**Comparing Different Classification Algorithms**
**By: Ali Khan, Edgar Fonseca, Franco Gonzales**

**Introduction** **1**

**Data Sets** **1**

Fatal Police Shootings 2

Spambase Data Set 2

UTKFace Data Set 2

**Algorithms** **3**

K-Nearest Neighbor 3

Support Vector Machine (SVM) 4

Gaussian Process Classifier 4

Linear Discriminant Analysis 5

Decision Tree 5

Gradient Descent 6

Naives Bayes 7

Convolutional Neural Network 7

**Discussion and Conclusion** **8**

## Introduction

Our group had decided to do our project on comparing the different variations of classification algorithms. The whole point of classification is to be given specific features of a data set, and be able to classify it based off of those features. There are many different types of these algorithms, with varying run time and accuracy. Accuracy is not going to be homogenous throughout all classifiers. Specific algorithms perform better or worse relative to the type of features inputted. In this report, we'll be feeding different algorithms various types of data. Categorical, continuous, binary, structured, and unstructured. Then, we'll be analyzing to see where each algorithm excels.

## Data Sets

With the main goal of this project is to compare various algorithms. Therefore, we'd like to have a diverse set of data for them to run through. There's not point in having 5 sets of all numerical data. We'd like to have structured and unstructured sets, continuous and discrete, binary vs non binary, etc. Below are the datasets used, as well as a brief description of each.

- Mushroom Classification
  - Source: UCI Machine Learning
  - Link: https://www.kaggle.com/uciml/mushroom-classification

This data contains features which correspond to mushroom coding. The classes used to describe the mushrooms are edible or poisonous. The corresponding features are unique in that they are categories themselves, with no numeric values. We'd have to perform some way of label encoding to be able to throw this data set into our algorithms.

- Red Wine Quality
  - Source: UCI Machine Learning
  - Link: https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009

This data set is a complete contrast to the mushroom data above. It contains 11 features which correspond to one class, the quality of the wine. The quality of the wine is on a scale from 1-10, with all of the features also following a numeric scale. There little to no preprocessing required here, as the algorithms can take in these features no problem.

- Fatal Police Shootings
  - Source: The Washington Post

- Link: https://www.kaggle.com/washingtonpost/police-shootings

This dataset is a compilation of every fatal shooting in the U.S. by a police officer since 2015. Included is a lot of information about the scene: race, gender, was the victim armed, were they fleeing, their name, was the officer wearing a body camera, etc. For this data set, we'll be using victim race as the class.

- Spambase Data Set
- Source: UCI Machine Learning
- Link: https://archive.ics.uci.edu/ml/datasets/Spambase

This is a collection of emails both considered spam and non-spam. This was aggregated in order to create a spam filter. The features are unique in that contain continuous variables that summarize the emails.

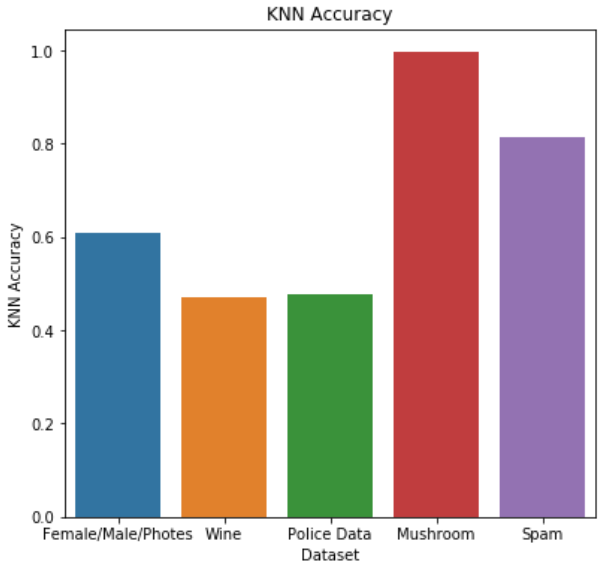
- UTKFace Data Set
- Source: GitHub
- Link: https://susanqq.github.io/UTKFace/

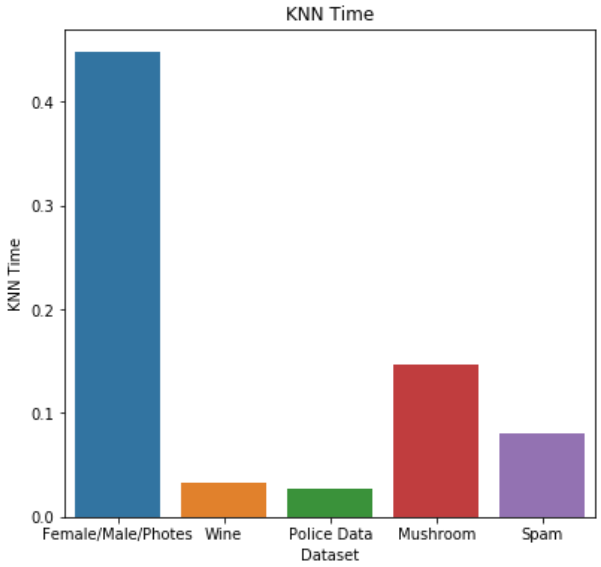
This dataset consists of 20000+ images of various people. The dataset contains continuous variables that specifies the person's age, race, gender, as well as the date & time. We decided to use a small sample of 3000 images with our end goal to classify the images by gender.

## Algorithms

Listed below are our algorithms of choice as well as a brief description of the math behind the methods. The run-times were calculated using Python's built in time library. The sci-kit learn provided the majority of the algorithms except for the convolutional neural network (CNN). The CNN was built using TensorFlow.

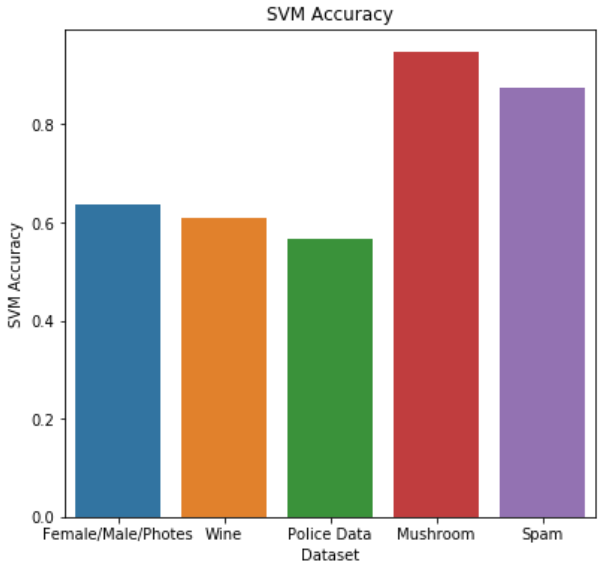
- K-Nearest Neighbor

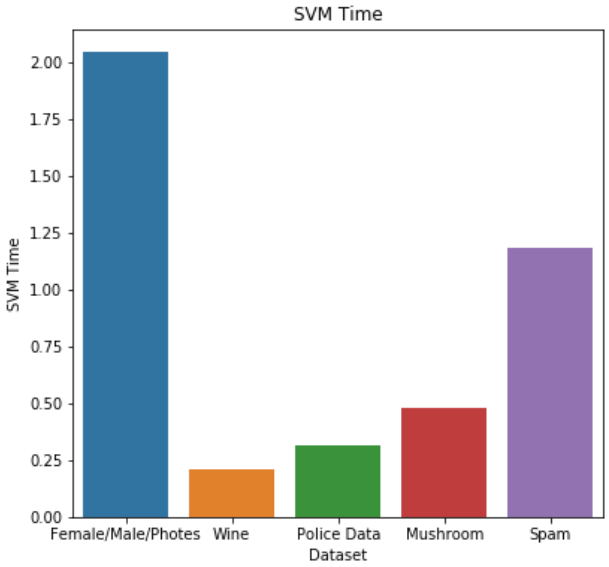



KNN is a supervised learning algorithm that classifies new observations depending on the classification of its K nearest neighbors. When using KNN on five of our different datasets we found that the speed was quite quick, but it did handle continuous datasets quicker than categorical datasets. For example our two datasets that contained the most/if not all categorical data had vastly longer times than the continuous data. But we can see that the accuracy for the mushroom and spam dataset had higher accuracy than the other datasets. This may be due to the fact that these two datasets were created for classifying purposes.

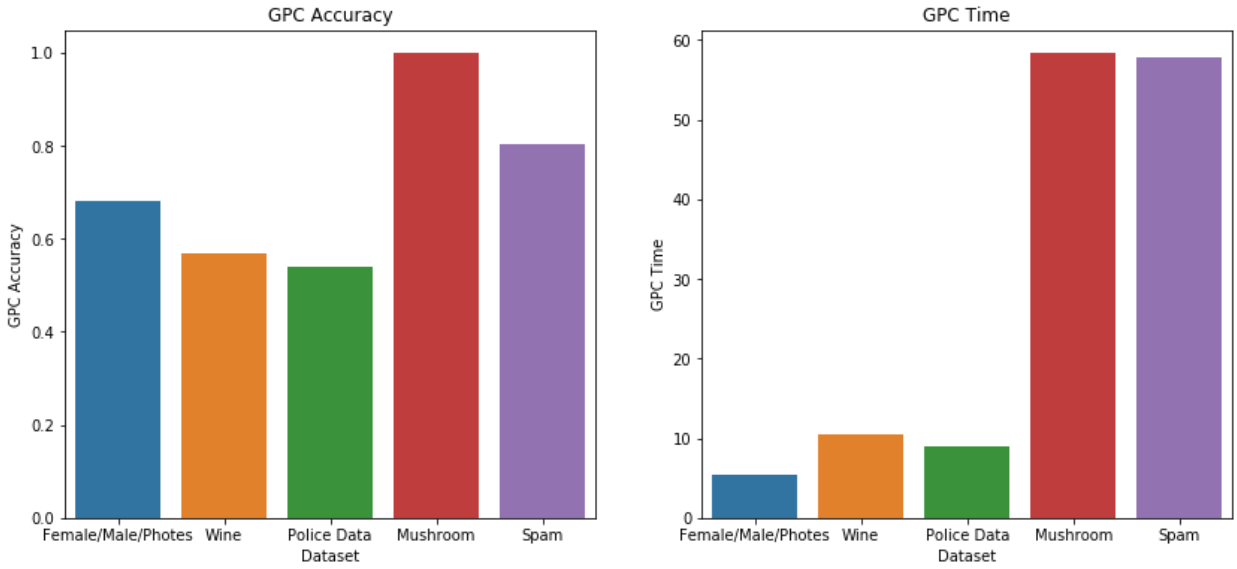
- Support Vector Machine (SVM)



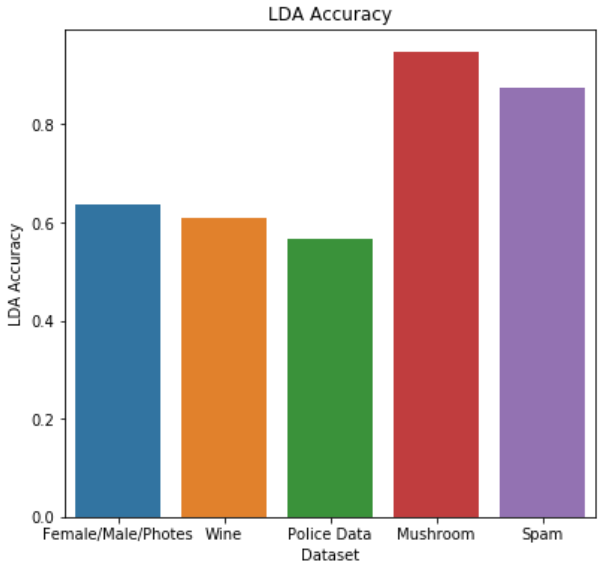

This supervised learning algorithm wishes to create a hyperplane between classes. We can see that performance in accuracy was better for the categorical datasets and relatively even for the continuous datasets. Looking at the time plot we can see that compared to KNN algorithm the times were similar except for spam dataset.

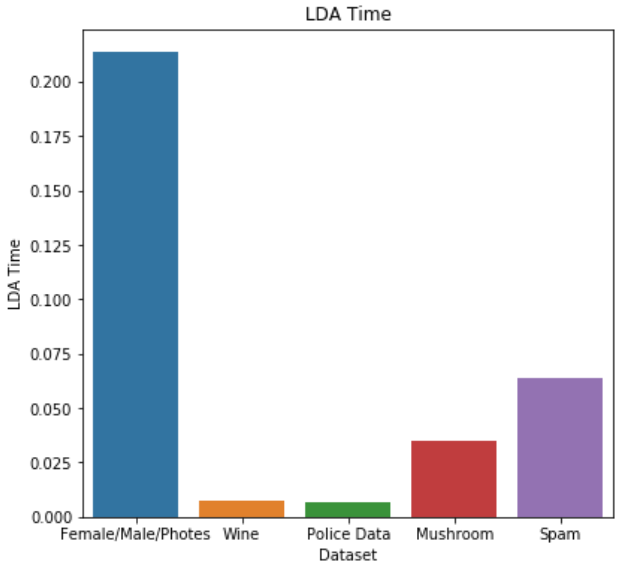
- Gaussian Process Classifier




Gaussian processes are iterative non-parametric method that uses covariances between features to make inferences on the data. The algorithm is then able to make place specific classes given a feature-space. One thing to notice here for GPC is the differences in time compared to the other algorithms. While accuracy is more or less the same, you end-up having to use more computational power in order to do so. This is due to the fact that GPC is based off of iterations. Sk-learn by default has a maximum number of 100 iterations for this model. Using GPC may or may not be worth it depending on our data points, as the accuracy even compared to other models are not too far from each other. Surprisingly, one benefit that came from GPC was that comparing the runtime of the datasets it had handled the images dataset faster than the other algorithms.

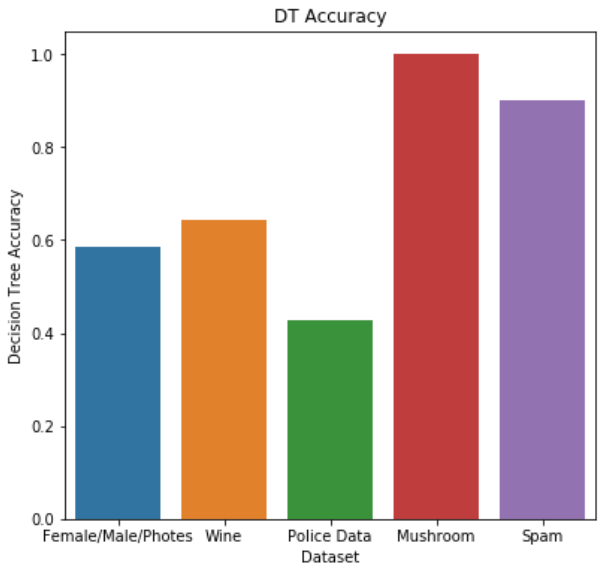
- Linear Discriminant Analysis

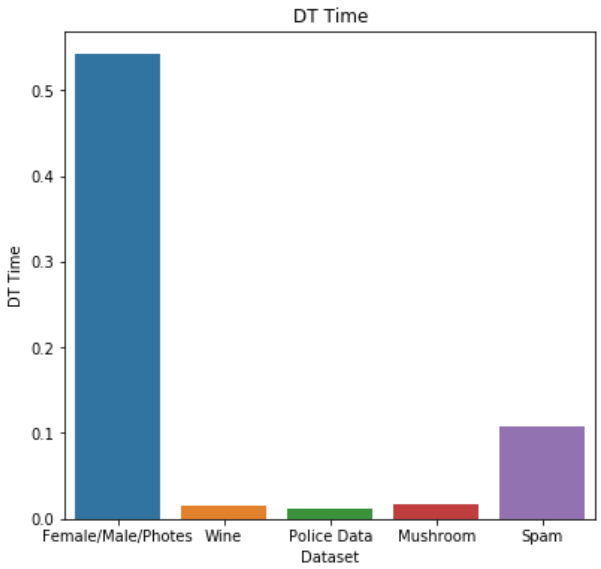



Also a bayesian based methods, LDA assumes that classes are normally distributed and contain a class-specific means and class-specific variances. LDA is a dimension reduction based algorithm, using a discriminant to represent the feature-space into a projected lower dimensional space. The results here are nothing too surprising. The run-time for LDA is not too bad, as well as accuracies that are similar to the other algorithms.

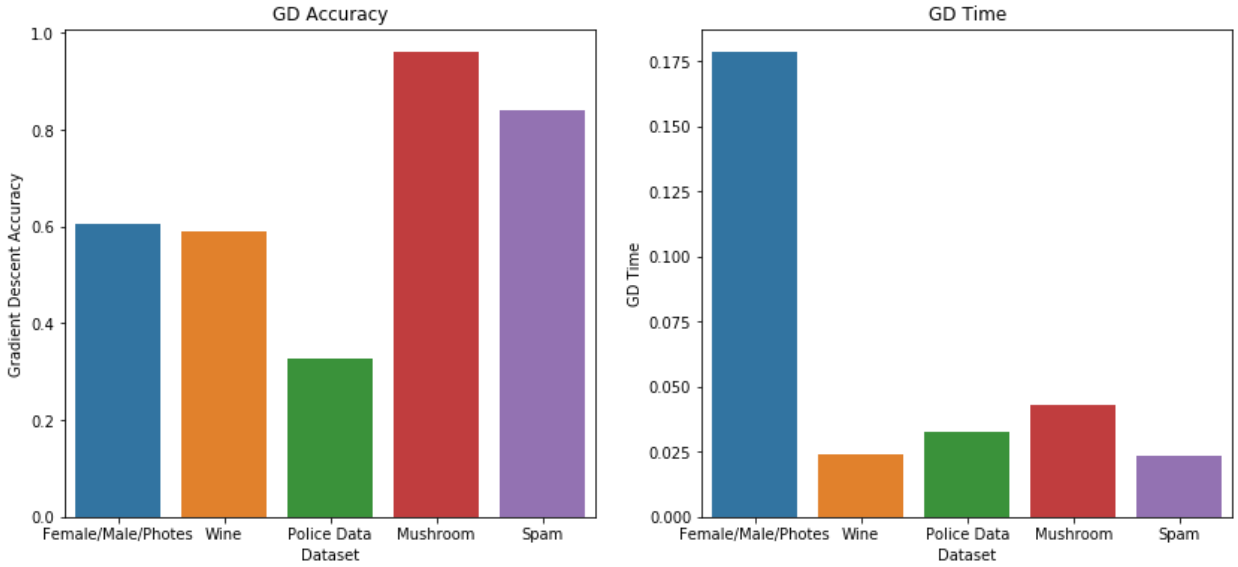
- Decision Tree



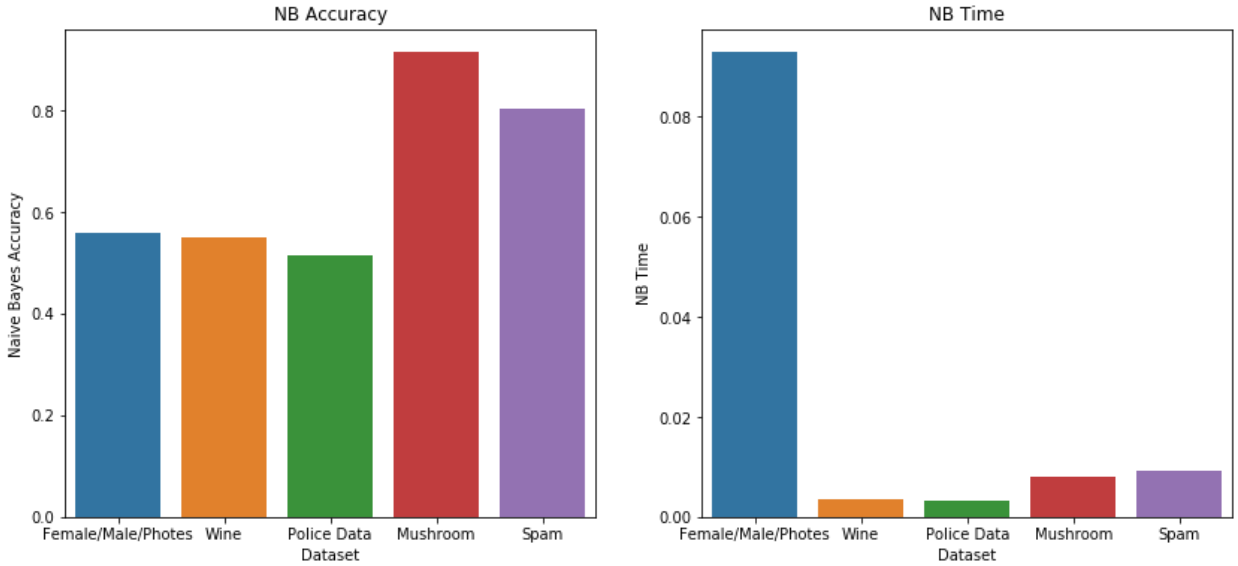

The decision tree algorithm decides what features to use and what conditions to use for splitting. Based on these conditions the algorithm classifies new observations. We can see that the times for the Decision tree was tremendously faster for the wine, police, and mushroom dataset.

- Gradient Descent



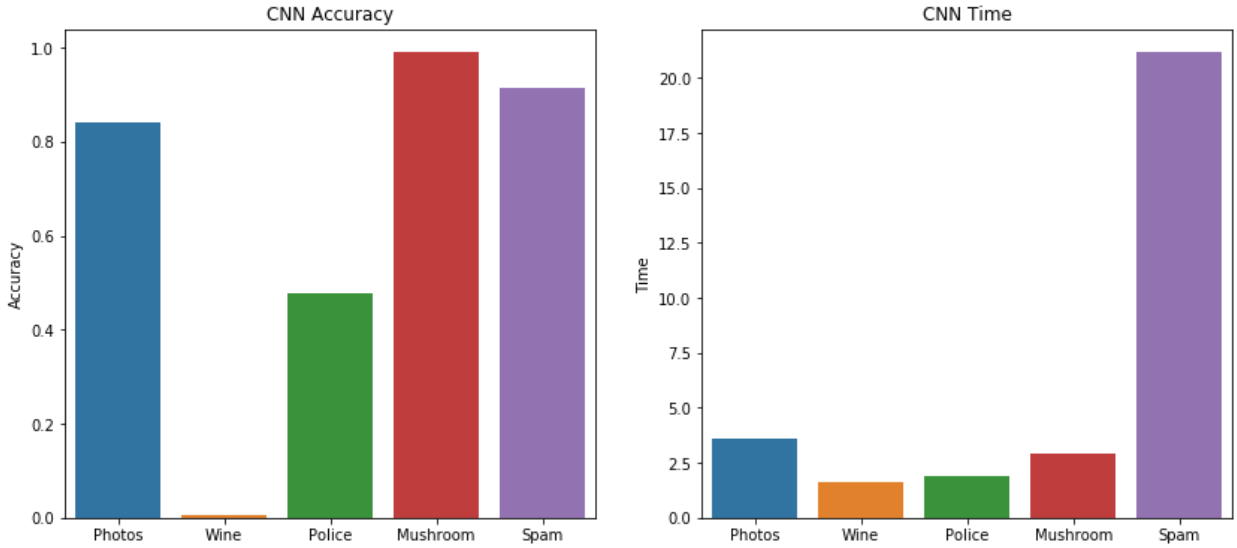

Gradient descent is an optimization algorithm that uses iterative methods until a minima is reached. It uses "small steps" per iteration until the desired point is reached. As we can see here, the run times are nothing ridiculous. In addition, there's not much of a difference in terms of accuracy compared to the other algorithms above.

- Naives Bayes




Naive Bayes algorithm uses Bayes theorem to compute the posterior probability of each class, which can then classify a new observation based on the highest posterior probability. We can see that the times using this algorithm are relatively similar between wine, police, mushroom, and spam dataset but slow for images dataset. The accuracy seems to look the same as the other algorithms.

- Convolutional Neural Network




CNNs are specifically used mainly for image recognition in machine learning. Inputs are put through a series of neurons which all lead to a specific output; in this case, the classes. Due to how the data sets were set-up the algorithm for the photos had 10 training iterations, while the rest had 50. Note, that the spambase data took so long to iterate over, that we had to take the log values of the time to be able to plot it. What's important to note here is that the gender recognition is around 10-20% more accurate compared to the other algorithms. In addition, the wine accuracy got completely thrown out the window, with it being less than 1%. This was the only real dataset that was all numerical. The run times are significantly longer than the other algorithms, especially for the spambase. Obviously, neural networks are more complicated than the other algorithms. As such, it's to be expected.

## Discussion and Conclusion

The run times for the algorithms slowly increased as the complexity and depth of the algorithm increased. To measure "depth" we pretty much summed it up as how much more googling we have to do in order to be able to understand what the algorithm does. For example, KNN by nature is very simple to understand. Therefore, the run-time is what we'd expect, quick. Iterative methods such as Gaussian Processes or Neural Networks also took up the most run-time. There is one thing to note. As run time does increase, there isn't necessarily a large increase in accuracy as well. Look at the differences in accuracies between K-nearest-neighbors and GPCs, there really isn't much, if any.

The neural network was the most interesting out of all of our algorithms. For some reason, the wine data was almost incompatible with the neural network. Just taking a look at the type of features provided by the wine quality, they are all continuous numerical values. The classes range on a scale of 1-10, and this is where the issue might lie. All of the other data sets have a low amount of classes.

So now let's see what algorithms are the best for each dataset.

- Face Dataset

Easily, CNN was the most accurate out of the bunch. Unfortunately, it took the most power out of all to algorithms. We used Google Colaboratory, and running it without a GPU would kill all the RAM in the session. This is the tradeoff between accuracy, runtime, and power that you have to consider. All of the other algorithms had around a 60% recognition rate, while the CNN was around 80%.

- Wine Quality

With this dataset containing pretty much all numerical features, the decision tree had the highest accuracy score out of all the other algorithms. In addition, the decision tree was extremely fast, being less than half a second. The neural network is not recommended for this dataset. Obviously there are different types of neural networks, but that specific one we built should not be used for the wine data.

- Poisonous or Edible Mushrooms

This classification dataset had no problem getting nearly 1.0 accuracies in all of the algorithms. Therefore for a dataset that contains all categorical variables we focused on run time. We found that the decision tree algorithm handled the dataset with ease with a runtime of a fraction of second.

- Police Data

The accuracy scores for this dataset were very similar throughout each algorithm, except for Gradient descent. Unlike the mushroom data, this dataset wasn't made for classification so it had a more realistic accuracy, but LDA and Decision Tree were still able to handle it at really fast speeds.

- Spambase Data

The highest scores for this dataset was the CNN. With the neural network being the highest score, it also ran the longest. The data consisted of integers from 0 to 100 which had specific descriptions of the emails. Compared to the other algorithms, the 2nd highest was an even split between LDA and SVM. the differences in accuracy was around 4% for both, but the neural network took a significant amount of time relative to the other two. A neural network may or may not be  be overkill for a spam filter.

Overall, we thought that this project was very interesting in terms of the results. Accuracy and run-time had no real relationship as shown by our plots above. GPC took an astronomical amount of time compared to some of the other algorithms, but the gains in accuracy were minimal or insignificant.  As a result, we

believe that there is no end-all-be-all classification algorithm that will fit every dataset perfectly. We'd recommend to try multiple algorithms and go from there.