Group Members:

1886648 – Mikayla Narothan

1963195 – Oluwademilade Osikoya

676400 – Kabelo Komape

1879247 – Leseli Mothibe

ML Assignment

2020

1. Data Set Information:

The data was extracted from images that were taken from genuine and forged banknote-like specimens. For digitization, an industrial camera usually used for print inspection was used. The final images have 400x 400 pixels. Due to the object lens and distance to the investigated object grey-scale pictures with a resolution of about 660 dpi were gained. A Wavelet Transform tool was used to extract features from images.

Number of samples: 1372
Number of features: 4
Features:

- variance of Wavelet Transformed image (continuous)
- skewness of Wavelet Transformed image (continuous)
- curtosis of Wavelet Transformed image (continuous)
- entropy of image (continuous)

Target: class (binary)

- 0 if forged
- 1 if genuine

Feature characteristics: Real
Missing values: None

Our dataset has four attributes to determine if a banknote is fake or real.
We are going to create several algorithms to learn from this dataset so that it can predict if a banknote is real or fake based on the 4 given attributes.

Sample data points:

	variance of image	skewness of image	curtosis of image	entropy of image	class
246	1.64720	0.482130	4.74490	1.22500	0
996	-2.31420	2.083800	-0.46813	-1.67670	1
252	0.57340	9.193800	-0.90940	-1.87200	0
1139	-1.52280	-6.478900	5.75680	0.87325	1
15	4.67650	-3.389500	3.48960	1.47710	0
1099	-0.71868	-5.715400	3.82980	1.02330	1
473	2.82090	7.310600	-0.81657	-1.87840	0
286	1.34190	-4.422100	8.09000	-1.73490	0
1112	-4.14290	2.774900	0.68261	-0.71984	1
1175	-1.75490	-0.080711	-0.75774	-0.37070	1

2. How data was structured:

Originally the data was in a text file but to make reading in the dataset easier and more efficient we converted it to a csv file. The Logistic Regression algorithm however used the text file format.

The data was checked for missing values but there were none. The feature values were in the same order of magnitude so no normalization. There were very few features so PCA was not needed. The data only consisted of real values besides the class which was binary so no encoding was needed.

To form the input matrix the class column was dropped forming a 1372 X 4 matrix. To form the target vector the class column was extracted.

Following this the data was then split into training, validation and testing data. We employed 2 methods of splitting the data. In the Decision Trees and Naïve Bayes algorithms, we used the Sklearn *train_test_split feature* to achieve this. While we explicitly randomized the order of the data and then split it in a for loop in the Logistic Regression Algorithm.

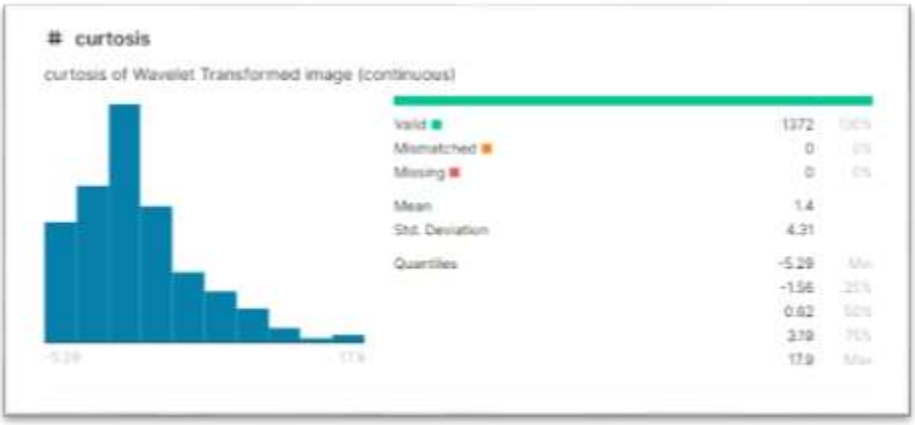
Training data consisted of 60% of the data (822 instances) for all 3 algorithms. The Decision Tree and Logistic Regression algorithms had testing data that was 20% of all available data (275 instances) as well as validation data that was also 20% of all available data. For reasons to be later discussed, the Naïve Bayes algorithm did not make use of validation data but instead had testing data that was 40% (550 instances) of all available data.

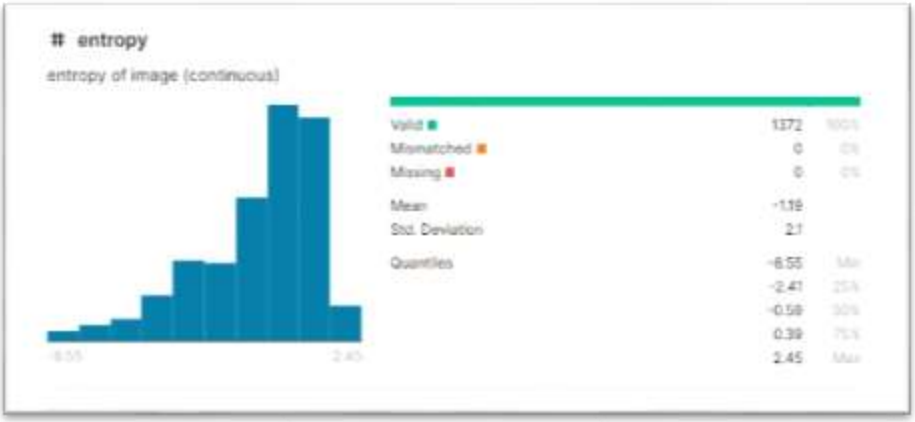
We explored three different ways of reading in the data. One for each algorithm.

- **Decision Tree:** The data was read in as a pandas dataframe and then when applying the algorithm it was converted to a numpy array.
- **Naïve Bayes:** The data is read into a numpy array using the *genfromtxt* feature.
- **Logistic Regression:** The data is read directly into a numpy array from a text file.

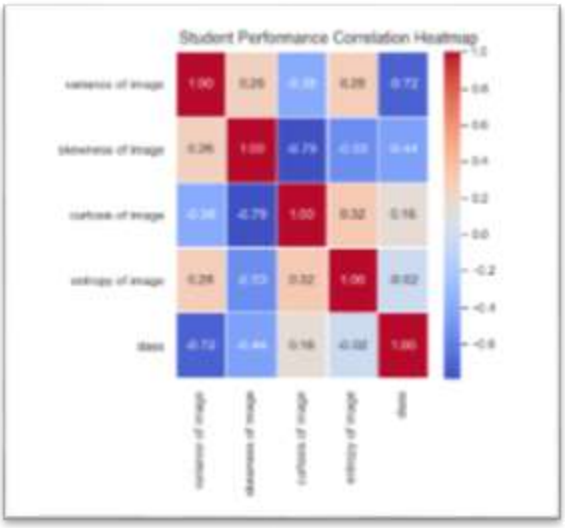
Data Visualizations:

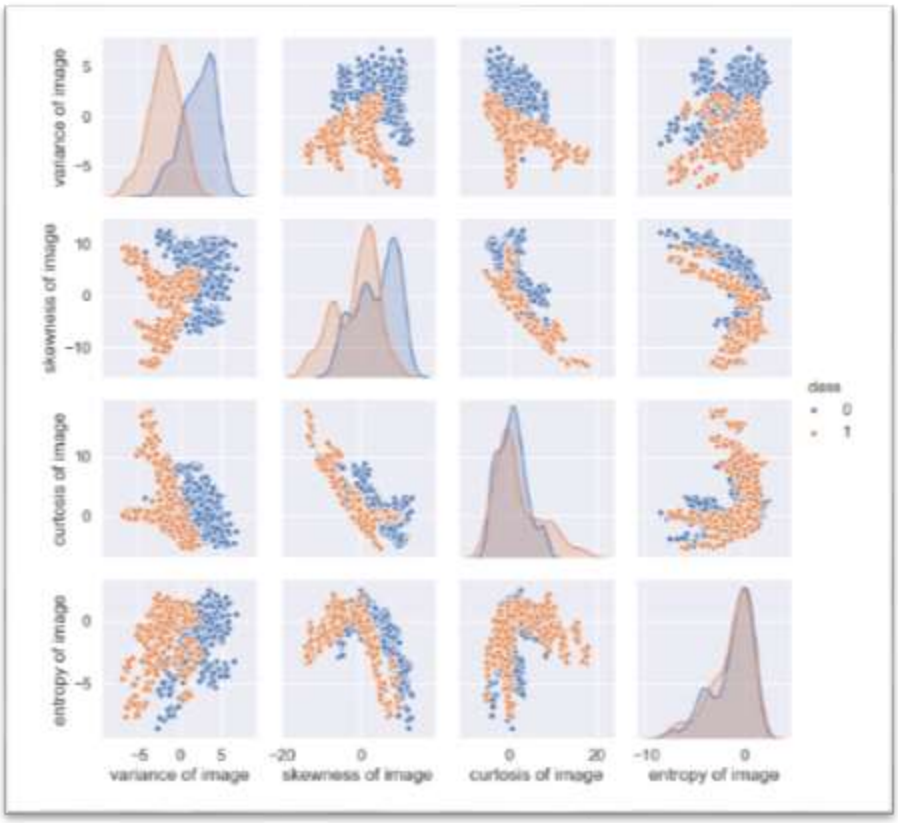

variance
variance of Wavelet Transformed image (continuous)
Valid 1372
Mismatched 0
Missing 0
Mean 0.43
Std. Deviation 2.84
Quantiles
-7.04
-1.77
0.5
2.82
6.82


curtosis
curtosis of Wavelet Transformed image (continuous)
Valid 1372
Mismatched 0
Missing 0
Mean 1.4
Std. Deviation 4.31
Quantiles
-5.29
-1.56
0.62
3.19
17.9


entropy
entropy of image (continuous)
Valid 1372
Mismatched 0
Missing 0
Mean -1.19
Std. Deviation 2.1
Quantiles
-8.55
-2.41
-0.59
0.39
2.45


0
55.5%
44.5%
1
class





There are no outliers in the data and the data samples have an almost equal number of genuine to forged outcomes so this indicates the learning will not be biased.

3. Classification Algorithms

3 classification algorithms were implemented:

- Decision Tree
- Naïve Bayes
- Logistic Regression

Decision Tree:

We chose the decision tree algorithm because it is a fairly simple and straight forward approach when training classification data. Decision trees make all possible alternative paths explicit and traverses each alternative path which makes comparison among alternatives easy. Decision trees have reduced ambiguity when decision-making since it assigns specific values to outcomes of decisions. Every possible scenario from a decision finds representation by a node.

The ID3 algorithm was used which creates the tree based on the entropy and information gain of a potential split in the tree.

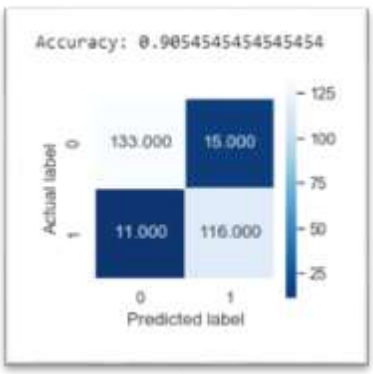
Entropy was calculated using the formula:

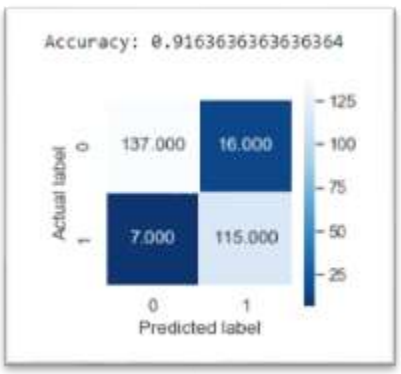
$$H(D) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

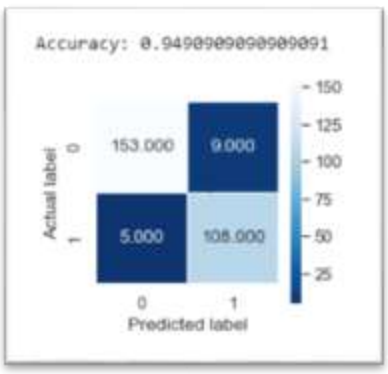
Information gain was calculated using the formula:

$$Gain(D, F) = \mathrm{H(D)} - \frac{1}{|D|} \sum_{\mathrm{f} \in \text{values of F}} |D_f| H(D_f)$$

Examples of accuracy and confusion matrices when learning with the decision tree:







Naïve Bayes:

We chose to implement the Naïve Bayes classifier because we chose data with very few features as well as a normal distribution. These conditions are optimum for the application of the Naïve Bayes classifier. Unfortunately, the algorithm's independence assumption limits the accuracy of its class predictions. Additionally, the algorithm does not have leeway in the form of hyper parameters as it is dictated by the strict formula of Bayes' Rule:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

This further limits the algorithm's accuracy as there is no opportunity to reduce overfitting using validation data. When implementing the algorithm, we used the following form for class conditionals to ease the programming process without compromising on accuracy.

$$p(x_i|y_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{(x_i-\mu_j)^2}{2\sigma_j^2}}$$

After training the model using training data, we used the testing data set to check the accuracy of our model in a function that we named predict. The modal accuracy for our model is 84.76% and a confusion matrix achieved through the use of the Sklearn confusion_matrix can be found below.

		Actual Class	
		0	1
Predicted class	0	136	21
	1	27	91

Logistic Regression:

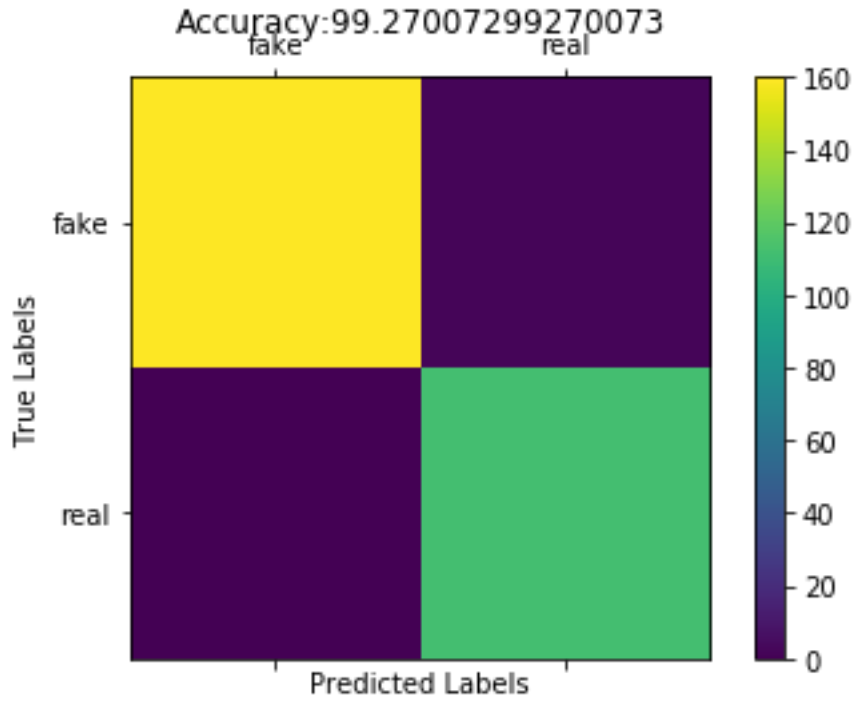
We chose this algorithm because our dataset has a binary outcome and the logistic regression formula gives a probabilistic output that is perfect for datasets that requires binary outputs. We did not regularize the algorithm because it was running fine without regularization and the weights are not getting too big so we felt regularization was not needed.

The basis function we used was:

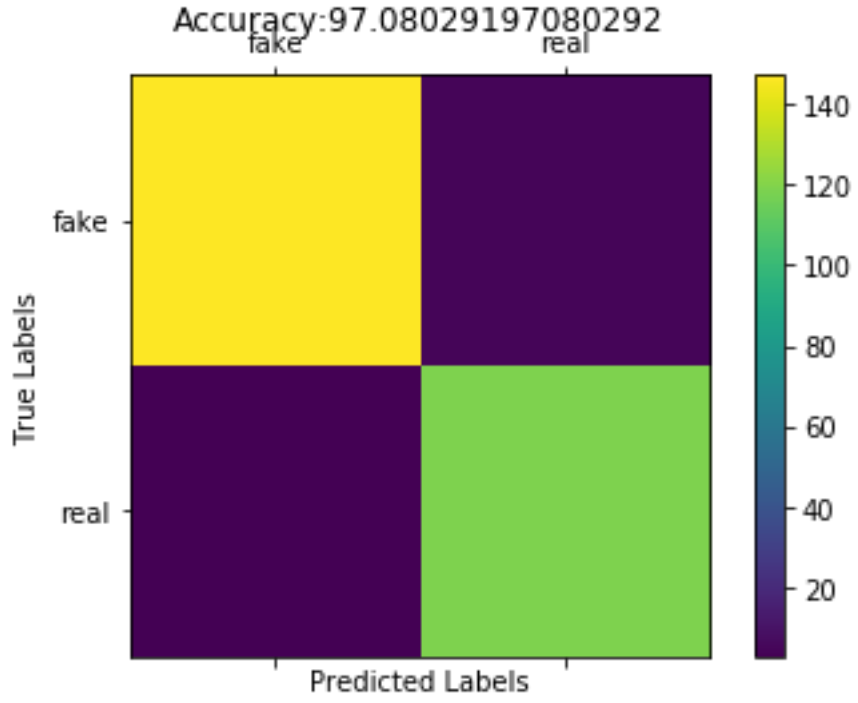
$$y = x0a0 + x1a1 + x2a2 + x3a3 + x4a4$$

where x0 is 1 and is our bias. We trained the model using gradient decent method with a learning rate of 1e-2 and a tolerance of 0.001. After checking multiple values of the tolerance and learning rate we found those values gave us the highest accuracy.

```
[[160   2]
 [  0 112]]
```


Accuracy:99.27007299270073
fake
real
fake
real
True Labels
Predicted Labels
160
140
120
100
80
60
40
20
0

```
[[147   5]
 [  3 119]]
```


Accuracy:97.08029197080292
fake
real
fake
real
True Labels
Predicted Labels
140
120
100
80
60
40
20

4. Results

Logistic Regression: As said in the previous question a learning rate of 1e-2 and a tolerance of 0.001 worked best. Using this algorithm and this dataset the best possible performance we achieved was an accuracy in the high 90%. We did that by training the model with over half of the dataset and further setting the parameters with the validation set.

Naïve Bayes: as previously mentioned, the most occurring accuracy for the data was 84.76%. This is a low accuracy in comparison to the other algorithms which we applied. We would thus recommend that a less general learning algorithm be used to classify the bank notes. Logistic regression has proven to be a stellar method of classification although others may be explored.

Decision Tree: The results from using a decision tree were really good, with the highest accuracy reaching up to 95%. The maximum depth allowed was 3 but if another dataset needed to be used this could be changed based on what that algorithm need, so it is very versatile. A large training dataset was used so this allowed for the training to be more accurate.

It seems the logistic regression worked best this might be due to the fact that our data does not have a lot of attributes and as such all the attributes are very useful in contributing to the output hence the algorithm's high accuracy. The best performance we got for this dataset is when our algorithm failed to classify just two bills correctly which was an accuracy of approximately 99.3%

Overall we would recommend that someone interested in our data set try using a different algorithm like neural networks as this dataset is fun to work with and experimenting with an algorithm we didn't use would be interesting. We would also recommend that this dataset be used as is without any pre-processing, such as standardization or normalization, as this doesn't change the data or effect the accuracy too much.

5. Code

For each algorithm there is a jupyter notebook in the Code folder. Run each kernel in order from the top of the notebook to the end to implement the algorithm. The dataset is stored in the Dataset Folder.

Helper Functions

- Sklearn
  - Train_test_split
  - Metrics
  - Confusion Matrix
  - StandardScaler
- Numpy
- Scipy
- Math
- Pandas
- matplotlib
- Collections
  - Counter