1. LINEAR Regression of NYC_Real_Estate data

OVERVIEW: Target variable is 'SALE PRICE'

We will be using the <u>Multiple Regression</u> as there are more than one variable that will affect the SALE PRICE. Namely, Total Units and Gross Square Feet.

Data has 12 columns:

- BOROUGH-represents where the property is located
- BLOCK/LOT: represent a unique key where for where the property is located
- ZIP CODE
- RESIDENTIAL UNITS: the number of residential units at the property
- TOTAL UNITS: Total number of units at listed property
- GROSS SQUARE FEET: total area of floors of the building, incl land and interior space within building
- YEAR BUILD AT TIME OF SALE: when the property was built
- TAX CLASS AT TIME OF SALE:
- Building class at time of sale: classification used to describe the property's construction use at time of sale
- SALE PRICE: what the property was sold at.

Target variable is 'SALE PRICE'

Data Cleaning:

- Check for NUL values
- Combined columns BOROUGH/BLOCK/LOT/YEARBUILD/BUILDING CLASS AT TIME OF SALE into one column called 'BuildingInfo'
- Dropped the columns used to combine and ZIP CODE and TAX Class AT TIME OF SALE.
- Removed rows where columns = 0
- Filtered SALE PRICE to a threshold of values that are over 10000; because there were 854 rows with unrealistic values below the threshold. Unrealistic for NYC standards.

Conclusion:

- Multiple Regression did not result in a good model, as the R_sqr value is very very low at 0.151
- The Coefficients:
    - Total Units =196119.20
    - GROSS SQUARE FEET=-91.445578

2. Logistic Regression of Credit_Risk Data

OVERVIEW: Target variable is 'LOAN STATUS

Data has 12 columns and 32581 rows:

Columns: person_age, person_income, person_home_ownership, person_emp_length, loan_intent, loan_grade, loan_amount, loan_int_rate, loan_status, loan_percentage_income, cb_default_on_file and cb_person_cred_hist_length.

Target variable is loan_status.

Data Cleaning:

- Check for null values using df.isnull().sum()
- Person_emp_length has 895 Null values
- Loan_int_rate has 3116 Null values.
- Checked NULL values visually to see if we can fix them
- Null values on in columns. No need to delete rows
- Person_emp_length: replaced null values with its median because the values are very skewed.
- Loan_int_rate: replaced null values with its mean because the values are somewhat normally distributed.
- Checked to see the null values again= no more null values.

Information on data:

- 8 columns are either of type Int64 & float64: [person_age, person_income, person_emp_length, , , loan_amount, loan_int_rate, loan_status, loan_percentage_income and cb_person_cred_hist_length.
- 4 columns are object(categorical): [person_home_ownership, loan_intent, loan_grade & cb_default_on_file]

Outliers:

  - Person_age < 95: there were a few people over 95 years of age
  - Person_income<$500,000 too many outliers making over $500,000 (not sure why they would need a loan)
  - Person_emp_length<30: too many outliers above 30yrs.

Dummy Variable:

  - Converted all the feature variables into dummy variables for new dataframe called df1.

Split the Data for Logistic Regression:

  - X: Features include all dummy variables columns, except 'Loan_Status'
  - Y: Target Variable 'Loan_Status'

Conclusion:

Please see Logistic Regression Confusion Matrix.

The confusion Matrix is a table that is often used to describe the performance of classification model on a set of test data for which the true values are known.

The confusion matrix predicted how well our model predicted the Loan_Status.

At True Loan_Status of 0, our model predicted 97% of 0 and 3.2% of 1

At True Loan_Status of 1, our model predicted 79% of 1 and a 29% at 0

The model predicted, ok. We could have got a better result of we would have found more outliers, but also, we could have used the help of DecisionTree-Classifier to help us pick the best features, instead of picking all the features.

The Training Accuracy for the Train model was 91%

And the Training Accuracy for the Test model was 89%