|  |
| --- |
| Supervised Learning for Linear and Logistic Regression |
| LAB 3 |
| Khoua Chang  Dr. Jay Urbain  EE5980 Machine Learning  March 30, 2017 |

Objective

In this lab, I am using SciKit with Python libraries to develop machining learning applications in Supervised Learning for Linear and Logistic Regression. Supervised learning implies that all data is labeled and the algorithms learned to predict the output from the input data. This assignment will introduce and help me learn the wide range of supervised learning algorithms in Python. The idea is to approximate the accuracy of the mapping function so the new input data can predict the output variables for that data. The learning algorithm stop when the algorithm achieves a minimum performance.

Problem Statement

For the Linear Regression note book I will be working through a linear regression application using SciKit-learn. I will discuss about importing and visualizing the data, Ordinary Least Squares Regression, univariate linear regression, error in my model, multivariate regression, training and validation set, predicting prices, and generating residual plots. The Boston Housing Dataset will be used to implement this this method.

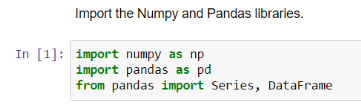
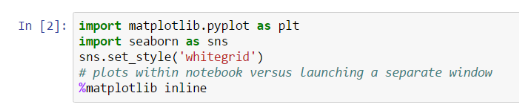
For Logistic Regression note book I will be working through a logistic regression classification application also using Scikit-learn. My dataset is from the Credit Card Default Data and I will use this dataset to predict which customers will default on their credit card. The classification method will give me an output base on how I categorize my variables.

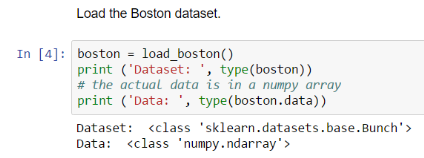
Procedure

**Linear Regression:**

1&2. Importing and visualize data

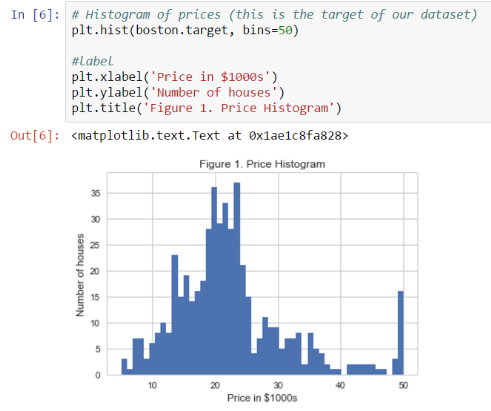
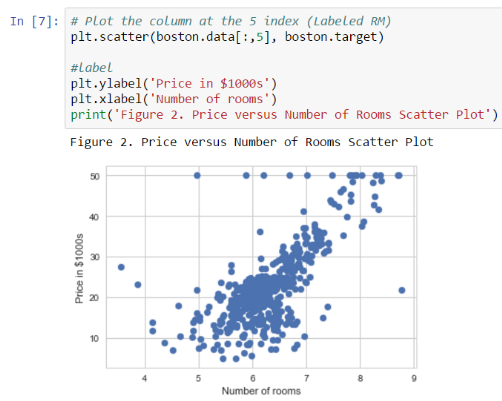
It is important to make sure that all libraries and packages are imported to be able to display and visualize the data. This lab also includes searbon statistical visualization library.

By using print(boston.DESCR) it gives me access to the meta-data via the DESCR field. Where I can see the characteristics and attributions.

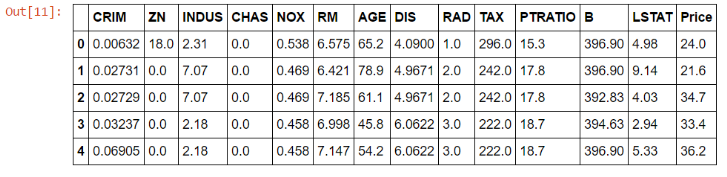
Using matplotlib.pyplot I can generate a histogram and call a scatter price of prices versus the number of rooms. In which I notice that the home prices increase with an increasing number of rooms.

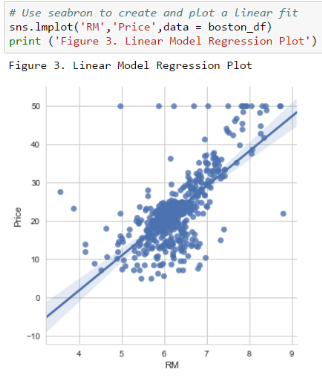
**Q1) There's an anomaly in the price target variable in the Boston dataset. What is it? See figures 1 and 2 above. Answer in the cell below.**

The anomaly in the price target variable are the 15 houses at $50,000 despite the number of rooms. Which have rooms that ranges from 5 to 8.75.

Using the Seaborn statistical visualization package to plot a linear regression model on the scatter plot to visualize how well a linear regression model fits the data.

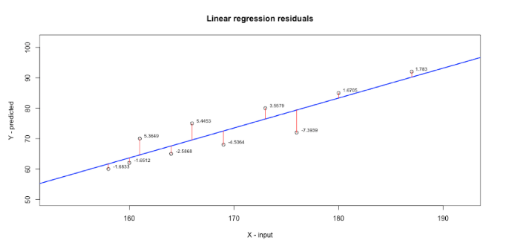


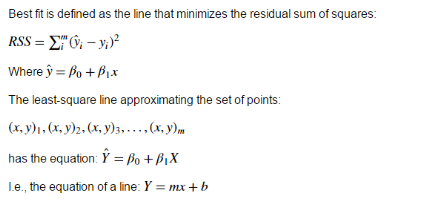
The Seaborn lmplot() function, fits a linear regression model to the data, plots the data as a scatter plot, and adds the regression line. Notice the transluscent bands corresponds to the confidence interval, where ci=95% .



3. Review Ordinary Least Squares (OLS) regression.

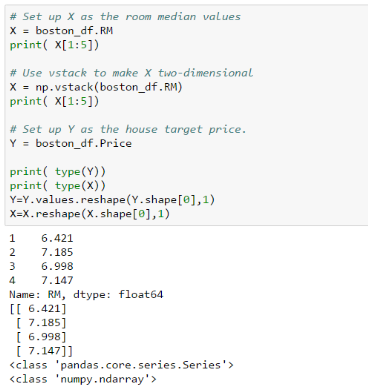
OLS regression fits a linear model (line in univariate regression, plane with two variates) by minimizing the residual sum of squares. Residuals are shown as red lines in the sample plot below. The residuals represent the difference between the predicted (point on line) and actual data (point).





4. Use Numpy for Univariate Linear Regression.

Numpy expects a 2D array. The first dimension contains the different values. The second dimension contains the attribute number. In this case, the value is the mean number of rooms per house. Since this is a single attribute, the second dimension of the array is 1. Therefore the shape array is (506,1).



We can plot the data using the original data format of the Boston housing information. We performed the matrix transformations to utilize the numpy least square method.



5. Determining the error of our model fit.

**Q2) Calculate the RSS (as defined in step 3) for the univariate linear regression model of the Boston dataset created in step 4**



The result array has the residual squared error (RSS). For each element, it checks the difference between the line (our prediction) and the true value, squares it, and returns the sum of all these. This is the RSS value. Note that the root mean squared error is similar to the standard deviation.

**Q3) Calculate the RMSE.**

**Q4. How much will the price of a house vary 95% of the time?**

Since the root mean square error (RMSE) corresponds to the standard deviation, we can say that the price of a house will not vary by more than 2 times the RMSE 95% of the time.

**Q5) Calculate the TSS for the univariate linear regression model of the Boston dataset.**

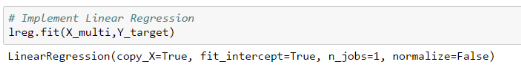
**Q6) Calcualte**R2R2**for the univariate linear regression model of the Boston dataset.**

6. Use *scikit-learn* to implement multivariate regression.

Note that Scikit-learn can be used for univariate or multivariate regression. The [sklearn.linear\_model.LinearRegression](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html) class is called an estimator.

Estimators predict a value based on the observed data. In scikit-learn, all estimators implement the fit() and predict() methods. The fit() method is used to learn the parameters of a model, and the predict() method is used to predict the value of a response variable for a given predictor variable using the learned coefficients.

In creating a linear regression object, I have to create linear functions in separating the Boston data frame into data columns and the target column. Now I can use this to fit the linear regression model to my X and Y values.



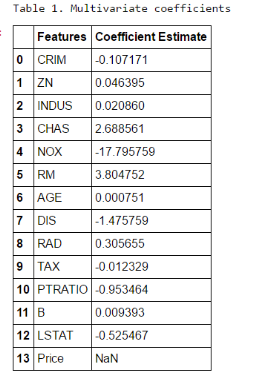
Which estimated,

Intercept coefficient=36.49

# of coefficients used=13



Using this model to create a multivariate dataframe to examine the model and the estimated coefficients.

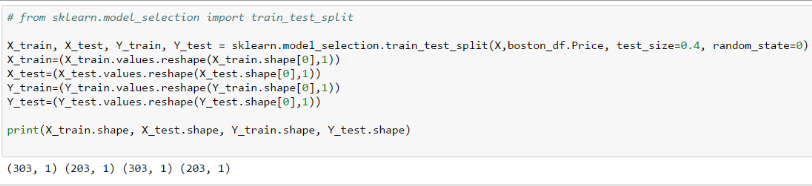


**Q7) Which coefficients, excluding nitric oxide (NOX) have the strongest correlation with the target variable**

DIS have the strongest correlation with the target variable

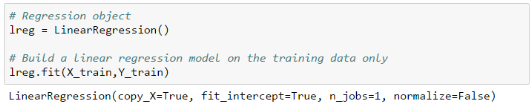
7. Use training and validation data sets.

Using the called train\_test\_split to Separate training and test sets to train and validate the model. Samples for each set should be randomly selected. I will create a separate training and tests sets, holding out 40% of the data for testing.

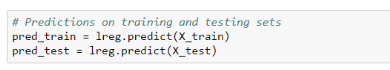


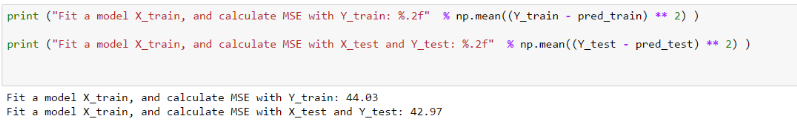
8. Predicting prices

I can use our training set to build the model, and the test set to evaluate the performance of our model.



Perform prediction on both the training set and the test set. And calculate the mean square error for each.

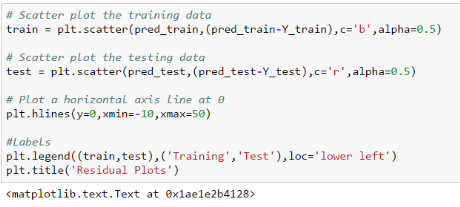




9. Generating residual plots

A residual plot is a graph that shows the residuals on the vertical axis and the independent variable (x) on the horizontal axis. If the points in a residual plot are randomly dispersed around the horizontal axis, a linear regression model is appropriate for the data; otherwise, a non-linear model is more appropriate.

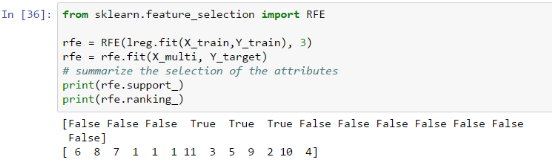
Residual plots are a good way to visualize the errors in data. A good model fit will show data points randomly and evenly scattered around line zero. If there is some structure or pattern, that means the model is not capturing some aspect of the data. There could be an interaction between predictor variables that we are not considering, or the data may be inherently non-linear.

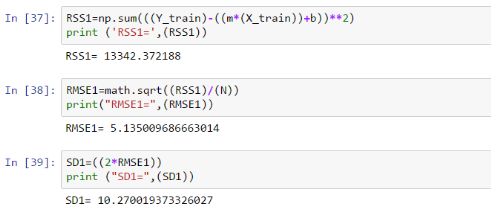


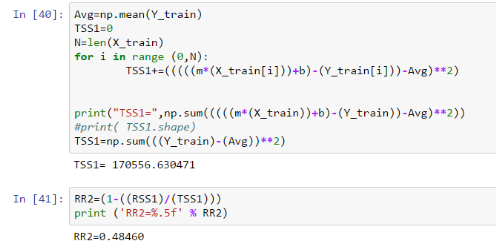


overall the majority of the residuals seem to be randomly distributed above and below the horizontal.

**Q8) Review Table 1. Multivariate coefficients. Think about the meaning of a linear regression model, i.e., the coefficient reflects the change in the target variable for a one unit change in an input variable, with all other variables held constant. Identify a subset of features and build a model with these features. See if you can reduce RMSE and increase**R2R2**.**







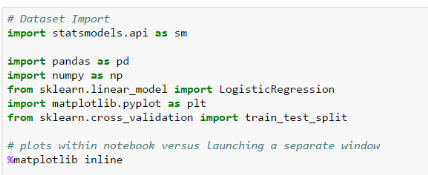
The result here are a bit better in comparison to the initial algorithm. The RSS has a smaller gap between the prediction and true value. This results in less slightly small RMSE, SD,TSS, and R2.

|  |  |  |
| --- | --- | --- |
|  | RUN1 | RUN2 |
| RSS | 22061.88 | 13342.37 |
| RMSE | 6.6031 | 5.135 |
| SD | 13.206 | 10.27 |
| TSS | 278971.9 | 170556.6 |
| R2 | 0.48353 | 0.4846 |

**Logistic Regression:**

1. Importing the data

I will be importing my dataset from Credit Card Default Data. I will use this dataset to predict which customers will default on their credit card.



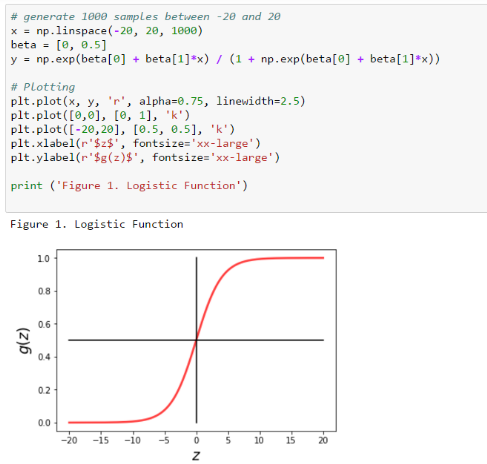
2. The Logistic Function

The standard logistic function takes an input from negative to positive infinity and outputs a value between 0 and 1.

Let be the probability of a positive classification

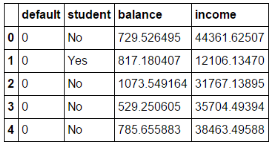
Let be the probability of a negative classification

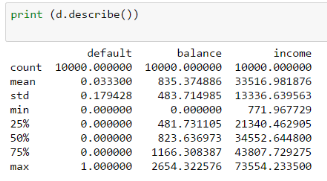
Plotting this function with rang (-20,20) and 1000 observation



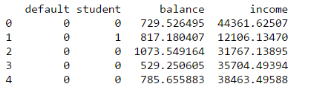
3. Getting, preparing, and visualizing the data

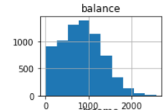
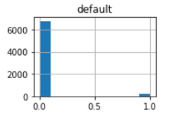
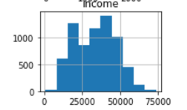
Converting the data file into data frame results in output 3. Using this the data frame I can also describe the data in terms of the averages, standard deviation, ect… for each category.

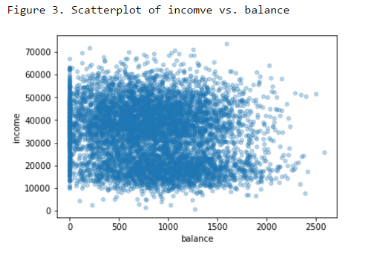
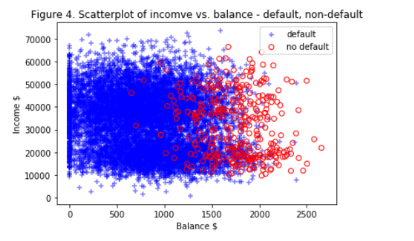




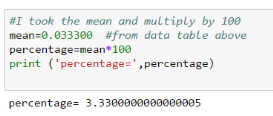
Now I can covert these classification attribute into numeric binary value to get binary classification and histogram of all variables. Below are also scatter plots of income vs. balance. The scatter plots are useful for visualizing the behavior between the two variables.



**Q1) What precentage of individuals in the dataset default?**



**Q2) From 'Figure 4. Scatterplot of income vs. balance - default, non-default':**

Can you identfy an income threshold for default vs. non-default? If so, what is it?

Can you identfy a balance threshold for default vs. non-default? If so, what is it?

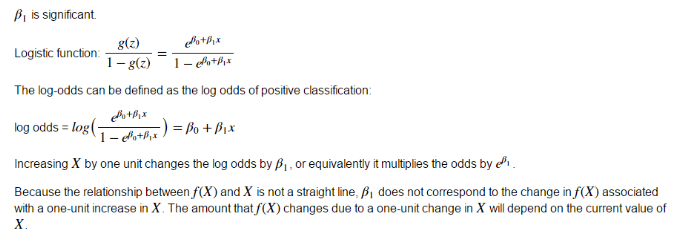
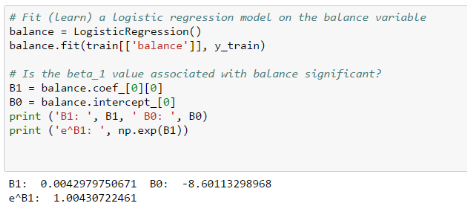
Which individual attribute, i.e., income or balance is likely to be the more accurate predictor?

Provide your answers in the cell below.

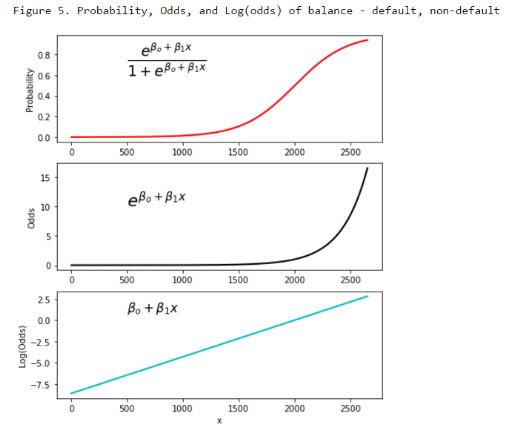


4. Logistic Regression classification

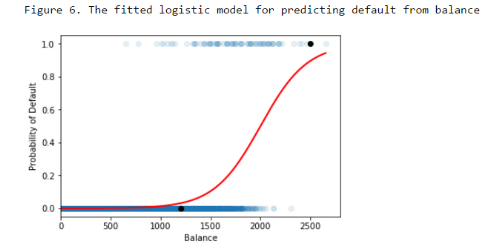
Fitting a logistic regression model on the balance variable



Predicting the probability of default for someone with a balance of $1.2kand $2.5k



And the fitted logistic function over top of the data points

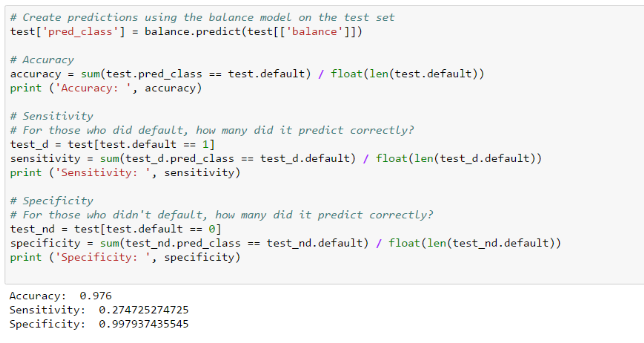


5. Evaluating the model on the training data

Measuring the overall accuracy correctness of the classification is describe by

Sensitivity is measuring the proportion of positives that are correctly identified as such. Also call true positive rate or Recall.

Specificity measures the proportion of negatives that are correctly identified as such. Also called true negative rate.



Class imbalance can result in accuracy measures that are misleading. How does our overall classification accuracy compare to the non-default rate?



What this means is that you could have just guessed not going to default, and would be correct 96.67% of the time.

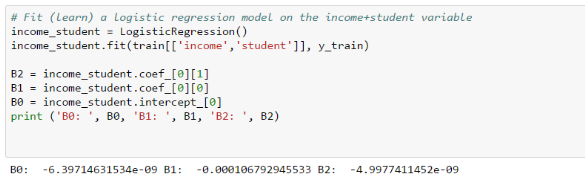
6 – Using Multivariate Logistic Regression to fit the model on the balance variable resulted in,

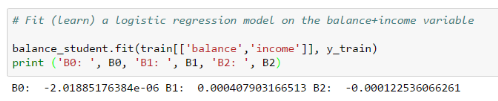


**Q3) For a given income level, who are more likely to default, students or non-students?**

Students are more likely to default because B2<0

**Q4) Which combination of predictors: student, balance, or income has the highest predictive accuracy?**





7. Summary for applying Logistic Regression with scikit-learn

Using Scikit-learn in applying logistic Regression helps with the classification and regression algorithm methods. These algorithm helps to identify which category the data (object or variable) belongs to (output variable takes class labels) and predicts or output a continuous value with that information. In other words,

Regression involves estimating or predicting a response. So given a set of data, it finds the best relationship that represent the set of data.

Classification is identifying group membership. So given a known relationship (i.e. income+balance), it identify the class that the data belongs to.