Experimentor

+ Queries()
+ Pearson()
+ Spearman()
+ AssocRules()
+ DecisionTree()
+ Classification()

Queries

+ queryPearson()
+ querySpearman()
+ queryAssoc()
+ queryComponents()

Spearman

+ datafile
+ xData
+ yData
+ correlation

+ spearmanr()
+ scatter()

Pearson

+ datafile
+ xData
+ yData
+ correlation

+ flatten()
+ pearsonr()
+ scatter()

AssocRules

+ datafile
+ frequentCombonations
+ rules

+ oneHotTransactions()
+ apriori()
+ association_rules()

DecisionTree

+ datafile
+ xData
+ yData
+ regTree

+ preprocessing()
+ train_test_split()
+ decisionTreeRegressor()
+ predict()
+ export_graphviz()

Classification

+ datafile
+ xData
+ yData
+ DTAccuracy
+ NNAccuracy
+ NBAccuracy

+ preprocessing()
+ train_test_split()
+ decisionTreeClassifier()
+ nueralNetworkClassifier()
+ naiveBayesClassifier()

Although our actual python program is a single script, we broke it down into each individual components to better represent the program in UML. Since we only had our database working on a single computer, we ran our queries in mySQL workbench and output them to .csv files and then read them into the python program so each group member had access to the data. This had the additional advantage of helping to streamline our python code by not needed to run the queries each time we tested the program. Each class shown in the diagram represents a single experiment preformed on the database, using data files obtained from the queries. Only four queries were preformed for our five experiments because the decision tree and the classifier both used the same components from the database.