Machine Learning

I. There are 4 machine learning approaches discussed in this course. You will need to know when to use each approach and how to use each approach. You will also need to know how to pick up the first approach with a given problem.

1. Linear Regression

* It is used to study linear relationships among different variables.
* Use existing data and answers with one or more variables to predict another variable’s values.
* The predicted value is continuous.

2. Logistic Regression

* It is extended from linear regression where the predicted value is discrete, like true or false, yes or no, or multiple-class classification.

3. K-means Clustering

* No answer is known for existing data.
* Unsupervised machine learning.
* Dividing data points into a number of groups such that similar data points are in the same group.
* It can be used for classification problems.

4. K-Nearest Neighbors Algorithm (KNN)

* It is used for classification problems.

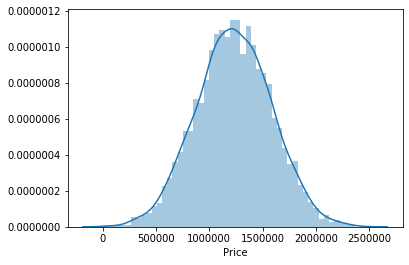
2. The steps of performing the approaches are pretty similar. Practice the questions we did in the past to remember these steps and then you will be fine with solving the questions.

* Prepare the data
* Select an approach
* Split the data into a training set and a test set, if supervised machine learning
* Apply the approach by using the fit function
* Use predict function to predict values (for supervised machine learning)
* Compare the results

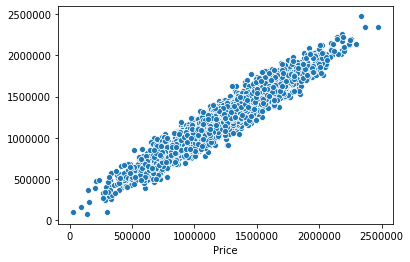
3. These are two projects we have done inside the class.

Q1. Use the USA housing data set to predict house values.

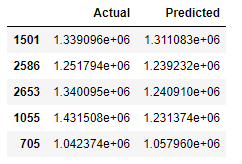
* Read the data set into a DataFrame.
* Practice the head(), info(), describe() and corr() functions.
* Create a distplot chart (bins=50) for the price.



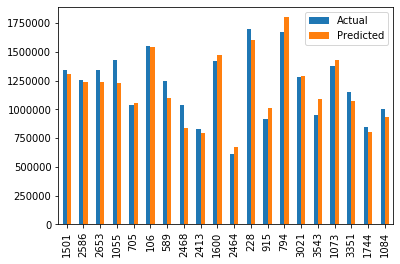
* Select the independent variables from the DataFrame.
* Select the dependent variable from the DataFrame.
* Split the data set into training set and test set.
* Create the Linear Regression Model.
* Train the data by using the fit function.
* Use the predict function to predict values.
* Use the predicted values and actual values to create a scatter chart.



* Create DataFrame by using the predicted results and the actual results. Display the first five entries.



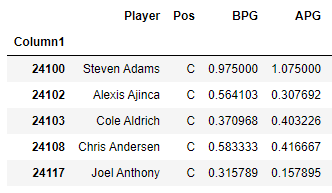
* Create a bar chart based on the first 20 entries of the DataFrame created above.



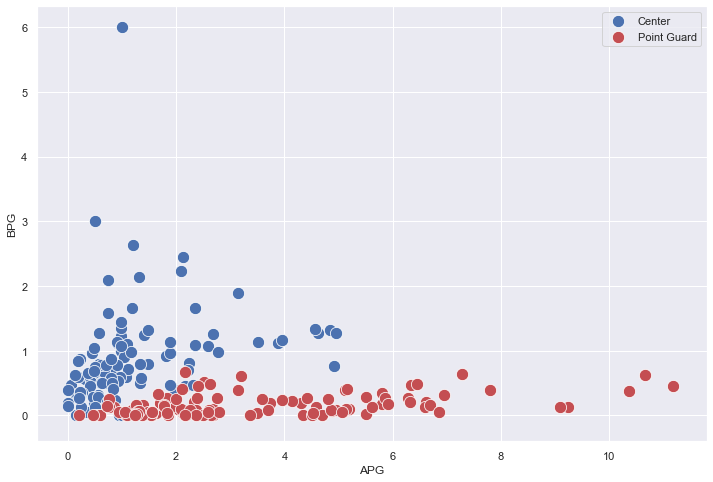
* Predict a single house’s value by using a single record.
* Calculate the mean\_absolute\_error between the predicted values and the actual values.

Q2. Use all of the K-Means, Logistic Regression and K-Nearest-Neighbors (KNN) to solve the NBA point-guard and center positions classification problem.

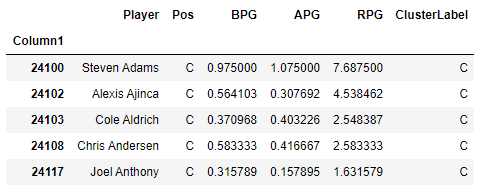
* Read the data set into a DataFrame. Create another DataFrame that it only contains the players either with a PG or a C position.
* Only use the Players, Pos, BPG, APG columns. Keep the players either have BPG > 0 or APG > 0.



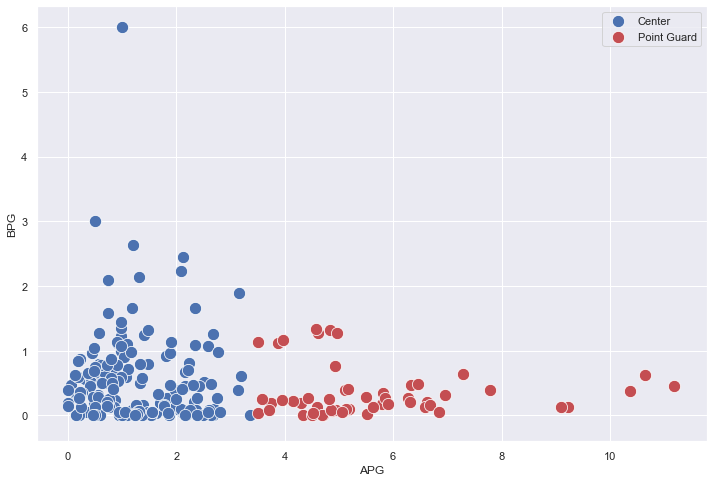
* Create a scatterplot chart based on APG and BPG:



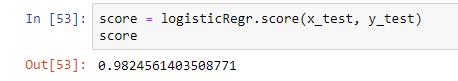
* Create the Kmeans model.
* Train the data using the fit function.
* Display the cluster centers.
* Display the predicted labels. Change the labels to C or PG, save them in a list.
* Add a new column called ClusterLabel and assign the labels list above to the column.



* Calculate the correct rate.
* Create a scatter chart based on the predicted labels.



* Use Logistic Regression to solve the problem.
* Note: logistic Regression takes array-like data as the input. Do not use DataFrame[columns] for the data. Use the columns’ values as the data.
* Split the data set into a training set and a test set.
* Create the logistic regression model.
* Use the fit function to train the model by using the training data set.
* Use the predict function to predict the values based on the test data set.
* In logistic regression model, there is a score function that can be used to directly calculate the prediction correct rate.



* Use KNN to solve the problem. The same as the logistic regression, use array-like data.
* You can use the same training set and test set created above.
* Create the KNN model and define the number of neighbors you use initially.
* Use the fit function to train the model by using the training data set.
* Use the predict function to predict the values based on the test data set.
* Use the score function to display the correct rate of predictions.
* Find the best number of neighbors.
* Determine which approach is the best one to be used to solve this problem.