780561446
```

Observations

- V GUCCI WAHOO
- mape_test is above 5% => issue?

Cross-Validation

```
### Cross validate with Linear Regression model to get C-V scores
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import cross_val_score

lreg=LinearRegression()
cv1 = cross_val_score(lreg, X_train, y_train, cv = 10)
cv2 = cross_val_score(lreg, X_train, y_train, cv = 10, scoring = 'neg_mean_squared_error')

print('R^2 value:',round(cv1.mean(),3),'+/-',round(cv1.std()*2,3))
print('MSE:',round(-cv2.mean(),3),'+/-',round(cv2.std()*2,3))
```

```
R^2 value: 0.733 +/- 0.232
MSE: 0.041 +/- 0.023
```

Observations

- R^2 value is very similar to the R^2 value of our OLS model of 0.767, although the standard deviation in the cross-validation is very large
- MSE value is likewise similar to the MSE value of the model
- This points to our model being well fit to our training data. We will check below how it fairs on test data

Final Model

```
print(ols_model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   MEDV   R-squared:                       0.767
Model:                            OLS   Adj. R-squared:                  0.762
Method:                 Least Squares   F-statistic:                     142.1
Date:                Fri, 13 Oct 2023   Prob (F-statistic):          2.61e-104
Time:                        20:02:15   Log-Likelihood:                 75.486
No. Observations:                 354   AIC:                            -133.0
Df Residuals:                     345   BIC:                            -98.15
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          3.7487      0.096     39.211      0.000       3.561       3.937
CRIM          -0.9191      0.125     -7.349      0.000      -1.165      -0.673
CHAS           0.1198      0.039      3.093      0.002       0.044       0.196
NOX           -0.5133      0.082     -6.296      0.000      -0.674      -0.353
RM             0.3074      0.105      2.928      0.004       0.101       0.514
DIS           -0.4846      0.087     -5.561      0.000      -0.656      -0.313
RAD            0.1805      0.046      3.890      0.000       0.089       0.272
PTRATIO       -0.4559      0.058     -7.832      0.000      -0.570      -0.341
LSTAT         -1.0610      0.082    -12.949      0.000      -1.222      -0.900
==============================================================================
Omnibus:                       32.514   Durbin-Watson:                   1.925
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               87.354
Skew:                           0.408   Prob(JB):                     1.07e-19
Kurtosis:                       5.293   Cond. No.                         21.0
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

```
pd.DataFrame({'Independent Feature':ols_model.params.index,'Coefficient':ols_model.params.values})
```

	Independent Feature	Coefficient
0	const	3.748700
1	CRIM	-0.919131

Equation of Model

log(**MEDV**) = 3.7487 + (-0.9191)**CRIM** + (0.1198)**CHAS** + (-0.5133)**NOX** + (0.3074)**RM** + (-0.4846)**DIS** + (0.1805)**RAD** + (-0.4559)**PTRATIO** + (-1.061)**LSTAT**

Actionable Insights and Recommendations

Observations and insights

- The most important variables that influence the median value of owner-occupied homes are the percentage of the suburb's/town's population that is of 'lower status' and the town's per-capita crime rate.
    - Both these features have incredibly strong negative correlations with median house prices
    - This is what we would expect. Houses that are found in areas with high crime rates will inevitably have lower demand and this will mark down prices. Similarly, areas where a significant proportion of the population are below average in income would see lower house prices to accommodate residents' means
- The least influential variables that still have a statistically significant effect on the median value of owner-occupied homes are the Charles River dummy variable and the index of accessibility to radial highways
    - If we assume a rudimentary model of price vs demand, we can infer here that an area's proximity to the river or its accessibility to radial highways have little impact on Bostonian residents' interest in living there compared to the average for the Boston housing market
- It is important to note that 3 features were removed from the final model for failing to have a significant effect on the dependent variable. These are:
    - The proportion of residential land zoned for lots over 25,000 sq.ft.
    - The proportion of non-retail business acres per town
    - The proportion of owner-occupied units built before 1940
- 1 feature was removed for having too strong collinearity with other features, This is:
    - The full-value property-tax rate per 10,000 dollars
        - This feature was most strongly correlated with the weighted distances to five Boston employment centres and the proportion of non-retail business acres per town

Business Recommendations

We are fortunate that two features have an incredibly strong negative correlation with median house values. This means that the company can