Ames Housing OLS Regression Project (300 Points)

DELIVERABLES

- Your write up in PDF Format (no zip files). Your write up should have five sections. Each section should have enough detail so that I can follow your logic and someone else can replicate your work.

- A file that contains all the python code you used in your analysis. I should be able to run this file and get all the output that you got.

- A csv file,which has the scored records values from ames_test_sfam. There will be only two columns in this file: index and p_saleprice. You will be graded on how your model performs versus my model and those of other students in the class. You can see how well your model performance improves bycontinuing to submityour csv file tokaggle.

Section 1.Modeling & More (100 points)

Submit at least 4 models for this assignment. It is a continuation of the model building process for the Ames Housing Data. Your models should predict SalePrice with increasingly complex models. However, keep in mind the principle of parsimony. Find the best model with the least complexity and a model that you can explain.

Show all of your models and the statistical significance of the variables in the model. This should include coefficient estimates, t-values, p-values, etc.

Do not skip any of the sections in this assignment. I want you to try everything. Not everything will improve your model but you need to try it all.

Answer all of the questions.

Neighborhood Accuracy

Use one of your models from HW01. Make a boxplot of the residuals by neighborhood. Which neighborhoods are better fit by the model? Do you have neighborhoods that are consistently over-predicted? Do you have neighborhoods that are consistently under-predicted?

Compute actual and estimated mean price per square foot for each neighborhood.

Group the neighborhoods by actual price per square foot. Create between 3 and 6 groups. Code a family of indicator variables for the neighborhood groupsto include in your multiple regression model. See Chapter 5 p131. Your indicator variables should be of the form Group1 = 1 if ppsf is in some range, Group1 = 0 otherwise. The dummy or indicator variable notation, has a long history in mathematics. It is also referred to as the Kronecker delta. As an example, think about a simple way for an indicator using a single variable to represent gender. Gender = 1 if male, Gender = 0 otherwise. Expand this to the neighborhoods where you have more than two possible outcomes. Decide which neighborhood group should be the reference group. Then for each of the other groups set up the 1 or 0 naming. 1 means you are in that group, 0 means you are not in that group. If each of the groups you define has the value 0 then that will define the reference group. There is much written on this topic and it is based on knowing how to set up the dummy, 1, 0 notation. Also, the matrix design for solving this type of problem uses the 1, 0 notation. Common terminology in statistics for random variables is let X = 1 if something happens, X = 0 if it doesn't happen. So, the notation P(X = 1) has a clear meaning. To understand how your model works you have to understand this notation.

What is your base category? Refit your multiple regression model with your indicator variables.

Two new variables are defined in the python shell code.

```
df['qualityindex'] = (df.overallqual*df.overallcond)
df['totalsqftcalc'] = (df.bsmtfinsf1+df.bsmtfinsf2+df.grlivarea)
```

Include these in your models. Can you think of other variables that make sense to define? Try something creative. It might work.

Section 2.Model Comparison of Y versus log(Y) (20 points)

In this section, fit two models using the same set of predictor variables, but the response variables will be SalePrice and log(SalePrice). You may use any set of predictor variables that you wish, but the models must include at least four continuous predictor variables and any discrete variables that you wish.

Respond to all of these bullet questions:

- How do we interpret these two models?
- How is the interpretation of the log(SalePrice) model different from the price model?
- Which model fits better?
- Did the transformation of the response to log(SalePrice) improve the model fit?
- In general when can a log transformation of the response variable improve the model fit?
- Should we consider any transformations to the predictors? If so, then fit one more model using any transformations that you find appropriate.

Compute the VIF values for the models. If the models have highly correlated pairs of predictors that you do not like, then go back, add them to your drop list, and re-perform the variable selection before you go on with the assignment. The VIF values do not need to be ideal, but if you have a very large VIF value (like 20, 30, 50 etc.), then you should consider removing a variable so that your variable selection models are not junk too.

Produce the relevant diagnostic plots to assess the goodness-of-fit of each model. On what criteria are you assessing the model fit? Always report the fitted model when we fit a linear model. This means that your report should contain a table with the coefficient estimates, t-values, p-values, etc.

Optional

If you would like to try an automated selection algorithm, try feature selection and f_regression for numeric variables:
fromsklearn import feature_selection
fromsklearn.linear_model import LinearRegression

Sklearn DOES have a forward selection algorithm, although it isn't called that in scikit-learn. The feature selection method called F_regression in scikit-learn will sequentially include features that improve the model the most, until there are K features in the model (K is an input).

It starts by regressing the labels on each feature individually, then observing which feature improved the model the most using the F-statistic. Then it incorporates the winning feature into the model. Then it iterates through the remaining features to find the next feature which improves the model the most, again using the F-statistic or F test. It does this until there are K features in the model.

Some code for this:

```
y = train['saleprice']
X = train[['intercept',
'qualityindex','totalsqftcalc','yearbuilt','yearremodel','wooddecksf','openporchsf'
]].copy()
X.head()
model = feature_selection.SelectKBest(score_func=feature_selection.f_regression, k=3)
#you will need to convert y to an array
results = model.fit(X,y)
#scores are the F values for each variable, bigger is better
results.scores_
```

Try it. See if f_regression improves your model.

Use Kaggle to compare your models and other metrics.

It is OK to submit to Kaggle many times.

Section 3.SELECT MODELS (20 Points)

Decide on the criteria for selecting the "Best Model". Will you use a metric such as Adjusted R-Square or AIC? Will you select a model with slightly worse performance if it makes more sense or is more parsimonious? Discuss why you selected your model. Put the metrics in a table to display the results.

If you are just using Kaggle, that is OK. Just tell me what you are doing.

Section 4.Model Formula (10 Points)

Write a formula that will predict the sale price of a home. **Make sure you include this as a section in your report. Do not expect that I will search your report to find it.** This step should allow someone else to deploy your model.

The variable with the predicted saleprice should be named:

p_saleprice

Section 5.SCORED DATA FILE (150 POINTS)

Pick your best model or send me more than one model. I will score several models with my code for you. Use your best model. Score the data file ames_test_sfam. Create a file that has only TWO variables for each record:

index
p_saleprice

The first variable, index, will allow me to match my grading key to your predicted value.

Name your file yourname_410_hw02.csv.

If I cannot score your model, you won't get a grade. So please include the index number. The second value, p_saleprice is the predicted price for a property per your model.

Your values will be compared against …

- A Perfect Model
- Shell Code Example Model
- Performance of Other Students
- Predict the Average value for every home (MEAN)

If your model is not better than simply using an AVERAGE value, you will lose points.

BONUS

Optional (10 pt Bonus): Assess the predictive accuracy of your model using cross-validation. Use python code to split the full dataset for the Ames Housing Datafor your own train and test datasets. See HW02 shell code.

A defining feature of predictive modeling is assessing model performance out-of-sample. You will use uniform random numbers to split the training data into a 70/30 train/test split. With a train/test split you have two data sets: one for in-sample model development and one for out-of-sample model assessment.

If you want Bonus Points, write a brief section at the top of your Write Up document and tell me exactly what you did and how many points you are attempting. If I cannot see your Bonus work, I cannot give you credit. Bonus is difficult to grade and I don't have time to go back looking for it. If you don't tell me it's there, I cannot give you points.

The policy with Bonus is: ***All Sales are Final !***

- (10 Points) Once you select a modeltry something else. Are the results the same? Are there any differences?
- (?? Points) Roll the dice … think of something creative and run with it. I might give you points.

PENALTY BOX

- (Lose 10 Points) If you don't have PDF format
- (Lose 10 Points) If you don't have a GOOD Introduction
- (Lose 10 Points) If you don't have a GOOD Conclusion
- (Lose 10 Points) If you don't put your NAME in the file names of any files you hand in