MAT 128 Lab 16: Rsquared and Multiple Linear Regression

Motivating question: Can we predict when a building was built near Times Square?

Dataset: Times Square Property Data (Commercial and Retail Properties) from OpenDataNYC

This dataset stores information about various buildings near Times Square.

1) Download the full dataset.
2) What kind of information is stored in this dataset?

Cleaning the data:

1) We are only going to use the following columns: 'Year Built','Number Of Stories','Land Area (AC)', 'Rentable Building Area','Typical Floor Size','Number Of Elevators', 'Percent Leased'

To load only these columns into the dataframe: put their names in a list, and add the option/parameter `usecols=name_of_list` in `read_csv`. Try it before looking at the code below.

One way to do this:

```
fields = ['Year Built','Number Of Stories','Land Area (AC)',
'Rentable Building Area','Typical Floor Size','Number Of
Elevators', 'Percent Leased']

ts =
pd.read_csv('Times_Square_Property_Data__Commercial_and_Retail_pro
perties_.csv',usecols =fields)
```

2) Since the column names have spaces and we will be building a linear model (where the command will not take column names with spaces), we need to rename the columns:

```
ts.columns =
['Year','Stories','Land_area','Rentable_area','Floor_size','Elevat
ors','Percent_leased']
```

3) Run your code, and run the `head()` command to check that the dataframe has been loaded property. What do you notice about rows 1, 3, and 4?

4) `NaN` is used to represent missing data. We want to remove the rows with missing data, since they will cause problems later. Assuming your dataframe is called `ts` add the following code:

```
ts = ts.dropna(axis=0, how = 'any')
```

`axis=0` drops the entire row containing the missing data, whereas `axis=1` drops the entire column

`how='any'` drops the row/column if any of the values are missing, whereas `how='all'` only drops the row/column if all of the values are missing

Exploratory Data Analysis:

1) Compute the correlation matrix for the dataframe (ie. for all columns), and display it as a heatmap. Which pairs of columns have the greatest correlation? Are there any pairs of columns that have no, or almost no, relation?

If you want to change the color scheme of the heatmap, add the option/parameter: `cmap='name_of_color_map'`

For example: `cmap = 'YlGnBu'`

`cmap` stand for 'color map'. Some possible color maps are listed here: http://www.r-graph-gallery.com/38-rcolorbrewers-palettes/

2) If you have time: plot all pairs of columns as scatter graphs. Hint: remember the pairplot command in Seaborn.

Do the graphs match what you noticed from the correlation matrix?

Model Building and Prediction:

1) Can we predict the number of stories a building has from the number of elevators? Try building a linear model to predict this. Hint: see code from Lab 15

2) Print out the coefficients/parameters of the model. For each new elevator, how many more stories does the model predict?

3) Use Seaborn to make the scatter plot of the Stories and Elevators columns data with the best fit line.

4) Predict how many stories a building with 20 elevators will have. Hint: see code from Lab 15

Validating the Model:

Validating the model means checking how accurate or good the model is at predicting the real data.

We will do this by computing R-squared:

$$R_{Sq} = 1 - \frac{RSS}{TSS}$$

where RSS = Residual Sum of Squares

$$RSS = \sum_{i=1}^{n} (y_i - y_{fitted})^2$$

and TSS = Total Sum of Squares

$$TSS = \sum_{i=1}^{n} (y_i - y_{mean})^2$$

This can be done in Python one line using `statsmodel`:
```
lm.rsquared
```

To print it:
```
print(lm.rsquared)
```

What's the R-Squared value for our model?

Multiple Linear Regression

We can use the date from more than one column to predict the number of stories. For example, to use the Elevators and Land_area columns:

```
lm2 = smf.ols(formula = 'Stories ~ Elevators + Land_area ', data = ts).fit()
```

We can still compute the parameters/coefficients the same way:
```
print(lm2.params)
```

And R-Squared:
```
print("R-Squared:",lm2.rsquared)
```

To predict the number of stories, we need to make a new dataframe with both the number of elevators and the land area:

```
new_data = pd.DataFrame({'Elevators':[10,20],'Land_area':[0.11,0.14]})
```

And then use it to make the prediction:
```
print("Prediction:",lm2.predict(new_data))
```

How many stories will a 10 elevator building on 0.11 acres of land be?

How many stories will a 20 elevator building on 0.14 acres of land be?

Classwork

What linear model can you build from these columns and data that has the best R-Squared value?

Does adding new variables to the model always increase the R-Squared value?