Comparison of Banknote Classification Methods

Evan Gomolin
Faculty of Science
Western University
London, Canada
egomolin@uwo.ca

David MacLean
Faculty of Science
Western University
London, Canada
dmacle22@uwo.ca

Matvey Skripchenko
Faculty of Science
Western University
London, Canada
mskripch@uwo.ca

Abstract— **Banknote classification has been a problem every since people started creating counterfeit money. In the past couple decades, the rise of machine learning has lead to this problem being solved relatively easily. Using an easier problem, with a dataset involving complete knowledge of the problem, we decided to test and compare many of the classification models provided by the sklearn Python library. After eliminating most of the bias from test set selection, with cross validation, and feature selection, with many feature sets, training and testing revealed that Logistic Regression, Stochastic Gradient Descent, k-Nearest Neighbours, Decision Tree Classifier, Random Forest Classifier, Linear Support Vector Classifier, Radial Basis Function Classifier, and Multi-Layer Perceptron Classifier models performed quite well, with negligible difference in accuracy. Gaussian Naive Bayes significantly under-performed. However, the k-Nearest Neighbours model is experimentally the best sklearn model for banknote classification. It performed well on sub-optimal feature sets and trained in the shortest amount of time.**

Keywords— sklearn, banknote authentication, model comparison, kNN, machine learning

I. Introduction

Due to the recent growth in the usage of electronic financial transaction and the resulting decrease in the use of currency, real money transactions are still playing a very important role in the current global market [1]. Given the ongoing importance of using banknotes as a payment mechanism, a loss of confidence towards banknote utilization could have a substantial impact on the communities worldwide [2].

In the past, banks and businesses would visually analyze banknotes to test their legitimateness by using UV-light to search for certain patterns that did not align with the standards of the official banknote. However, with a rise of technology and improvement in counterfeiting techniques, various banknote processing machines were developed to drastically enhance the banknote recognition and guarantee consistency in security. The general banknote recognition process consists of the following stages: image preprocessing and segmentation, best-suited features extraction, classification of the extracted features and final verification [3].

For this project, we concentrated our attention on applying various sklearn library classification models to the already extracted data features from numerous legal and counterfeit banknote images. After performing model training and testing, we proposed the best and worst models based on their accuracy and efficiency.

II. Related Work

Counterfeit detection problem is branch of a banknote classification field that also deals with serial number and fitness classification. For example, the majority of Automated Teller Machines (ATM's) are equipped with fitness classification technology that is used to assess the banknote's physical condition and soiling level by applying visible light and Near Infrared (NIR) image information [1]. This classification type involves methods such as nonlinear Support Vector Machines (SVM) and Nearest Neighbor (NN) classifiers [4].

Over the years, identifying between real and forged currency has become more challenging due to enhancements in color printing, imaging, cloning and counterfeit notes being produced using best available technology that employs security paper [5]. As a result, various image analysis machine learning methods were developed to cope with the issue. There is a number of studies in a medical field where image classification methods are used to improve medical analysis for identifying the key findings in order to assign a proper treatment to the patient [6].

Banknote classification has developed tremendously, but not without facing challenges. Most of the refined models that were built to only support a limited number of banknote currencies. As the number of classees to be classified increased with the number of currencies, the classification efficiency decreased. The problem was resolved by applying Convolutional Neural Network (CNN) for classification that required a time consuming training to guarantee high performance classification [7].

III. Methods

Our purpose with this experiment is to compare the accuracy of different classification models in the sklearn library when classifying banknote images as authentic or counterfeit. Our methodology behind this experiment goes as follows:

1. Dataset analysis.
2. Preprocessing of the dataset.
3. Training and testing the models on the preproceessed data using K-Fold cross-validation.

4. Comparing the models trained on best and worst feature set.
5. Dissecting and analyzing the changes in accuracy of every model over various polynomial degrees and finilizing the results.

A. Data

Ideally for this experiment, we wanted a dataset which had a sufficiently large number of samples, images of decent quality, and descriptive metadata which would provide a fair foundation for the comparison of models. The dataset selected for this experiment was sourced from the University of California Irvine's Machine Learning Repository. This "banknote authentication dataset" was initially extracted from images of authentic and counterfeit banknotes. Each image in the dataset have dimensions that were preset to be 400 by 400 pixels with an approximate resolution of 660 dpi [8]. The features from the images were retrieved using the Wavelet Transform tool. The feature dictionary is presented in Table I.

TABLE I. DATA DICTIONARY

Colum n	Feature Name	Definition	Data Type
1	variance of image	pixel spread	continuous
2	skewness of image	image assymetry	continuous
3	curtosis of image	peakness/ flatness	continuous
4	entropy of image	pixel intensity	continuous
5	class	1-legal, 0-forged	integer

The metadata file, *data_banknote_authentication.txt,* included in the dataset contains the above features and has the dimensionality of 5 columns representing the features and 1372 rows representing the instances. The distribution of features 1-4 is shown in Fig. 1. From these distributions we can see that the variance and skeweness features appear to have a much higher standard deviation than the other features. This wider spread suggests that there may be more potential for accurate classification with the skeweness and variance features.

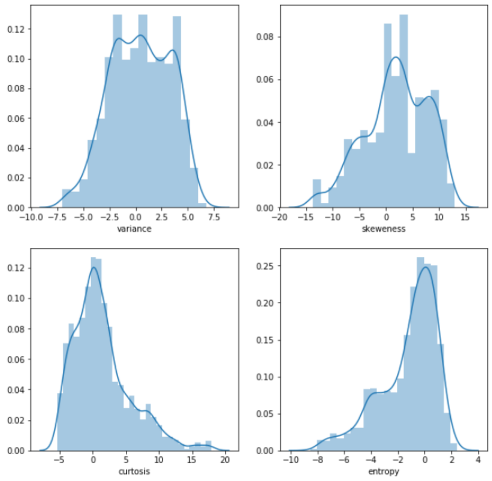


Fig. 1. Distribution of features 1-4 from the banknote authentication dataset. Each vertical axis measures the percent of samples. Each horizontal axis represents the values of each of the features.

Furthermore, we calculated the correlation between each of the features and produced the heat map in Fig. 2. From this heat map, we can see that except for skeweness and curtosis, the correlation of our features is quite low. This result inspires confidence in our feature set; each feature's independance shows that it provides relevant information to the classification. Due to the already low dimension of our feature space, it would not have been ideal to have to remove one of our features due to high interdependance.

As displayed in Fig. 3, we analyzed each of the features after splitting the samples by their labeled class. Notice across the main diagonal that variance and skeweness show more distinction between the two classes than curtosis or entropy. This is in accordance with our previous statement that their wider spread suggested more potential for separation. Additionally, we can see that higher correlation feature-pairs such as curtosis-entropy appear to be more dependant than a feature-pair with low correlation, like variance-skeweness. Therefore, we can expect that variance and skeweness to be more effective than entropy and curtosis.

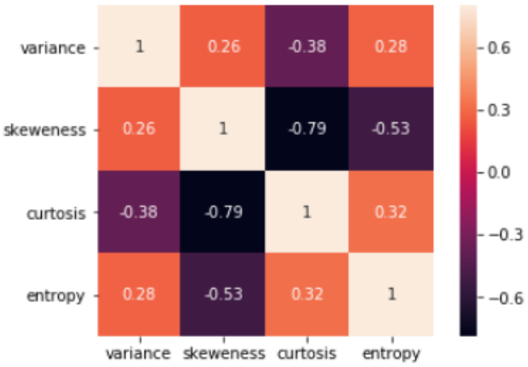


Fig. 2. Heat map of the correlation between each of the features in the bank note authentication dataset.

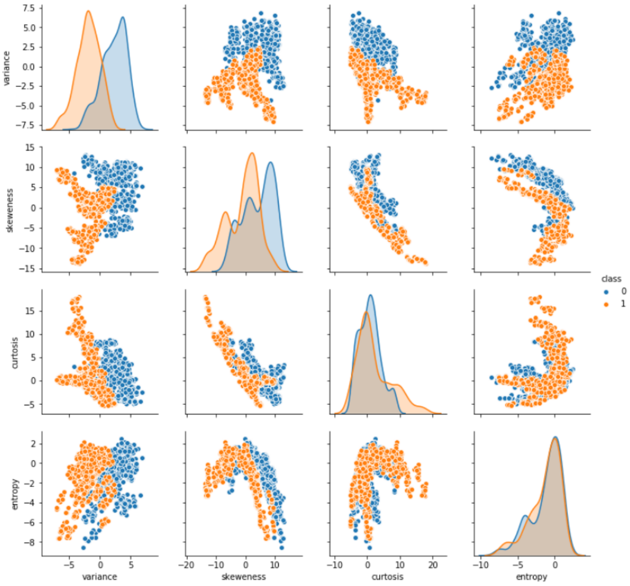


Fig. 3. Pairplot graph of all four features in the banknote authentication dataset. The samples in blue are the legal banknotes and the ones in orange are the illegal forgeries.

For this experiment, we are only using the variance, skeweness, curtosis and entropy features. Given our computational resources and the quality of the features extracted by the Wavelet Transform tool, it did not seem necessary to increase the computational load.

B. Data Preprocessing

In advance of training and comparing the various models, we developed a preprocessing stage to transform the banknote authentication dataset into a consistent, usable format. The goal with our preprocessing stage was to prepare the metadata in a generic way, such that the preprocessed metadata was equally useful to all the models. In other words, we wanted to provide our various models with many feature sets, so that the bias introduced by a selectively favourable dataset could be limited.

First, we constructed all possible subsets of the feature set, {curtosis, entropy, skeweness, variance}, with a size greater than or equal to 2. This allows each different model to select the feature set that produces the best results. Each model operates on its optimal feature set, and there is no bias from limiting the feature set to one option across models.

Our next consideration was splitting the data into training and test sets. However, we were concerned that the selection of specific training and test sets could introduce a bias towards certain models. As a result, we implemented K-Fold cross validation. Using sklearn, we split the dataset into k bins, so that when we trained the models, we could average the results of each bin taking a turn as the test set. Although bias remains in the selection of k, and how the bins are selected, by allowing every sample to contribute to both the training and test sets, we drastically decreased the chances of bias. Generally, the recommended k value is around 5 to 10 [9]. We selected k=5 for our experiment because our dataset of 1372 samples is quite small, and even though higher k values would eliminate more potential for bias, we did not want to run into over-fitting issues.

C. Model Training

For this experiment we selected 9 models from the sklearn library to compare. When selecting the models our goal was to pick classifiers that were relatively similar, avoiding deep neural networks, to provide a more interesting comparison. Furthermore, we selected models that were easily implementable with sklearn as these are the most practical in most industry. The models we chose and how they were configured are as follows:

1) Gaussian Naive Bayes (NB)
    - We configured the default sklearn function GaussianNB with a variance smoothing of 1e-09. [10]
2) Logistic Regression (LR)
    - We used the default LogisticRegression sklearn function with the "saga" solver . As training time was a factor in our comparison we selected the solver which had "fast convergence" [10].
3) Stochastic Gradient Descent Classifier (SGD)
    - We implemented the default sklearn function SGDClassifier with default parameters [10].
4) K-Nearest Neighbours Classifier (kNN)
    - We used the default sklearn function KNeighborsClassifier with k = 5. We selected a small k value because our dataset was relatively small, and experimentally it was the optimal parameter [10].
5) Decision Tree Classifier (DT)
    - We configured the default sklearn function DecisionTreeClassifier with the max depth parameter set to 10. Due to our computation resources and optimizing for both performance and training speed, limiting the height of the tree to 10 proved experimentally to be the best option [10].
6) Random Forest Classifier (RF)
    - We used the default sklearn function RandomForestClassifier with the n_estimators parameter set to 25. We decided to limit the number of trees in the forest from the default value of 100 due to limits with our computational resources [10].
7) Linear Support Vector Classifier (linearSVC)
    - We implemented the default sklearn function SVC with the kernel parameter set to "linear" [10].
8) Radial Basis Function Support Vector Classifier (rbfSVC)
    - We configured the default sklearn function SVC with the kernel parameter set to "rbf" [10].
9) Multi-Layer Perceptron Classifier (MLP)
    - We used the default sklearn function MLPClassifier with the "adam" solver, as according to the sklearn documentation, it performs best on datasets over 1000 samples [10]. Furthermore, we set the maximum number of iterations to 1000, because experimantally this was produced optimal results.

In Fig 4. we show the training times for each model. The units are a relative measure, not absolute. They are computed from the number of CPU counter increments that occur. This unit of measurement does maintain the ratio of time taken relative too all the other models.

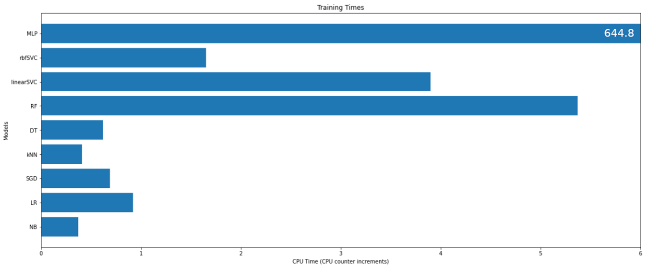


Fig. 4. Training times of each model. The time values are based on the CPU counter and should only be interpreted relative to each other.

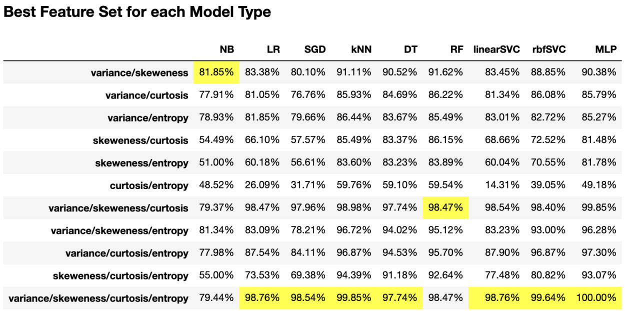
D. Model Testing

As previously mentioned, we preprocessed our data such that 5-fold cross validation could be used during testing. Given data split into 5 bins, for each of our models we trained and tested the models in 5 different iterations – each one having one of the bins as the test set and the rest of the bins as the training set. Then we summarized these 5 accuracy results into both the mean and standard deviation of the 5 iterations. We then did this for all possible feature sets. At the end of this, we had recorded the accuracy of each model, for each possible feature set, with limited selection bias from the test set and feature set chosen.

IV. Experimental Results

After training and testing all of our models on all of the banknote authentication dataset, we then collected the results and analyzed them to draw comparisons between the models.

First, we looked at which feature sets produced the best results for each model. As it shows in Fig. 5, most of the models benefit from having as many features as possible. However, both the Gaussian Naive Bayes, and Random Forest Classifier actually performed better with fewer features. Moreover, the correlation we discussed in Fig. 2 is perfectly on display here. We can see that adding entropy to the feature set "variance/skewness/curtosis" causes very little improvement in accuracy. It can also be seen that if a single feature set such as "curtosis/entropy" was selected, certain models would have an advantage over others; e.g. k-Nearest Neighbours would uncharacteristically outperform Linear SVC. Clearly, the best performing model is MLP with a perfect score; however, most of the model perform similarly given their optimal feature set.



Best Feature Set for each Model Type

	NB	LR	SGD	kNN	DT	RF	linearSVC	rbfSVC	MLP
variance/skeweness	81.85%	83.38%	80.10%	91.11%	90.52%	91.62%	83.45%	88.85%	90.38%
variance/curtosis	77.91%	81.05%	76.76%	85.93%	84.69%	86.22%	81.34%	86.08%	85.79%
variance/entropy	78.93%	81.85%	79.66%	86.44%	83.67%	85.49%	83.01%	82.72%	85.27%
skeweness/curtosis	54.49%	66.10%	57.57%	85.49%	83.37%	86.15%	68.66%	72.52%	81.48%
skeweness/entropy	51.00%	60.18%	56.61%	83.60%	83.23%	83.89%	60.04%	70.55%	81.78%
curtosis/entropy	48.52%	26.09%	31.71%	59.76%	59.10%	59.54%	14.31%	39.05%	49.18%
variance/skeweness/curtosis	79.37%	98.47%	97.96%	98.98%	97.74%	98.47%	98.54%	98.40%	99.85%
variance/skeweness/entropy	81.34%	83.09%	78.21%	96.72%	94.02%	95.12%	83.23%	93.00%	96.28%
variance/curtosis/entropy	77.98%	87.54%	84.11%	96.87%	94.53%	95.70%	87.90%	96.87%	97.30%
skeweness/curtosis/entropy	55.00%	73.53%	69.38%	94.39%	91.18%	92.64%	77.48%	80.82%	93.07%
variance/skeweness/curtosis/entropy	79.44%	98.76%	98.54%	99.85%	97.74%	98.47%	98.76%	99.64%	100.00%

Fig. 5. Table of the accuracy of each model on a given feature set. The cells highlighted in yellow indicate the max value for that column (model).

In a similar table, Fig. 6 shows which models best classify the banknotes given a specific feature set. Interestingly, the Random Forest Classifier performs is the best performing model in lower dimension feature spaces. In the features spaces with dimension greater than 2, the k-NN classifier and MLP classifier appear to perform equally well.

It is also important to consider the best and worst cases of these models. In a production environment, you may not be able to always recover the optimal feature set from the data. In Fig. 7 we detail the best and worst cases of feature selection for each model. Notice that Logistic Regression, Stochastic Gradient Descent, Linear SVC, rbfSVC and MLP all have extremely well performing best cases. However, without the optimal feature set they all perform worse than random chance in the worst case. Whereas the k-Nearest Neighbours, Decision Tree, and Random Forest classifiers have acceptably high performance on the optimal feature set, and a worst case that at least outperforms a coin flip.

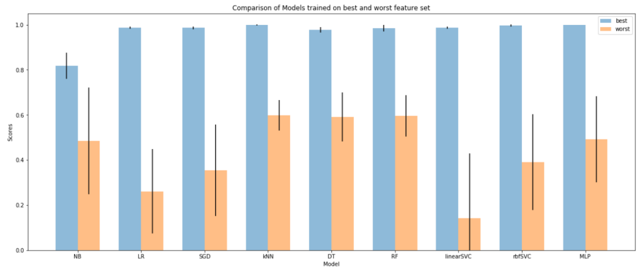
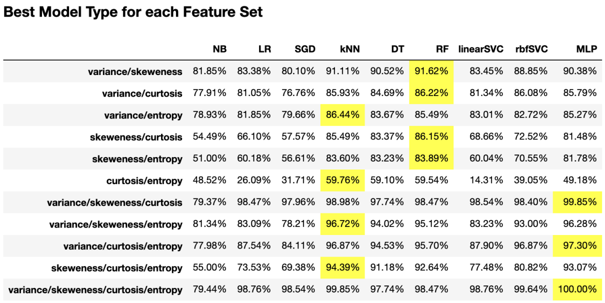


Fig. 6. Table of accuracy of each model on a given feature set. The cells highlighted in yellow indicate the max value for that row (feature set).



Best Model Type for each Feature Set

	NB	LR	SGD	kNN	DT	RF	linearSVC	rbfSVC	MLP
variance/skeweness	81.85%	83.38%	80.10%	91.11%	90.52%	91.62%	83.45%	88.85%	90.38%
variance/curtosis	77.91%	81.05%	76.76%	85.93%	84.69%	86.22%	81.34%	86.08%	85.79%
variance/entropy	78.93%	81.85%	79.66%	86.44%	83.67%	85.49%	83.01%	82.72%	85.27%
skeweness/curtosis	54.49%	66.10%	57.57%	85.49%	83.37%	86.15%	68.66%	72.52%	81.48%
skeweness/entropy	51.00%	60.18%	56.61%	83.60%	83.23%	83.89%	60.04%	70.55%	81.78%
curtosis/entropy	48.52%	26.09%	31.71%	59.76%	59.10%	59.54%	14.31%	39.05%	49.18%
variance/skeweness/curtosis	79.37%	98.47%	97.96%	98.98%	97.74%	98.47%	98.54%	98.40%	99.85%
variance/skeweness/entropy	81.34%	83.09%	78.21%	96.72%	94.02%	95.12%	83.23%	93.00%	96.28%
variance/curtosis/entropy	77.98%	87.54%	84.11%	96.87%	94.53%	95.70%	87.90%	96.87%	97.30%
skeweness/curtosis/entropy	55.00%	73.53%	69.38%	94.39%	91.18%	92.64%	77.48%	80.82%	93.07%
variance/skeweness/curtosis/entropy	79.44%	98.76%	98.54%	99.85%	97.74%	98.47%	98.76%	99.64%	100.00%

Fig. 7. Comparison of each model's best and worst performance on feature sets.

Lastly, note that the Gaussian Naive Bayes has significantly worse results than all other models. This is evident in Fig. 8 where, across all optimal feature sets for each model, Gaussian Naive Bayes' confusion matrix shows significantly more false positives and false negatives than any other model. Using the (1) and (2) we can produce the Fig. 9, showing the recall and precision scores of each model.

$$\text{RecallScore} = \text{Tp} \; / \; (\text{Tp} + \text{Fn}) \tag{1}$$

$$\text{PrecisionScore} = \text{Tp} \; / \; (\text{Tp} + \text{Fp}) \tag{2}$$

Where Tp is the number of true positive predictions, Tn is the number of true negative predictions, and Fp is the number of false positive predictions. Due to Gaussian Naive Bayes' relatively poor precision score, it is at greater risk of identifying counterfeit banknotes as legal tender than the other models. Additionally, its low recall score makes Gaussian Naive Bayes unreliable for the task of at least finding all of the legal banknotes.

Finally, we tested how generating polynomial combinations of our features would have on the accuracy of the models. To do this we generated every possible polynomial from degree 0 to degree 5 using only our original features. For example, for a given 2-dimensional feature [x, y], this method at degree 2 would produce the set $\{1, x, y, x^2, xy, y^2\}$. As shown in Fig. 10, none of the models benefited from any significant improvement. We were attempting to improve on either the Gaussian Naive Bayes or the worst-case feature set accuracy, but no such improvements were to be found.

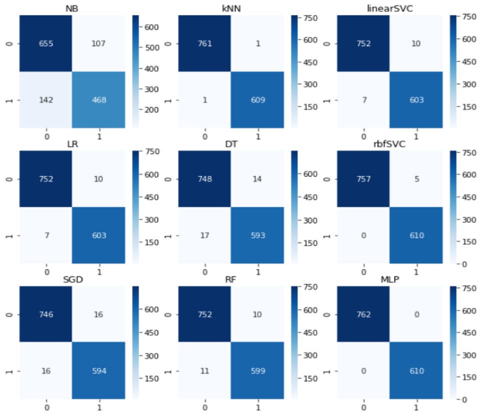


Fig. 8. Confusion matrices of each model's performance on their best feature set. 0 is the class of forged banknotes and 1 is the class of legal banknotes.

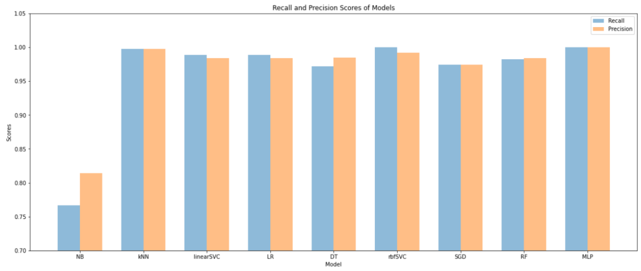


Fig. 9. Recall and Precision scores for each model, trained on their optimal feature set.

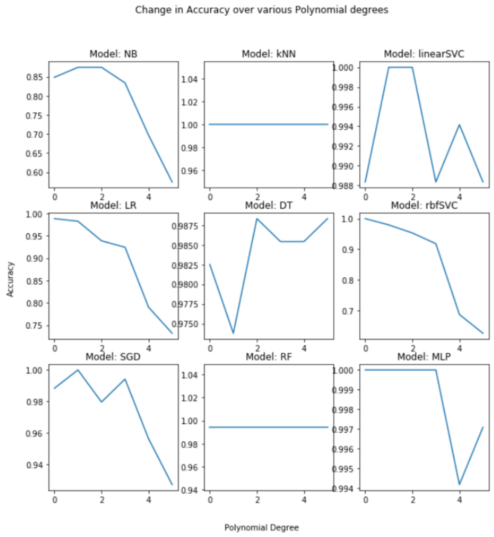


Fig. 10. Accuracy of each model as the degree of the polynomial features was increased from 0 to 5.

V. Conclusions

When choosing which type of machine learning technique to implement for banknote authentication, the decision is highly dependent on the dataset and feature set. Among all of the tested models, only Gaussian Naive Bayes significantly under-performed in terms of accuracy. Selecting between the rest of the models depends mostly on the situation. The Multi-Layer Perceptron classifier marginally performed the best with a perfect feature set, but was not as successful with sub-optimal feature sets. Whereas the k-Nearest Neighbours model provided relatively high accuracy across all sub-optimal feature sets. In reality, the difference between all of the non-Naive Bayes models is negligible; factors outside of model success would be more influential to a decision. Among these factors the most important are training time and development time. All of these models that we tested are of equal difficulty to implement. Our best performing model, MLP, was also took significantly more time to train than all other models. Therefore, we are recommending the use of k-Nearest Neighbours due to its high accuracy across different feature sets and quick training time. All code used in this experiment can be found at: https://github.com/EGoms/4442

VI. References

[1] Lee, Ji & Hong, Hyung & Kim, Ki & Park, Kang. (2017). A Survey on Banknote Recognition Methods by Various Sensors. Sensors. 17. 313. 10.3390/s17020313.

[2] Kim, E.; Turton, T. The next generation banknote project. Bull. Reserve Bank Aust. 2014, 2014, 1–12.

[3] Park, Y.H.; Kwon, S.Y.; Pham, T.D.; Park, K.R.; Jeong, D.S.; Yoon, S. A high performance banknote recognition system based on a one-dimensional visible light line sensor. Sensors 2015, 15, 14093–14115.

[4] Pham, T.D.; Park, Y.H.; Kwon, S.Y.; Nguyen, D.T.; Vokhidov, H.; Park, K.R.; Jeong, D.S.; Yoon, S. Recognizing banknote fitness with a visible light one dimensional line image sensor. Sensors 2015, 15, 21016–21032.

[5] A. Ahmadi, S. Omatu, T. Fujinaka, and T. Kosaka, "Improvement of reliability in banknote classification using reject option and local PCA," In Information Sciences, Vol. 168, No.1, pp. 277-293, 2004.

[6] C. Barillot, and N. Navab, "Medical Image Computing and Computer-Assisted Intervention," MICCAI 2013, Berlin, Germany, Springer, 2013, Nagoya, Japan.

[7] Pham, Tuyen Danh et al. "Multi-National Banknote Classification Based on Visible-light Line Sensor and Convolutional Neural Network." Sensors (Basel, Switzerland) vol. 17,7 1595. 8 Jul. 2017, doi:10.3390/s17071595

[8] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[9] C.W. Hsu, C.C. Chang, and C.J. Lin. A Practical Guide to Support Vector Classification. Technical report, 2000.

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