Statistical Learning Project Report

Sprint 2019 - By **Aref Moqadam Mehr**
for **Statistical Learning Course**
under supervision of **Dr. Farahan**i
presented in **Shahid Beheshti University**

Statistical Learning Project Report

This report is regarding to the project for Statistical Learning Course in *Shahid Beheshti University*, Faculty of Mathematics and Computer Science. This report covers the following contents:

- Brief description of the dataset
- Required packages in order to run the project
- Manual for how running the project
- Linear Regression
    - Feature Selection
    - Linear Regression
    - Other methods similar to linear regression
- Classification - I
    - Logistic Regression
    - Linear Discriminant Analysis
    - Quadratic Discriminant Analysis
    - Gaussian Naive Bayes
    - Linear Regression
- Classification - II
    - Tree-Based Regression
    - Random Forest Classifier
    - Decision Tree Classifier
    - Support Vector Classifier
- Clustering
    - K-Means
    - Hierarchical-Clustering
- Neural Network

Dataset

The dataset title is Wine Quality and it has been taken from [UCI Machine Learning Repository](). The data intention is to modeling the wine preferences by data mining from physicochemical properties.
The inputs include objective tests (e.g. PH values) and the output is based on sensory data (median of at least 3 evaluations made by wine experts). Each expert graded the wine quality between 0 (very bad) and 10 (very excellent). The classes are ordered and not balanced (e.g. there are much more normal wines than excellent or poor ones). The dataset is composed of 1599 red wine sample and 4898 white wine samples. Each sample has (11 + output) attributes.
Variables are listed below:

1. fixed acidity
2. volatile acidity
3. citric acid
4. residual sugar
5. chlorides
6. free sulfur dioxide
7. total sulfur dioxide
8. density
9. pH
10. sulphates
11. alcohol
12. quality (output variable which is a score between 0 and 10)

For more about the dataset please follow these links:

- https://archive.ics.uci.edu/ml/datasets/Wine+Quality

Requirements

This project is developed on Python. In order it you will need the following python-packages installed. You can download and install them via pip tool.

- numpy
- sklearn
- scipy
- metrics
- pandas
- keras *(for running NN section)*
- tensorflow *(for running NN section)*

How to Run

In order to run the project, inside the project root folder call the `run.py` with a project name argument. By running each project it will run and illustrate its results.

```
python run.py PROJECT_NAME
```

Where `PROJECT_NAME` could be one of these:

- `classification` or `cl` for running Classification
- `bench-mark` or `bm` for running Bench-Mark
- `feature-selection` or `fs` for running Feature Selection on regression data
- `linear-regression` or `lr` for running Linear Regression
- `reg-all` for running all Regression based modules

- `kmeans` or `km` for running k-means clustering
- `hierarchical-clustering` or `hc` for running Hierarchical Clustering (Agglomerative-Clustering)
- `neural-networks` or `nn` for running Neural Network Classification

You can also run these commands for running projects based on the part they're described in:

- `p1` for running part #1 (Regression)
- `p2` for running part #2 (Classification)
- `p3` for running part #3 (SVM)
- `p4` for running part #4 (Clustering)
- `p5` for running part #5 (Neural Network)

Regression

In this section I have used Best Feature Selection method to select the features with lowest BIC measure. Then, a linear regression method has been applied to these selected features. Below, we briefly discuss these two methods.

Feature Selection

At first, a Best Feature Selection method is applied on the data. To achieve this, every subset of features is selected and a linear regression applied to it. Then they compared together using BIC measure. Below, top 10 subset with lowest BIC is illustrated.

#	Subset of Features	R^2	BIC
1	alcohol	0.182	0.747
2	volatile acidity, alcohol	0.233	0.798
3	density	0.095	0.818
4	residual sugar, alcohol	0.194	0.830
5	free sulfur dioxide, alcohol	0.190	0.833
6	chlorides, alcohol	0.185	0.837
7	fixed acidity, alcohol	0.185	0.837
8	density, alcohol	0.185	0.837
9	sulphates, alcohol	0.184	0.837
10	pH, alcohol	0.183	0.838

Linear Regression

In this section, a Linear Regression model is applied on 1000 bootstrap samples of the data. Hence, after all models are trained, their parameters are averaged together, SE is calculated and with these data now we can calculate T-Statistics and P-Value measure. The result for both Full-Feature model and Best-Feature model has shown below.

Full Feature Model:

Field	Mean COEF	Standard Error	t-Statistics	P-value
fixed acidity	0.0799	0.0015	54.8226	0.000000
volatile acidity	-1.8624	0.0047	-399.1604	0.000000
citric acid	0.0228	0.0034	6.6304	0.000059
residual sugar	0.0875	0.0006	147.4628	0.000000
chlorides	-0.1663	0.0191	-8.7095	0.000006
free sulfur dioxide	0.0038	0.0001	74.1278	0.000000
total sulfur dioxide	-0.0002	0	-13.1209	0.000000
density	-168.8894	1.7635	-95.7697	0.000000
pH	0.744	0.0063	117.828	0.000000
sulphates	0.6561	0.0042	155.1275	0.000000
alcohol	0.1711	0.0022	79.3008	0.000000
	RSS	**TSS**	**R^2**	
Train Error	1080.025	1566.609	0.311	
Test Error	1701.619	2273.722	0.252	

Selected Feature Model:

Field	Mean COEF	Standard Error	t-Statistics	P-value
volatile acidity	-1.9789	0.0044	-445.9857	0.001427
alcohol	0.3243	0.0004	897.018	0.000710
	RSS	**TSS**	**R^2**	
Train Error	1170.381	1570.271	0.255	
Test Error	1753.746	2265.912	0.226	

Model Benchmark

The Linear Regression, Ridge, Lasso, ElasticNet models has been compared together. The results are shown below.

Method	Data	RSS	TSS	R^2
Logistic Regression	Train Error	1991.143	2811.063	0.292
	Test Error	775.305	1025.265	0.244
Ridge Regression	Train Error	2021.061	2811.063	0.281
	Test Error	776.245	1025.265	0.243
Lasso Regression	Train Error	2210.388	2811.063	0.214
	Test Error	851.525	1025.265	0.169
Elastic Net	Train Error	2653.042	2811.063	0.056
	Test Error	994.469	1025.265	0.030

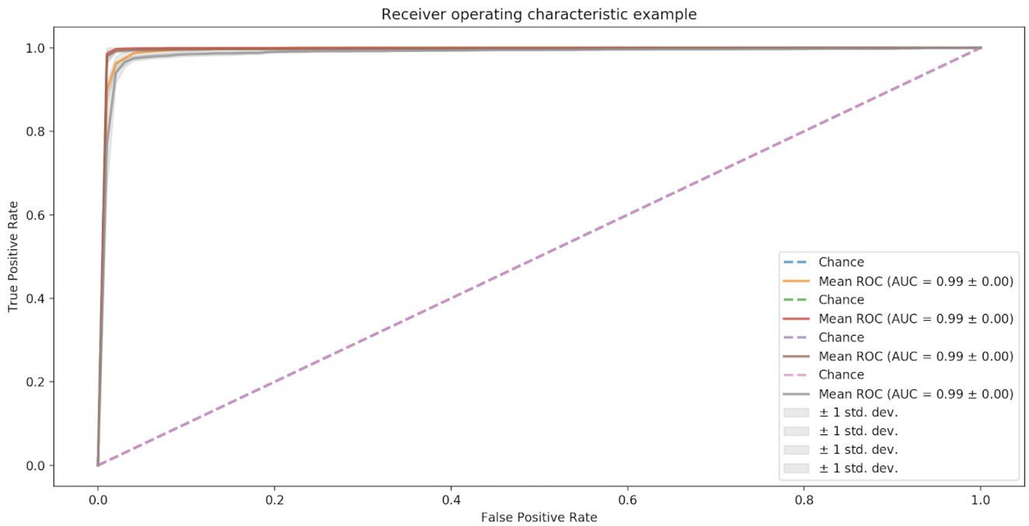
Classification - I

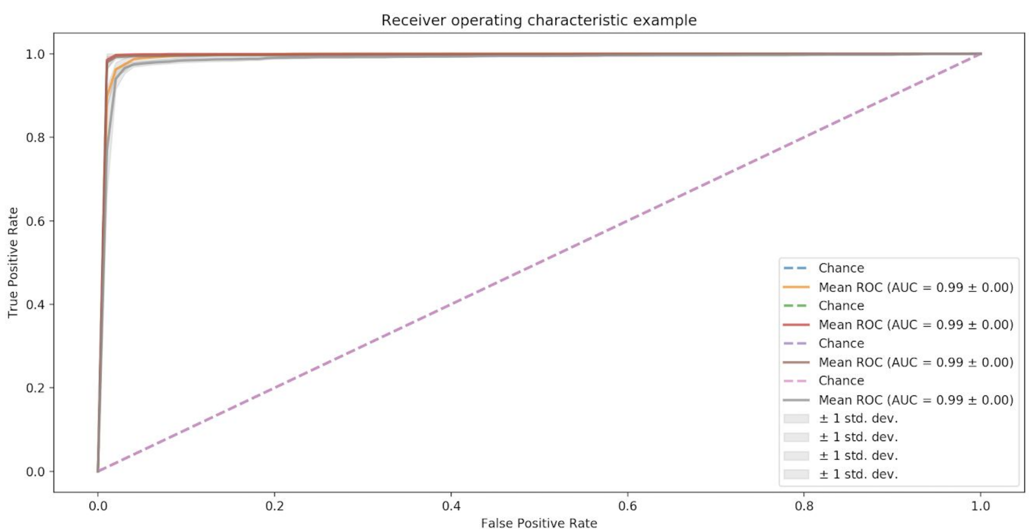
This section some classification model is applied to the data in order to detect either they are red wine or white wine. The original data for red and white wine is located in different files. However, we combine them and try to use machine learning method to distinguish them. The result of these methods are listed below. The model parameters are selected over 5-Fold cross validation and then they are applied to the test set of the data.

Method	Data	RSS	TSS	R2	Precision	Recall	F-Score
Logistic Regression	Train Error	90.00	832.964	0.892	0.970	0.976	0.973
	Test Error	32.00	359.303	0.911	0.976	0.979	0.978
Linear Discriminant Analysis	Train Error	26.00	843.111	0.969	0.992	0.993	0.992

	Test Error	7.00	359.812	0.981	0.994	0.996	0.995
Quadratic Discriminant Analysis	Train Error	67.00	864.928	0.923	0.986	0.975	0.980
	Test Error	20.00	367.443	0.946	0.990	0.982	0.986
Gaussian Naive Bayes	Train Error	140.0	876.597	0.840	0.968	0.951	0.959
	Test Error	45.00	374.056	0.880	0.978	0.962	0.969
Linear Regression	Train Error	120.097	724.029	0.834			
	Test Error	45.993	316.009	0.854			

In addition, the ROC Curve of the mentioned methods using 5-Fold Cross Validation and Leave-One-Out Cross Validation can be visible below.





Classification - II

The goal of this part is to divide the data into two different class using Tree-Based Methods. Then, this method is compared to a few other methods. In order to achieve better results for Tree-Based Regression, we have to choose right depth (also known as K) for the tree. To do so, we have trained trees with different depth (from 2 to 20) and calculate the mean error using Cross-Validation. Then, we selected the K value with the lowest error (we used R^2).

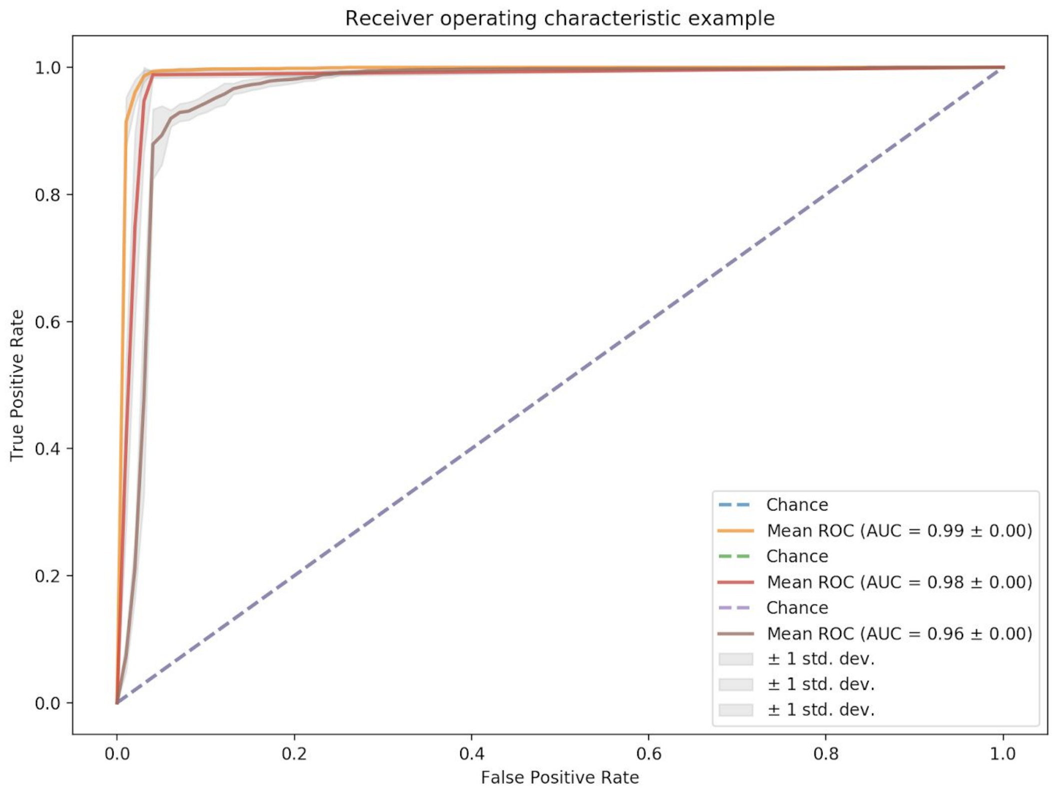
The best found value for k is 15 in this example. And here are the results for a Tree-Based Regression with depth of 15.

Method	Data	RSS	TSS	R^2
Tree-Based Regression	Train Error	273.908	2865.883	0.904
Tree-Based Regression	Test Error	782.631	975.073	0.197

Next, tree other method (Random Forest Classifier, Decision Tree Classifier, and Support Vector Classifier) are compared together in order to find the best classifier for this problem. Here are the results for this comparison.

Method	Data	RSS	TSS	R^2	Precision	Recall	F-Score
Random Forest Classifier	Train Error	120.000	841.074	0.857	0.949	0.980	0.963
	Test Error	52.000	364.372	0.857	0.948	0.980	0.963
Decision Tree Classifier	Train Error	13.000	841.074	0.985	0.996	0.996	0.996
	Test Error	33.000	364.372	0.909	0.978	0.976	0.977
Support Vector Classifier	Train Error	263.000	841.074	0.687	0.898	0.943	0.918
	Test Error	118.000	364.372	0.676	0.903	0.932	0.916

Here is the ROC curve for these methods:



Clustering

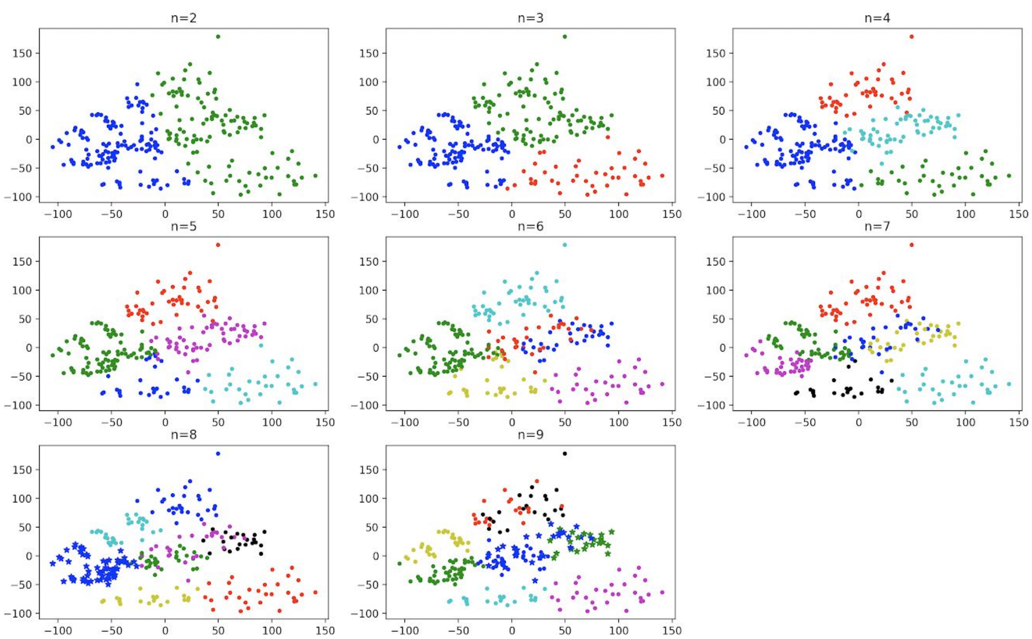
In this section we try to cluster the wines into different classes according to their attributes. The K-Means Clustering and Hierarchical-Clustering has implemented to solve this problem. Then, the results are compared via Silhouette Score. As the results show in the table below, 3 class worked best in both cases. However, the K-Means has higher Silhouette score.

K-Means		**Hierarchical-Clustering**	
N-Cluster (K)	**Silhouette Score**	**N-Cluster**	**Silhouette Score**
2	0.31	2	0.24
3	0.35	3	0.29
4	0.34	4	0.25

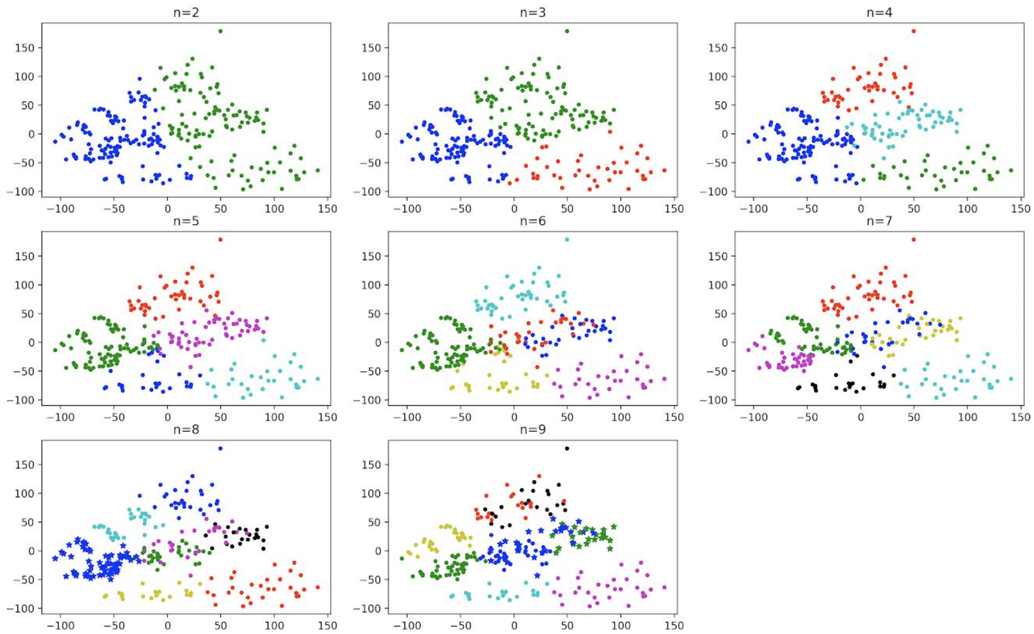
5	0.29	5	0.25
6	0.30	6	0.25
7	0.27	7	0.27
8	0.31	8	0.29
9	0.31	9	0.28
Best (k = 3)	0.35	Best (k = 3)	0.29

Also, the K-Means and Hierarchical-Clustering is plotted by their 2D-PCA plot respectively below:

K-Means:

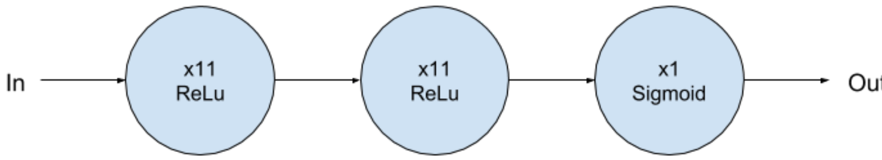


Hierarchical-Clustering:



Neural Network

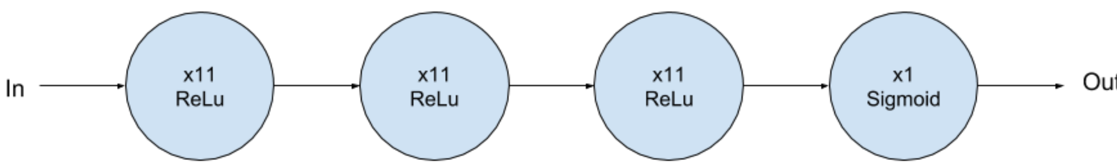
In this section five Multi-Layer Perceptron (MLP) with different architecture has been employed to predict the wine quality according to its attributes. These models are then compared via 5-Fold Cross Validation error. Models are depicted below as well as the results from 5-Fold CV.



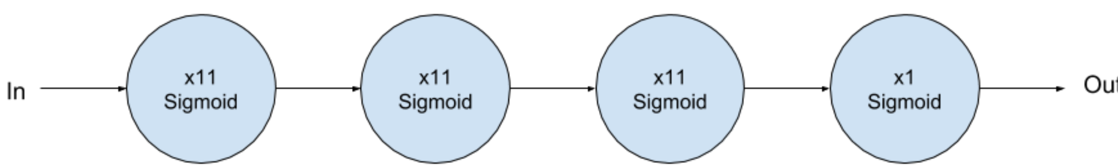
Model #1: Fully Connected MLP

Model #2: Fully Connected MLP

Model #3: Fully Connected MLP



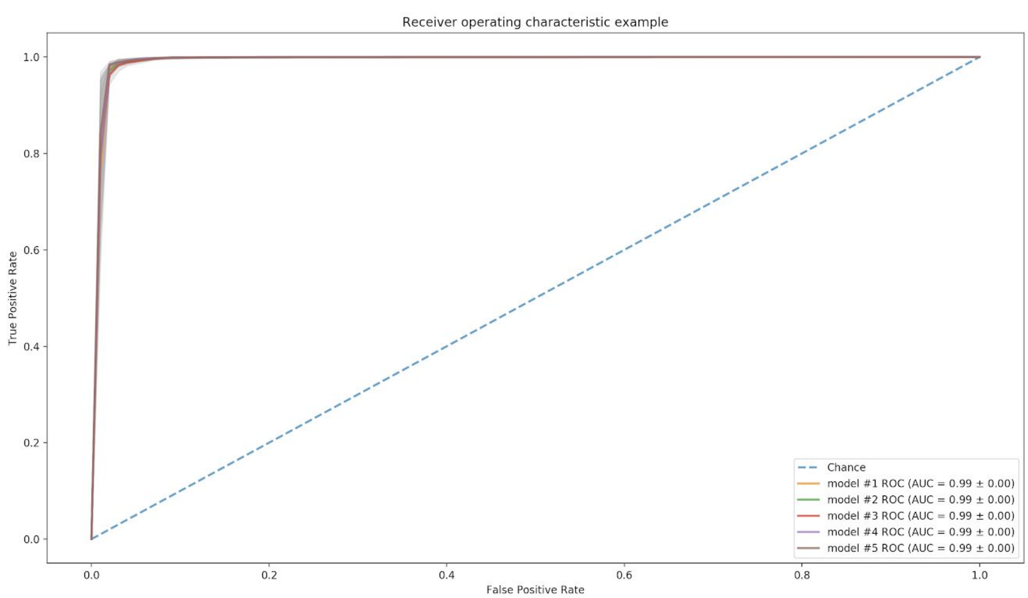
Model #4: Fully Connected MLP

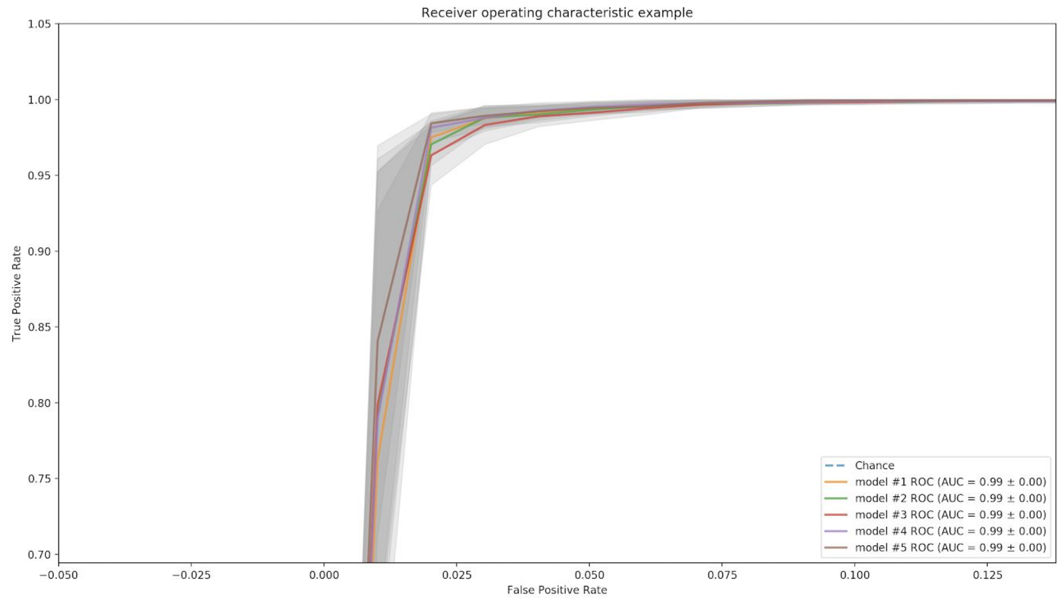


Model #5: Fully Connected MLP

Model	Data	RSS	TSS	R^2	Precision	Recall	F-Score
model #1	Train Error	80.980	839.030	0.903	0.970	0.972	0.971
	Test Error	36.507	366.374	0.900	0.967	0.977	0.972
mode l#2	Train Error	110.708	839.030	0.868	0.939	0.974	0.955
	Test Error	56.772	366.374	0.845	0.928	0.971	0.947
model #3	Train Error	95.325	839.030	0.886	0.953	0.975	0.963
	Test Error	47.468	366.374	0.870	0.949	0.973	0.960
model #4	Train Error	79.120	839.030	0.906	0.968	0.970	0.969
	Test Error	37.372	366.374	0.898	0.965	0.973	0.969
model #5	Train Error	73.548	839.030	0.912	0.969	0.977	0.973
	Test Error	35.177	366.374	0.904	0.966	0.980	0.973

The ROC Curve of these models is illustrated below:





Reference

The code base is on my github at: https://github.com/ArefMq/ML-Project
The dataset is located on: https://archive.ics.uci.edu/ml/datasets/BuddyMove+Dataset
For any question regarding this work please reach me via: aref.moqadam@gmail.com