Machine Learning Feature Selection and Analysis

Austin Dase
Department of Computer and Information Sciences
Towson University
Towson, United States
adase1@students.towson.edu

***Abstract*—A method for analysis of relative feature importance is demonstrated on a previously collected dataset. The method is applied to three types of classification algorithms. The results show how accuracy analysis based on feature subset selection can provide insight into what is often referred to as a 'black box' model.**

Keywords—neural network, machine learning, feature selection, support vector machine, nearest neighbor, wine quality

I. Introduction

Classification using Machine Learning (ML) techniques has grown popular in recent years and has been around as a concept since the mid-1900's. At first, the proposed networks were mainly theory based, as the computational power required for implementing them was far greater than what was available. In the early 1980's, as computing power had matured to a sufficient level, interest began to arise in the ML concepts first theorized in the 1940's and 1950's. By 1987 IEEE held its first International Conference on Neural Networks.

Today ML techniques are applied to a wide variety of real-world problems. One major subset of ML techniques deals with the problem of classification. Based on a dataset of features, which may be continuous or categorical, the instances are put into classes, also called labels. When the labels for the data used to train the classifier are known, the method is called supervised learning, when the labels are unknown it is called unsupervised learning. Classification techniques in their most basic form seek to come up with an accurate model for the data that can be then used to predict the class or label of an unknown instance of the same feature set.

The models that are built by classification algorithms when they are implemented can be massively complex and have a virtually unlimited number of variables. Because of this inherent complexity of the model, it is commonplace to treat the classifier's model as a 'black box' of sorts, where features are fed in and, without regard to the details of the classification process, the predicted class is returned.

In this paper, an attempt was made at extracting additional meaning from the model of three of the most well known classification methods: Neural Networks (NN), Support Vector Machines (SVM) and K-Nearest Neighbor (KNN). Statistical analysis, inspired by feature selection research served as the framework for the method of analysis used to extract meaning from classification attempts on a known dataset by each of the three aforementioned classification techniques.

II. Related Work

A. Wine Quality Data Set

The data set used for classification, the wine quality dataset was first used by P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. and is available online at the University of California, Irvine Machine Learning Repository [2]. In [2] an attempt to identify the relative importance of features for the SVM model was made using calculations for relative importance based on fluctuations in variance for each input variable when all other input variables are held at their average. This method was used to give a score, out of 100, of how relatively important each feature was to the model. [2] applied this technique only to the SVM model. In the present experiment, only the red wine data set from [2] was analyzed, this choice was made to allow the experiment to complete in a reasonable amount of time. Further, the predictive accuracy of all three classifiers in [2] was higher for the red wine data than for the white wine data. No attempt was made at explaining the reasons for this observation.

B. Feature Selection

Identify applicable funding agency here. If none, delete this text box.

Other papers have explored the field of feature selection from different angles. Some analyzing and comparing methods of feature selection and others using feature selection to draw conclusions about various features. Most feature selection methods attempt to either analyze the features themselves and remove any unnecessary or confounding inputs or they analyze the predictive accuracy of the same network trained on different feature subsets. The second of the two approaches, first introduced by Kohavi and John [3], is commonly referred to as a 'wrapper' approach. There are three general categories of feature selection methods. Filter methods measure the relevance of features in determining the class label and are removed if they do not meet the threshold of what is considered relevant. Wrapper methods evaluate feature subset selection based on predictor performance. Embedded methods are both filter and wrapper methods but applied within the training phase. For example, in [4] weights within a NN serve as a rank and weights that do not meet a certain threshold value after training are removed, in this way, redundant connections are identified and pruned from the network. This embedded filter method approach is called Network Pruning.

Chandrashekar and Sahin conduct a survey of feature selection methods in [5]. However, they do not consider this exhaustive combination approach as it is too "computationally intensive for larger datasets", instead they opt for variations on that method where search algorithms are used to determine which of all possible subset combinations to use.

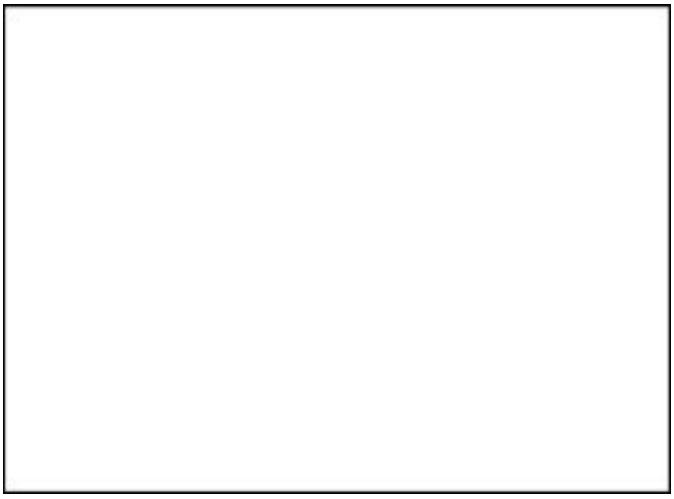
In [6] Hall gives a concrete definition of a good input feature, "Good feature subsets contain features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other.".

In general, much of the previous work has been done around the context of analyzing feature subsets to improve the predictive accuracy of the model. In the present paper, an attempt is made at applying a wrapper style feature section approach with the goal of gaining insight into the meaning of the model. In the context of the wine quality dataset, from [2], the model is interpreted in an attempt to meaningfully understand the impact the given physicochemical properties have on the subjective quality of the wine. This method of analysis is directly applied to the wine quality dataset but could be used in a variety of different ML applications.

III. APPROACH

A. Exploratory Data Analysis

Before constructing the models and performing the experiment, initial Exploratory Data Analysis was done on the data set. As mentioned in the Related Work section, the decision was made to only analyze the red wine data set. All three classifiers performed consistently better on the red wine data and therefore, it can be assumed that either the properties that were measured as each feature either have a greater degree of significance on the corresponding classification or that the data for red wines were better suited for use in classification. The dataset consists of 1599 red wine samples, each with 11 measured results of physicochemical tests and their corresponding class label, a grade, on a scale from 0 (very bad) to 10 (excellent). The details of the data collection methods and grading are available in [2]. The histogram of



class labels is shown in Fig. 53. To perform an initial analysis of the features and to provide a benchmark against which to compare the experimental results, each of the features was regressed against the known labels and an r squared correlation was determined for that feature, see Figures 3 through 13. From this analysis, none of the features have a large r squared value and therefore would not be considered to have a direct correlation to the true label. However, the relative r squared values can be compared. When comparing the relative r squared values, it can be predicted that alcohol seems to be the most correlated to the true label see Fig. 1. With the same regard, volatile acidity has the second highest correlation to the true label.

B. Build the Models

The first step in the process was to implement the classification methods for use with the dataset. Each classifier was built with the goals of reasonable performance and accuracy. In [2] with a tolerance (T) of .05. NN achieved a 59.1% accuracy and SVM achieved 62.4%. Those results were used as a benchmark against the variations of each of the classifiers.

The NN required the most modification of hyperparameters to achieve acceptable results. To achieve desired test accuracy, multiple variations on the NN were attempted and scored through baseline tests. Using this method, the NN was able to achieve a maximum predictive accuracy of 99.08% with a tolerance of .01, see Fig 1. These results show a significant improvement in accuracy from [2], however, the execution time required to achieve those results was prohibitively long at 128.77 seconds for one execution. In the interest of runtime, a less complex,

less accurate model was used in the final comparison, that model achieved a baseline classification rate of 58.26% in 12.98 seconds, which was considered acceptable for the experiment. The SVM and KNN classifiers achieved initial testing accuracy close to the benchmark from [2] and therefore were not modified further.

TABLE I.

Classifier Baseline Test Results		
Classifier	***Accuracy (%)***	***Execution Time (seconds)***
K- Nearest Neighbor	57.63	.14
Support Vector Machine	83.98	.25
Neural Network	58.26	12.895

C. Perform Experiment

After each classifier was successfully implemented and achieved a reasonable predictive accuracy and execution time for an input of all features, the collection of predictive accuracy results could begin. Given n features, there are 2^{n-1} possible combinations of feature subsets when zero length subsets are ignored. In this case, 1024 unique subsets of features exist. Each of those feature subsets was used to train as well as test the model. The data was split randomly into two sets, 75% of instances to be used for training and 25% of instances to be used for testing. For each feature subset, the model was re-trained using only those features as inputs. Additionally, for each feature subset, a new, random selection of test and training data was determined.

The NN model consisted of 5 hidden layers of 200 nodes, one input layer, and one output layer. Instances were fed to the model in batches of 100. 50 iterations of training, using the training portion of the data, were done each time the model was re-trained. A SoftMax cross entropy function was used as the loss function and the Adam Optimization Algorithm was used as an optimizer. KNN takes one hyperparameter K which is the number of neighbors to use as the cutoff. In this case, a K value of 15 was used. All other values for the KNN implementation were left to their defaults. SVM takes several hyperparameters, the most important of which is the kernel function, in this case, a Radial Basis Function (RBF) kernel, which was the default, was used. Other hyperparameters include C and gamma, the values used in this experiment were adopted from the values used in [2]. All other parameters were left at their default values.

For each model, after each iteration, the predictive accuracy and the features used in prediction were logged to a file so that separate analysis could be performed on the results.

IV. Evaluation Methodology

The results were analyzed both in the aggregate as well as individually by class. First, a matrix of dimensions (2^{n-1} , n) was allocated to store the results, with each column corresponding to a feature. Then, for each instance of predictive accuracy, the accuracy rate was recorded once in each column corresponding to the features that were removed. For example, when features X_1 and X_2 were removed and accuracy A was achieved, A is recorded once in column 1 and once in column 2. This process was repeated for each accuracy score for all possible feature subsets.

After aggregating the results into a single matrix, statistical data analysis was done to extract information about feature importance. First, each column was analyzed visually by plotting the histogram of the counts of each accuracy, the counts were scaled by dividing the count by the number of observations in each bin multiplied by the width of the bin. Then, the same histogram was overlaid with a normal distribution centered at the mean and with a standard deviation equal to the mean and standard deviation of all the results in that column, see Figures 16 through 49. Next, the mean of each of the columns was taken and compared to the means of the other columns, see Table II.

The above process was carried out for each of the three models, KNN, SVM and NN. The results are plotted in Figures 50, 51 and 52. As a reference, horizontal lines indicating, 1, 2 and 2.5 standard deviations from the overall mean were overlaid on the plot of means. The results of all the above tests were used to draw conclusions about the relative importance of each feature to the quality of the wine.

In this method of analysis, scored are recorded for features that are removed from the input list. Therefore, features that are observed to have lower mean predictive accuracy, had a positive impact on the predictive accuracy of the model. Conversely, those features that are observed to result in a higher mean predictive accuracy, have a negative impact on the overall predictive power of the model. The features that have a significantly negative (greater than two standard deviations from the mean) impact on the predictive accuracy of the model can be interpreted in several ways. First, they could make good candidates for removal from the inputs to the model and most likely will improve the accuracy. Secondly, it can be concluded that those features are less important to the real-life scenario the classifier is attempting to model. Additionally, the features that are observed to have a significantly positive impact on the predictive accuracy of the model can be considered essential to the accuracy of the model. It can also be concluded that those features also have a significance in the real-world scenario that the classifier is attempting to model. For example, in the following case, alcohol had a significant positive impact on the overall predictive accuracy of the model. The observed accuracy results, when alcohol was removed from the inputs was 55.04 for KNN and 48.32 for NN, both of which were more than 2

standard deviations from the mean accuracy of all results for the corresponding classification method.

TABLE II.

	Mean Predictive Accuracy by Feature		
Feature	***Accuracy by Classifier (%)***		
	KNN	***SVM***	***NN***
Fixed Acidity	57.34	64.09	51.29
Volatile Acidity	57.06	66.61	50.51
Citric Acid	57.25	67.04	51.25
Residual Sugar	57.51	65.42	50.94
Chlorides	57.27	67.35	51.15
Free Sulfur Dioxide	57.84	61.18	51.45
Total Sulfur Dioxide	57.75	**58.15**	50.83
Density	57.27	67.43	51.12
pH	57.24	67.20	51.08
Sulfates	57.05	67.76	50.97
Alcohol	**55.04**	62.83	**48.62**
Mean Accuracy	57.15	64.91	50.84
Standard Deviation	.007	.029	.007

a. Bolded values are further than 2 standard deviations from the mean

V. Methods

The experiments were carried out using the Anaconda3 distribution of Python 3.6.4. External ML libraries were used to implement the classifiers. NN classifier was implemented using the TensorFlow library provided by Google, version 1.8 [8]. SVM and KNN were implemented using scikit-learn's machine learning libraries version 0.19.1 [7]. All experiments reported in this work were written in Python and conducted in a Windows 10 environment with an Intel® Core™ i5-8600K 6 core CPU, as well as an NVIDIA GeForce GTX 1060 6GB GPU and 16GB DDR4 RAM.

VI. Experimental Results and Discussion

It can clearly be seen that there has been some value gained by analysis of the predictive results for all possible subsets of features. In the present case the data suggest that alcohol has the greatest impact on the KNN and NN models, while interestingly, SVM saw total sulfur dioxide as the factor with the greatest impact with alcohol having the third greatest impact. Furthermore, it is interesting to note that the spread of most of the features' mean is within one standard deviation of the global mean. This could indicate that each of the available features is independently significant to the overall classification.

It is also interesting to note the vast superiority of SVM in accuracy vs NN and KNN. NN was able to achieve a comparable accuracy rate but only after creating a network so complex that the execution time for one test was over one minute, while SVM achieved a high accuracy rate with sub-second execution time. This difference is even more pronounced than observed in [2]. Indeed, the NN with enough training iterations was able to achieve a predictive accuracy of 99%, however, the execution time required to achieve this result was 128.77 seconds which was not feasible for the goals of this experiment.

Based on the initial benchmark of regression analysis for each feature vs the true label. It was expected the alcohol and volatile acidity would have the strongest impact on the observed predictive accuracy. This expectation proved true of alcohol in two of the three classifiers, NN and KNN, and did not hold true in the SVM model. The expected impact of volatile acidity was not observed by any of the classifiers. SVM also identified total Sulphur dioxide as having the greatest significance of all the features, a result that the initial regression did not indicate. These observations indicate that the classification algorithms provide a deeper insight into the relationship of the features to their true class. In this case, the relationship between wine's chemical properties and its perceived quality.

Another interesting observation was the distribution of the results for each classifier. The distribution of the aggregate results, as well as the distribution of the results for individual features, visually looked as if it might be normally distributed. To analyze that hypothesis, a Shapiro–Wilk test of normality was applied. The results of the Shapiro–Wilk rejected the null hypothesis that the observed distributions came from the normal distribution. The results of p values for each of those tests are listed in Table 1.

TABLE III.

Test of Normality of Results Distribution - NN		
Feature	***Test Stat***	***p-value***
Fixed Acidity	.99	2.235e-6
Volatile Acidity	.98	8.928e-10
Citric Acid	.98	9.959e-10
Residual Sugar	.98	3.444e-10

Chlorides	.98	2.511e-9
Free Sulfur Dioxide	.97	1.485e-11
Total Sulfur Dioxide	.94	2.619e-19
Density	.98	3.072e-8
pH	.98	5.543e-9
Sulfates	.98	1.584e-10
Alcohol	.98	2.69e-7
Aggregate (All Features)	.97	3.622e-37

VII. Conclusion

It clear that there is a relationship between the inputs to a classification ML model and its predictive power. Quantifying the strength of that relationship, based on feature subset predictive accuracy, is something that the present experiment sought to accomplish. From the initial analysis of the correlation between features and their true labels, two features stood out as possibly having the strongest correlation to the true label, alcohol and volatile acidity. After performing the experiment, the results seem to confirm the hypothesis that alcohol levels in wine tend to have a strong correlation to the perceived quality of the wine. However, additional insight was gained after analyzing the experimental results. Alcohol had a significant impact on the accuracy of NN and KNN but did not have a significant impact on SVM. Furthermore, unlike NN and KNN, total sulfur dioxide had the most significant impact on SVM accuracy. These results could have a variety of implications that would require further study to confirm. Potentially SVM, because it's classification accuracy is also generally higher than NN and KNN, may be providing a better insight into the true nature of the relative significance of each of the features on the quality of the wine. Alternatively, due to differences in methodology, these results could imply that the impact of a feature on a ML classifier's accuracy may have more to do with the classifier being used than on the real-world problem being simulated. It is the opinion of the researcher that the former of these possibilities is the most likely. If the assumption that SVM provided the best model for the real-world scenario being simulated holds true, the implications of the experiment are powerful. Indeed, analysis of the mean accuracy for each feature when removed from the inputs, Fig. 51, showed total sulfur dioxide as having the strongest impact on predictive accuracy; a result that could not be predicted with simple regression analysis. This would imply that this method of analyzing subsets of features in the way described previously, can provide researchers with better insight into the meaning of a model that has often been treated as a 'black box'. While this method was applied specifically to the wine dataset from [2], this same method of analysis could easily be applied to any dataset and would only be bound by limitations of computational resources, depending on the number of features being analyzed.

References

[1] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), 386-408.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Modeling wine preferences by data mining from physicochemical properties," Decision Support Systems, vol. 47, no. 4, pp. 547–553, 2009.

[3] R. Kohavi and G. H. John, "Wrappers for feature subset selection," Artificial Intelligence, vol. 97, no. 1-2, pp. 273–324, 1997.

[4] R. Setiono and H. Liu, "Neural-network feature selector," IEEE Transactions on Neural Networks, vol. 8, no. 3, pp. 654–662, 1997.

[5] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," Computers and Electrical Engineering, no. 40, pp. 16–28, 2014.

[6] M. Hall, "Correlation-based Feature Selection for Machine Learning." Unpublished.

[7] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[8] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

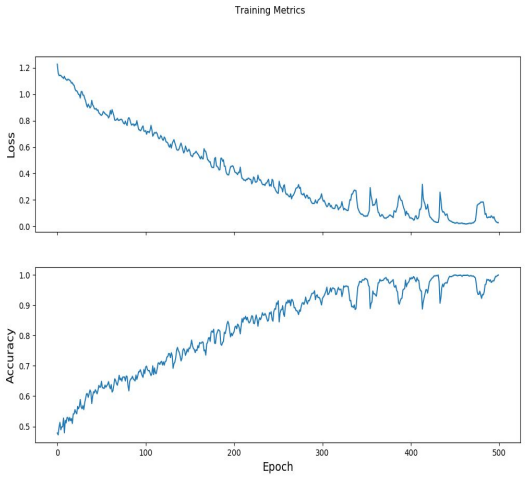


Figure 2 NN Training Metrics

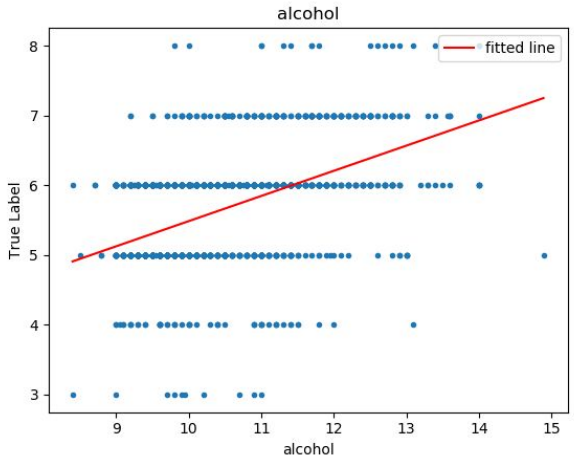


Figure 3

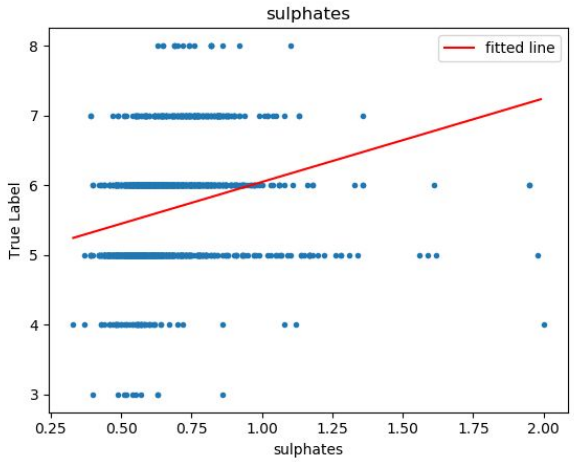


Figure 4

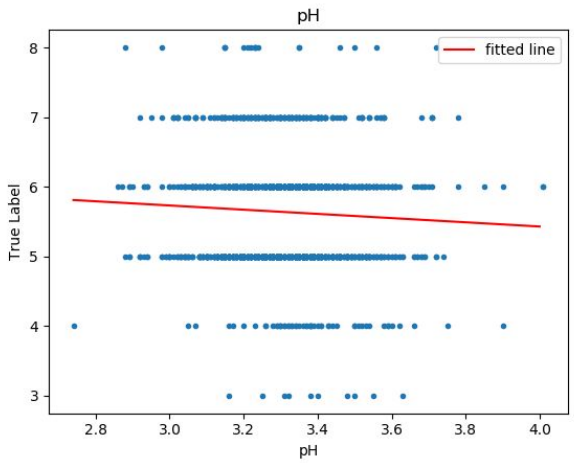


Figure 5

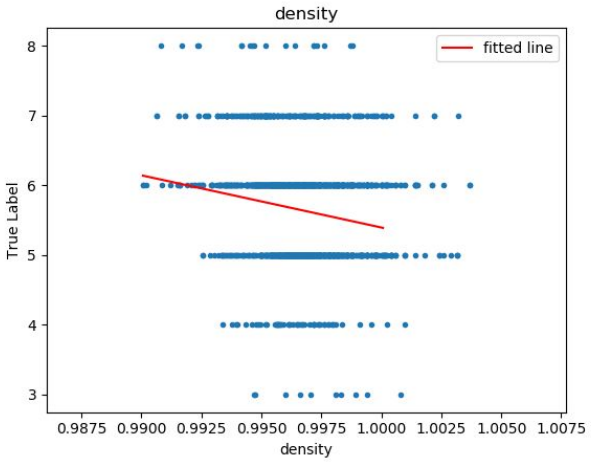


Figure 6

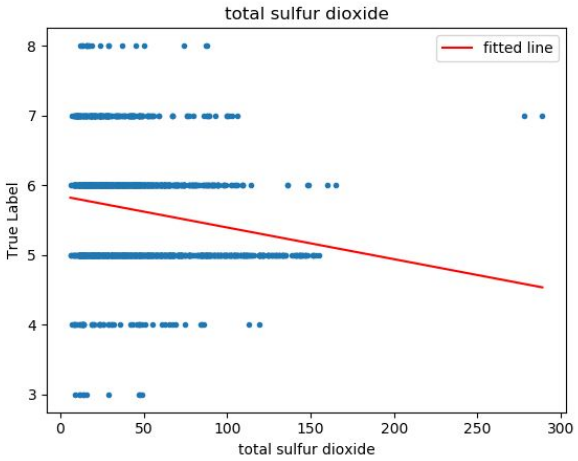


Figure 7

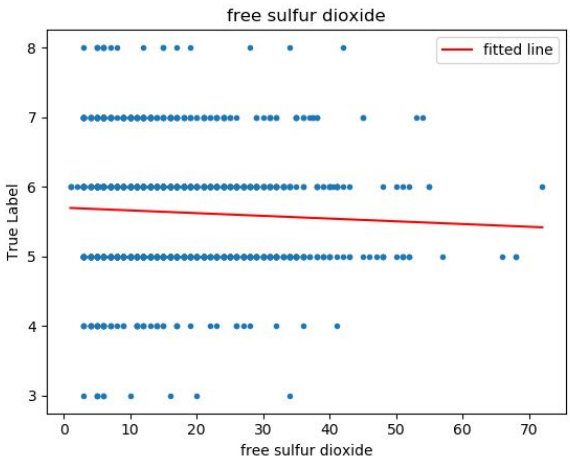


Figure 8

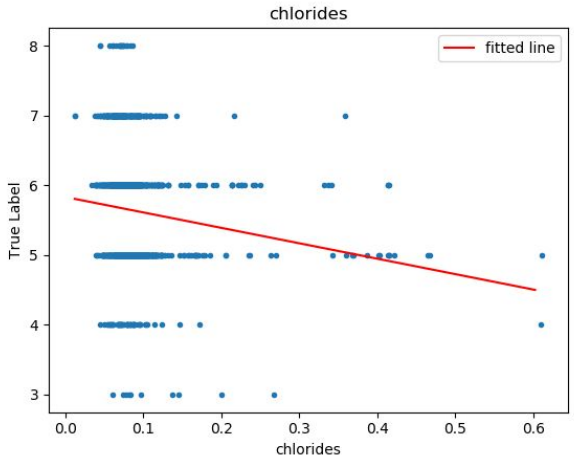


Figure 9

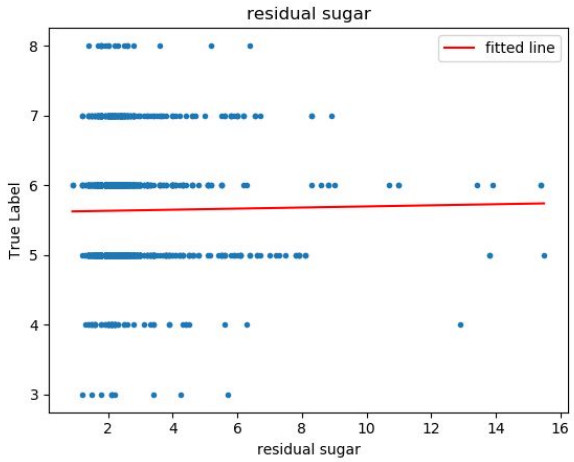


Figure 10

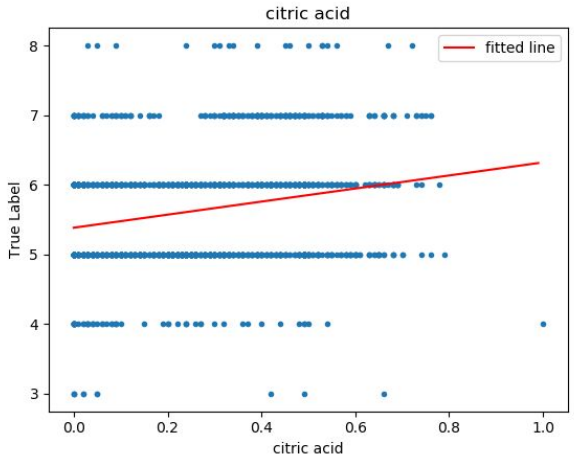


Figure 11

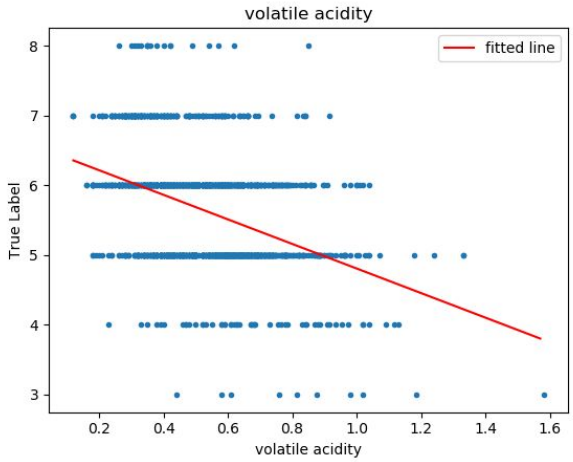


Figure 12

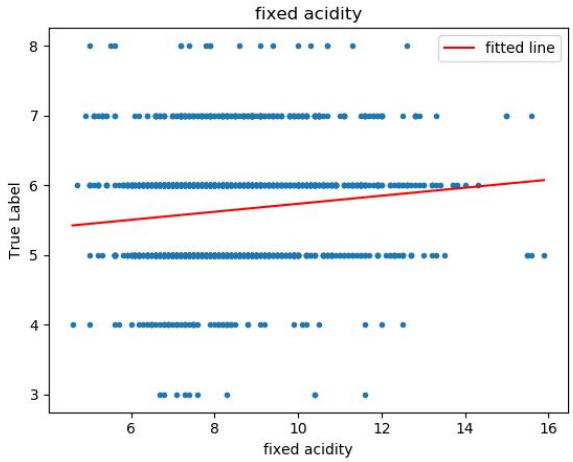


Figure 13

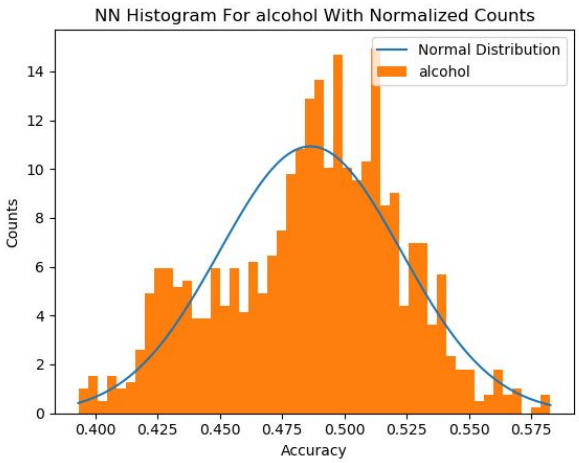


Figure 14

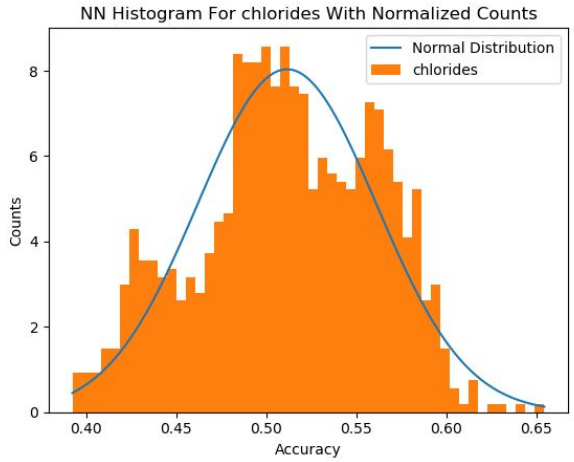


Figure 15

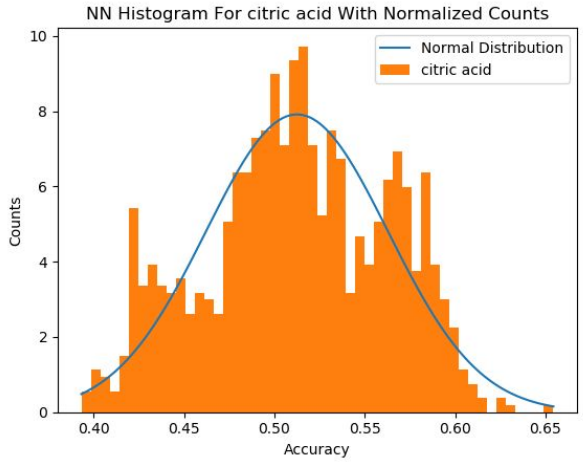


Figure 16

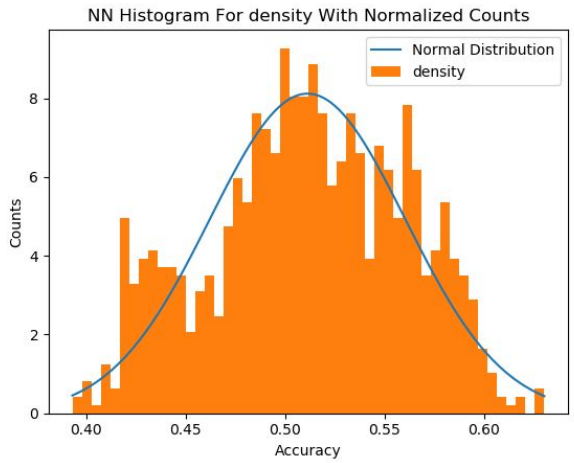


Figure 17

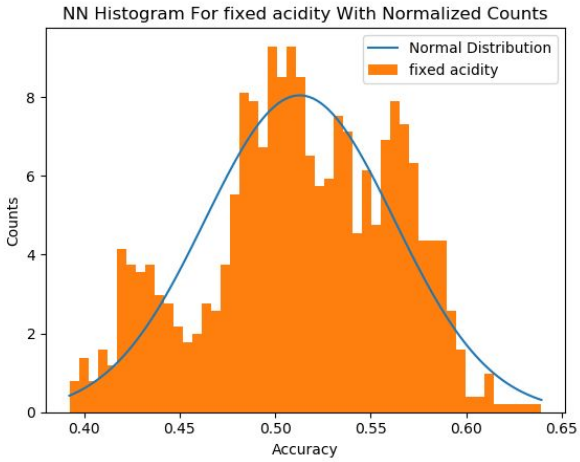


Figure 18

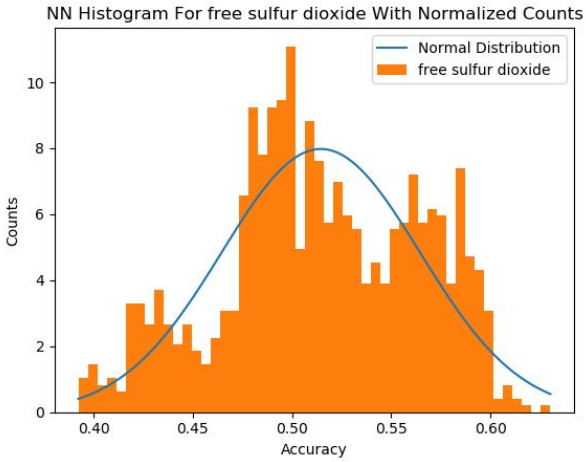


Figure 19

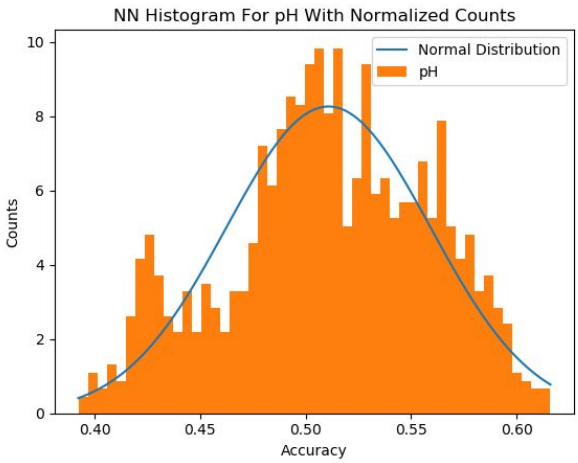


Figure 20

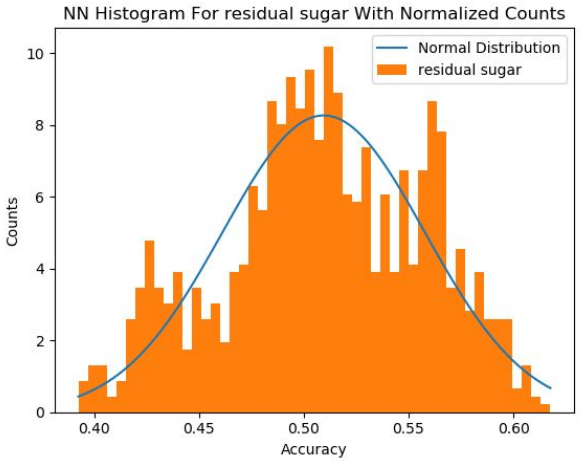


Figure 21

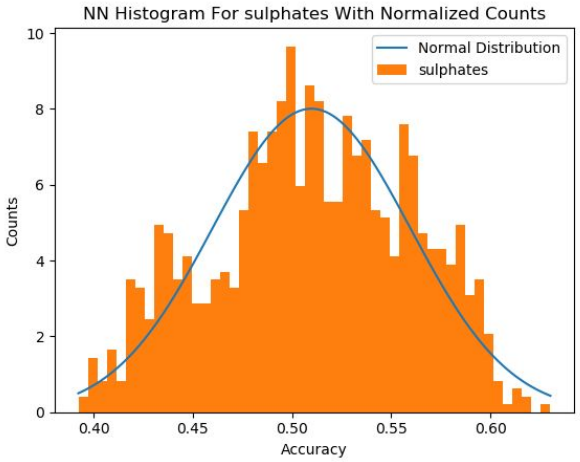


Figure 22

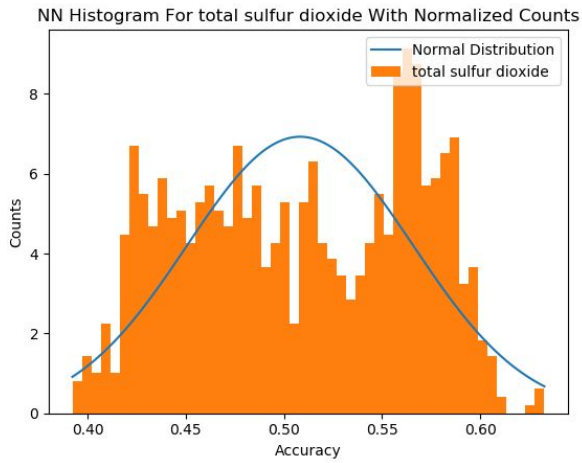


Figure 23

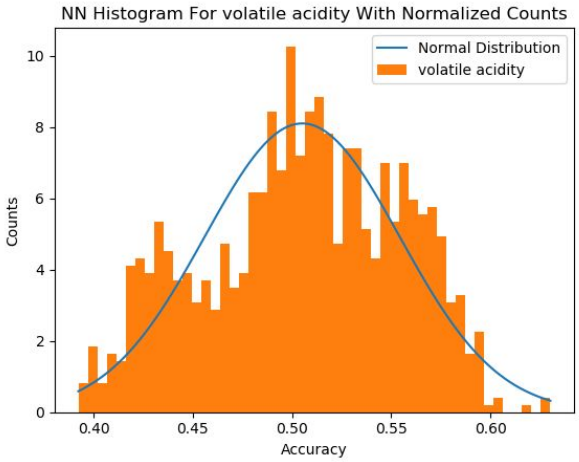


Figure 24

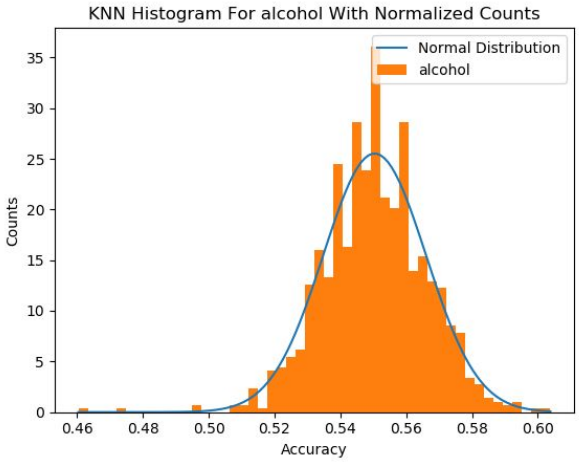


Figure 25

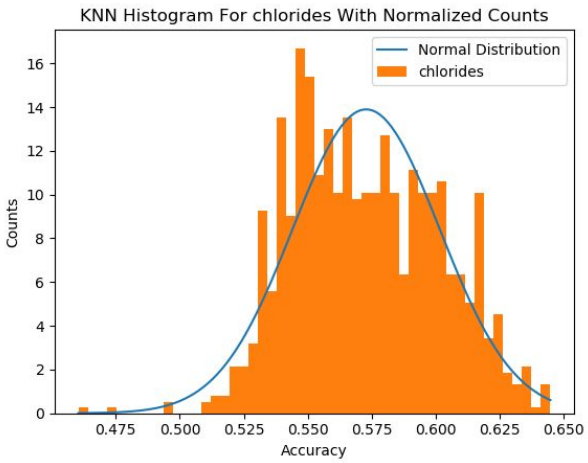


Figure 26

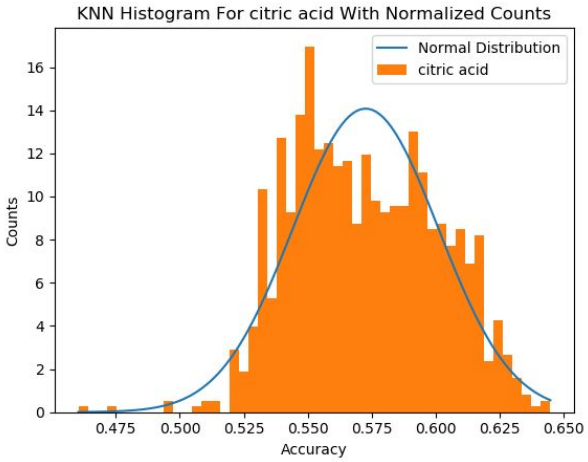


Figure 27

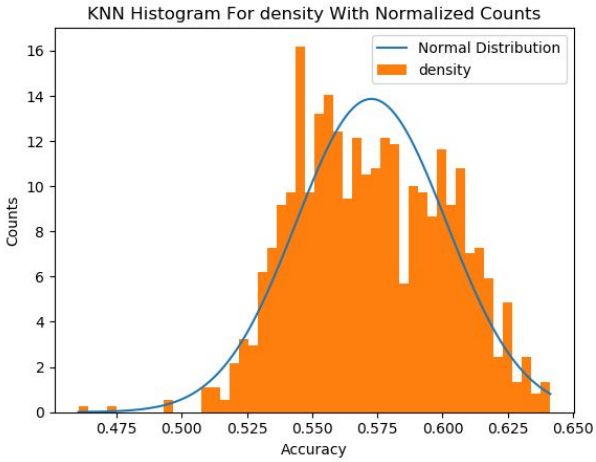


Figure 28

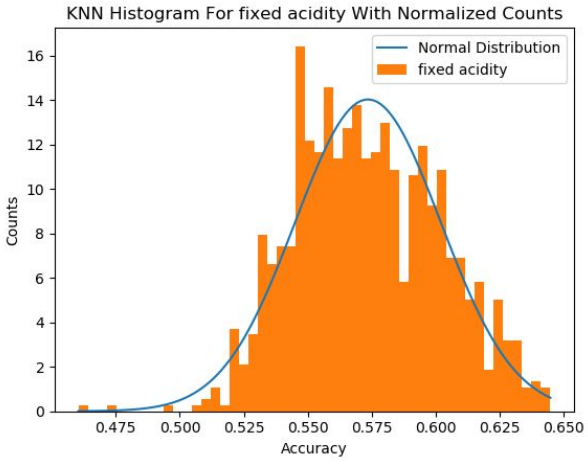


Figure 29

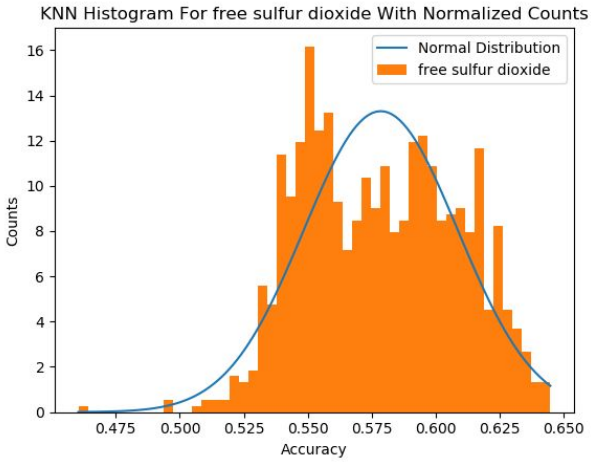


Figure 30

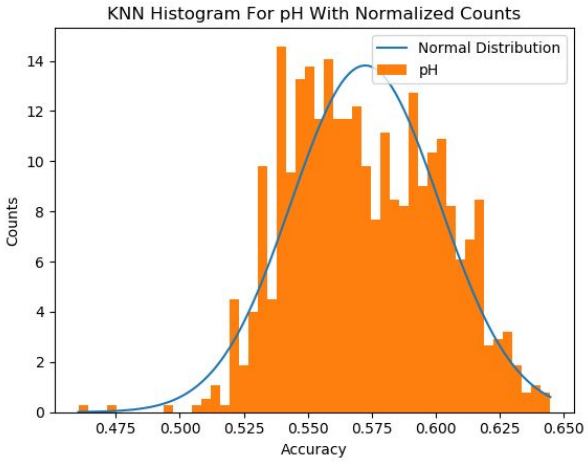


Figure 31

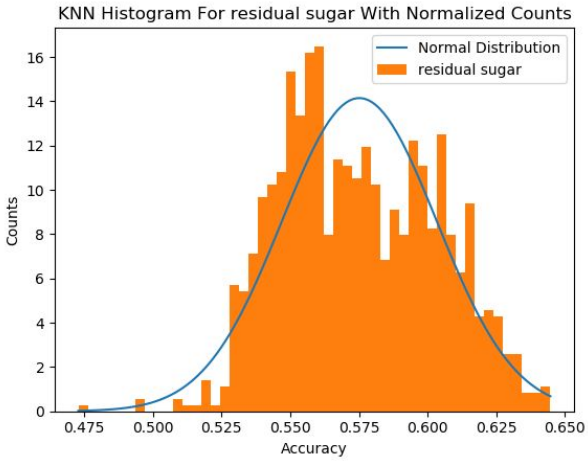


Figure 32

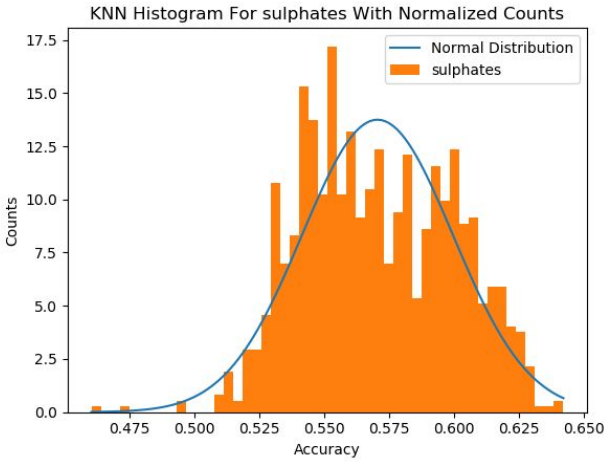


Figure 33

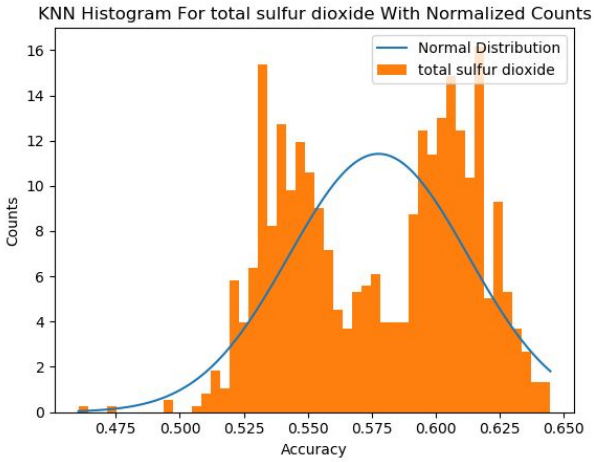


Figure 34

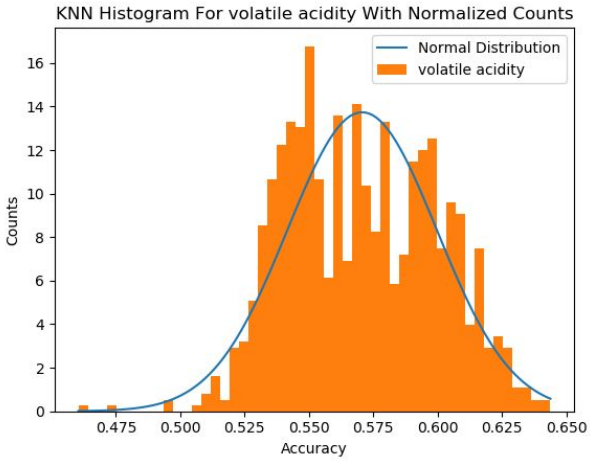


Figure 35

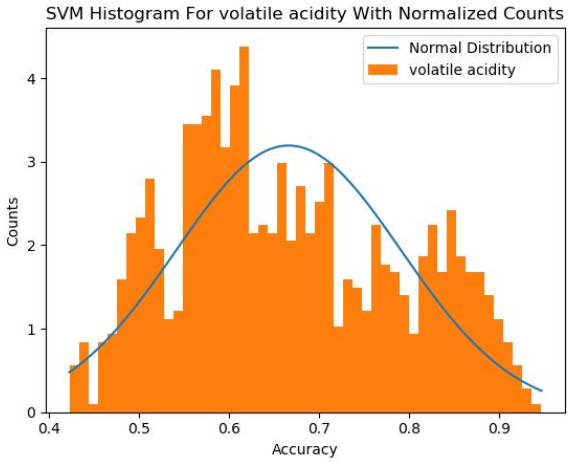


Figure 36

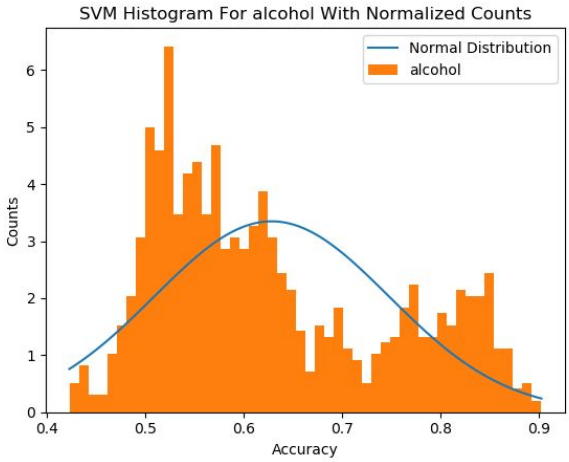


Figure 37

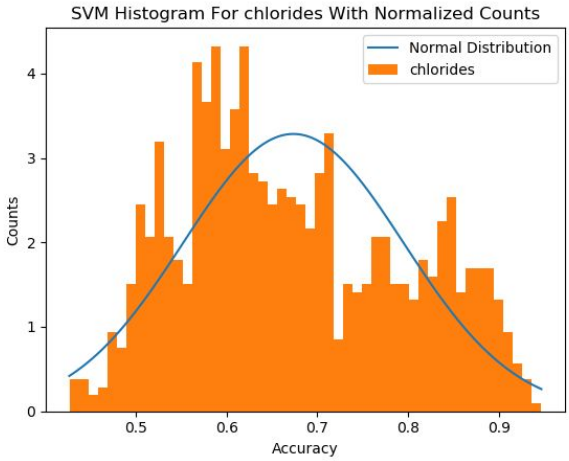


Figure 38

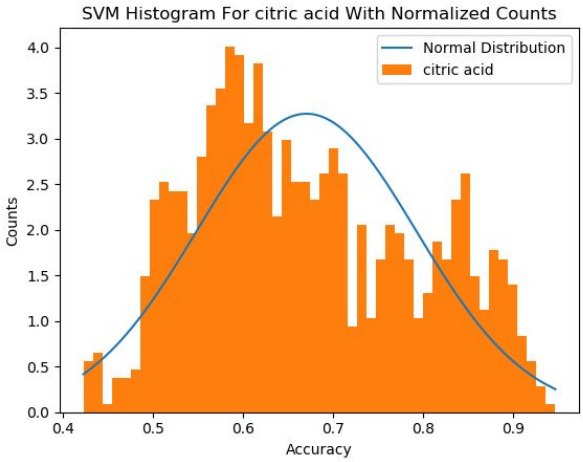


Figure 39

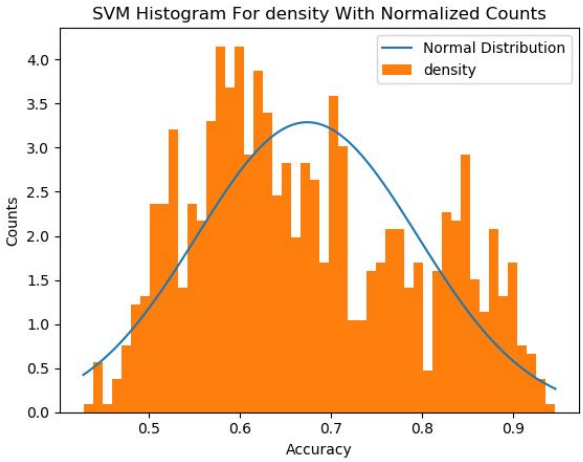


Figure 40

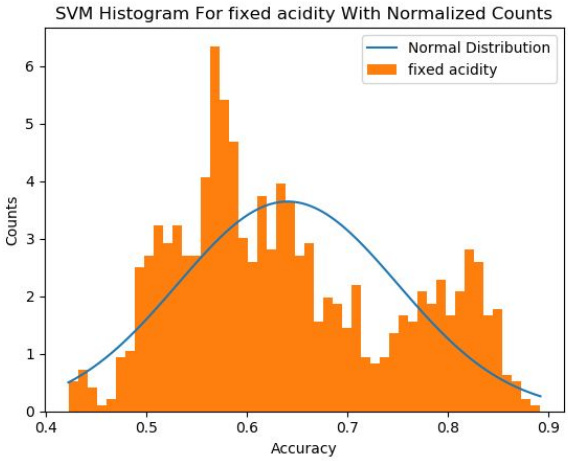


Figure 41

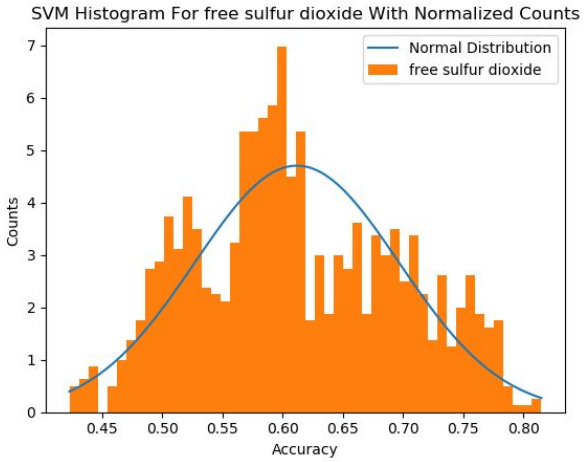


Figure 42

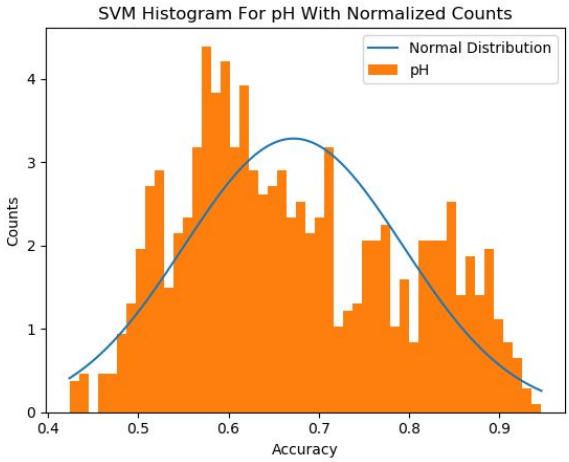


Figure 43

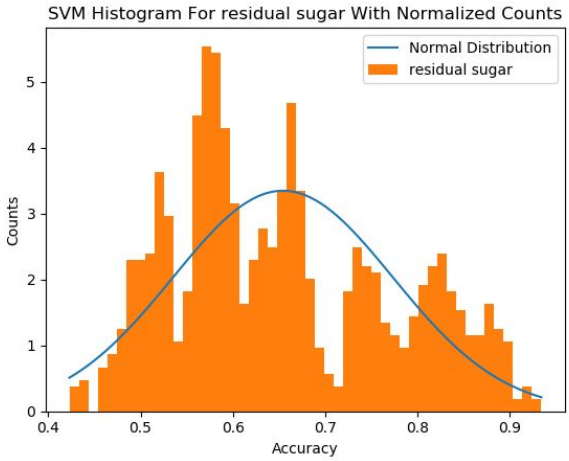


Figure 44

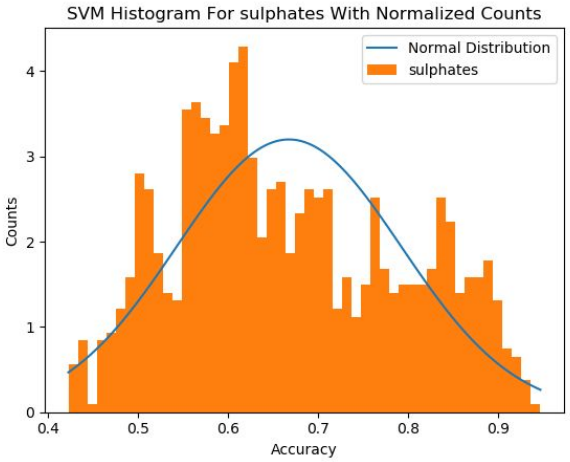


Figure 45

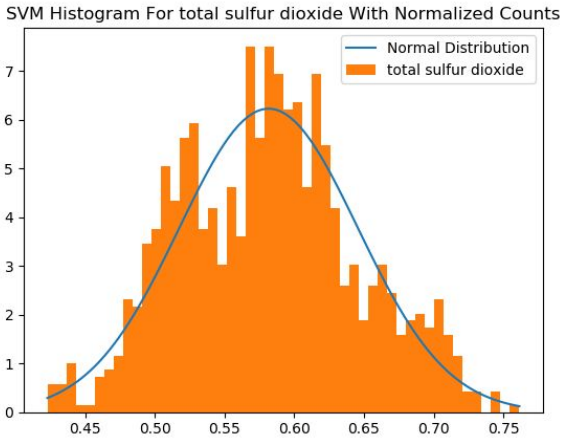


Figure 46

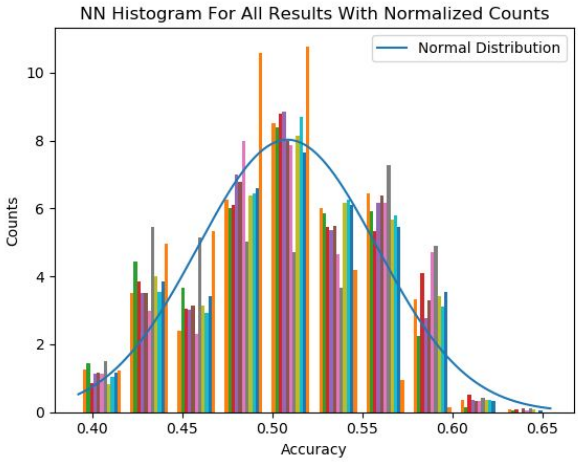


Figure 47

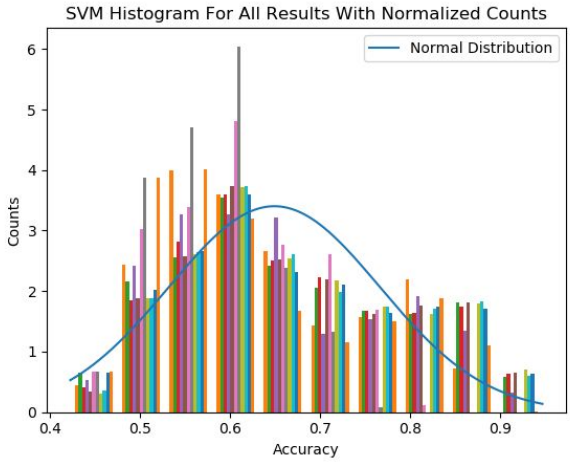


Figure 48

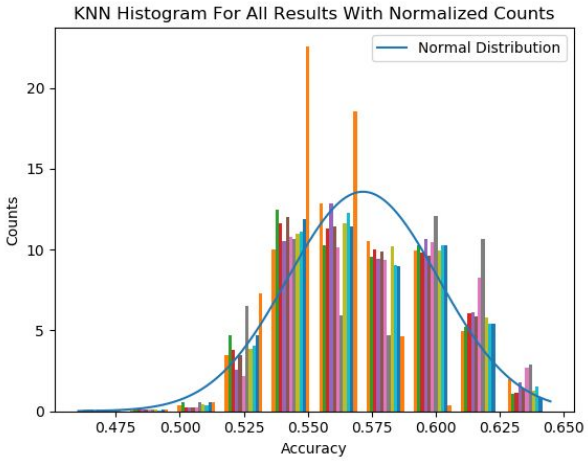


Figure 49

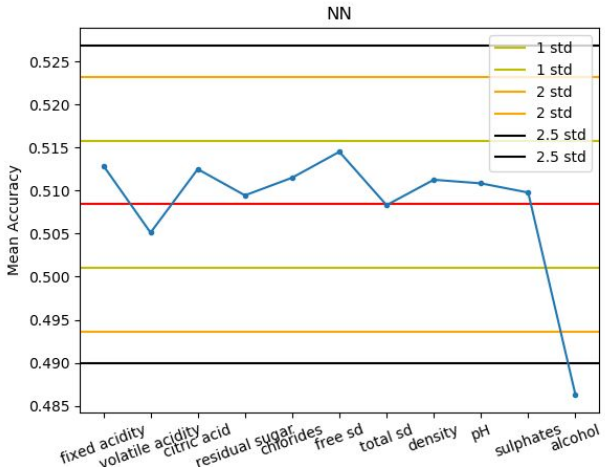


Figure 50

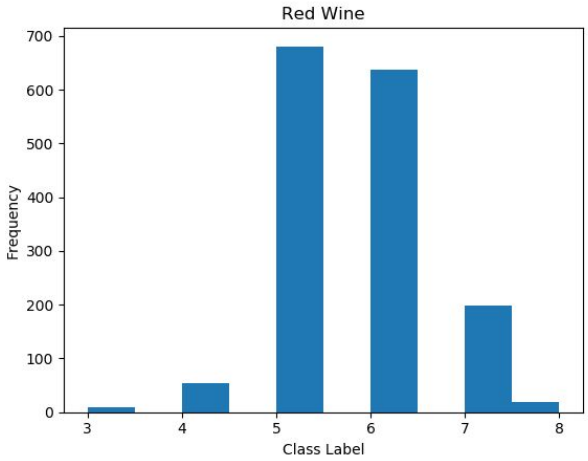


Figure 53

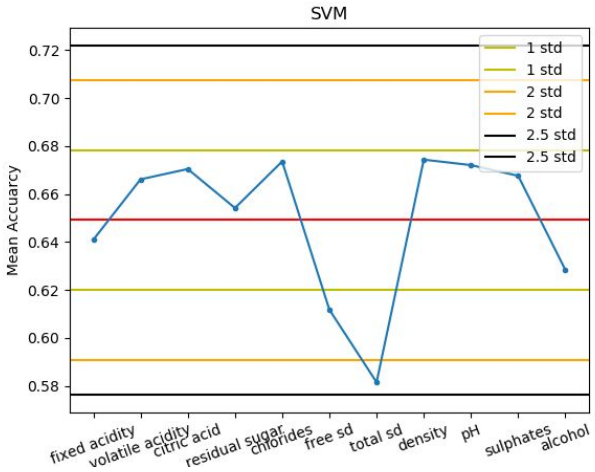


Figure 51

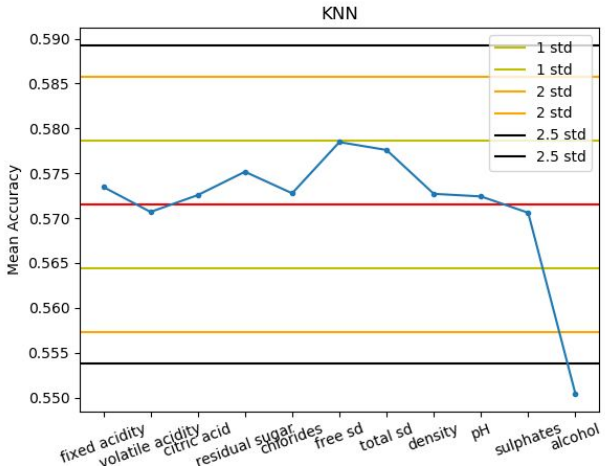


Figure 52