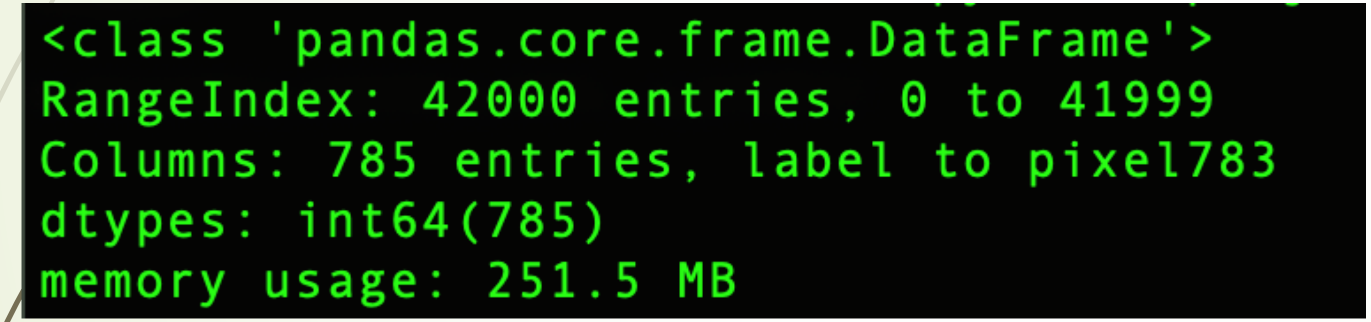
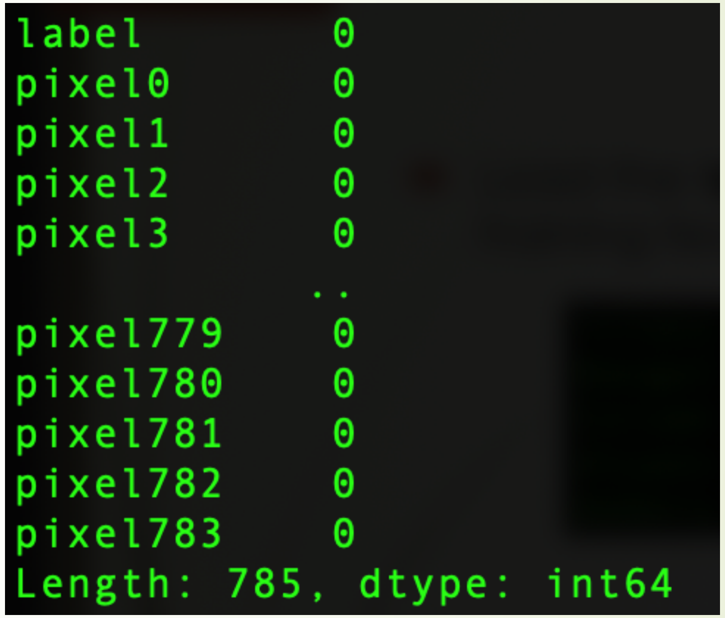
KNN for handwritten digit recognition

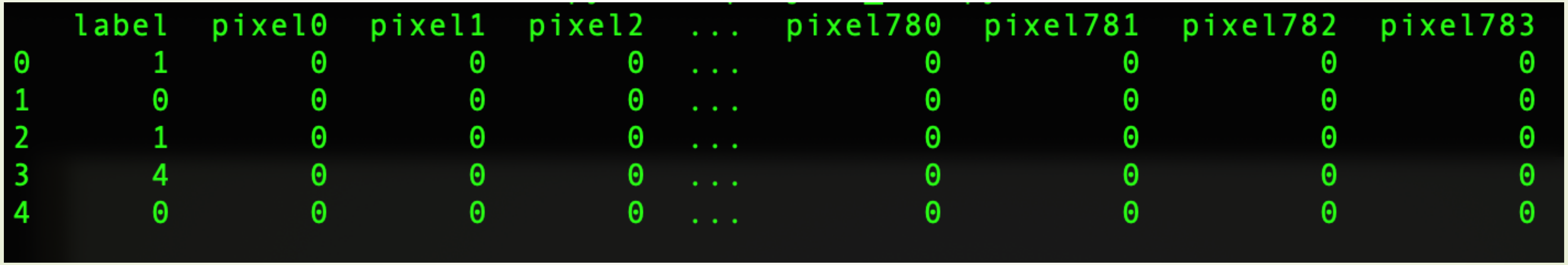
- Load the training dataset (42,000 images) and set the target (digit label) and training features (pixel values).
- Feature description: raw grayscale pixel intensities are used directly. No empty values and thus no features are dropped.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42000 entries, 0 to 41999
Columns: 785 entries, label to pixel783
dtypes: int64(785)
memory usage: 251.5 MB
```



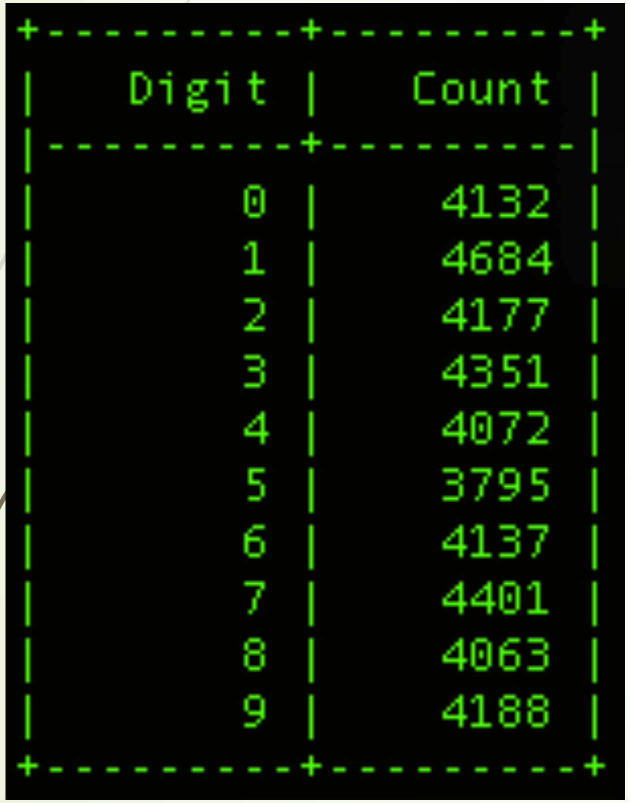
```
label       0
pixel0      0
pixel1      0
pixel2      0
pixel3      0
           ..
pixel779    0
pixel780    0
pixel781    0
pixel782    0
pixel783    0
Length: 785, dtype: int64
```



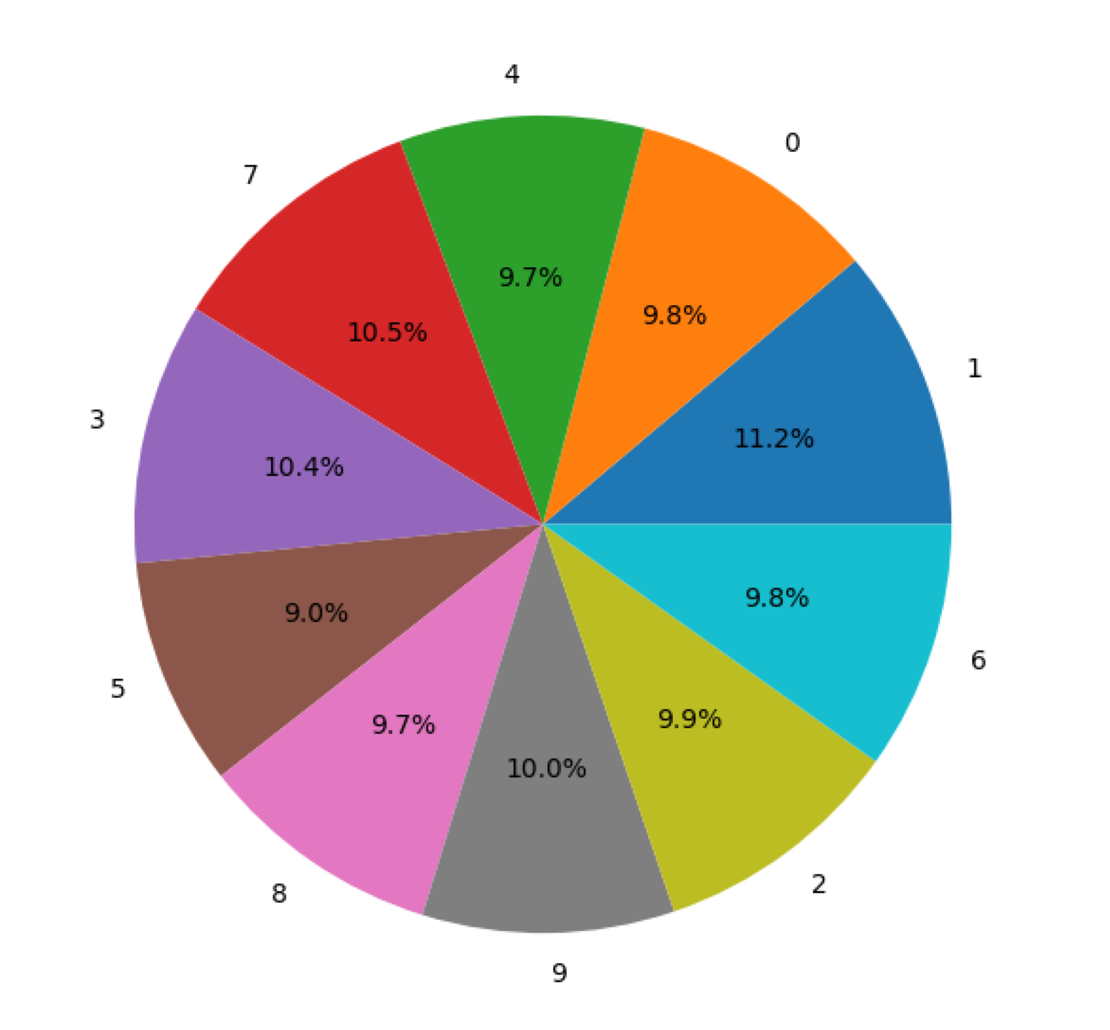
```
   label  pixel0  pixel1  pixel2  ...  pixel780  pixel781  pixel782  pixel783
0      1       0       0       0  ...         0         0         0         0
1      0       0       0       0  ...         0         0         0         0
2      1       0       0       0  ...         0         0         0         0
3      4       0       0       0  ...         0         0         0         0
4      0       0       0       0  ...         0         0         0         0
```

Training Data Analysis

- The count of each digit in the training data:

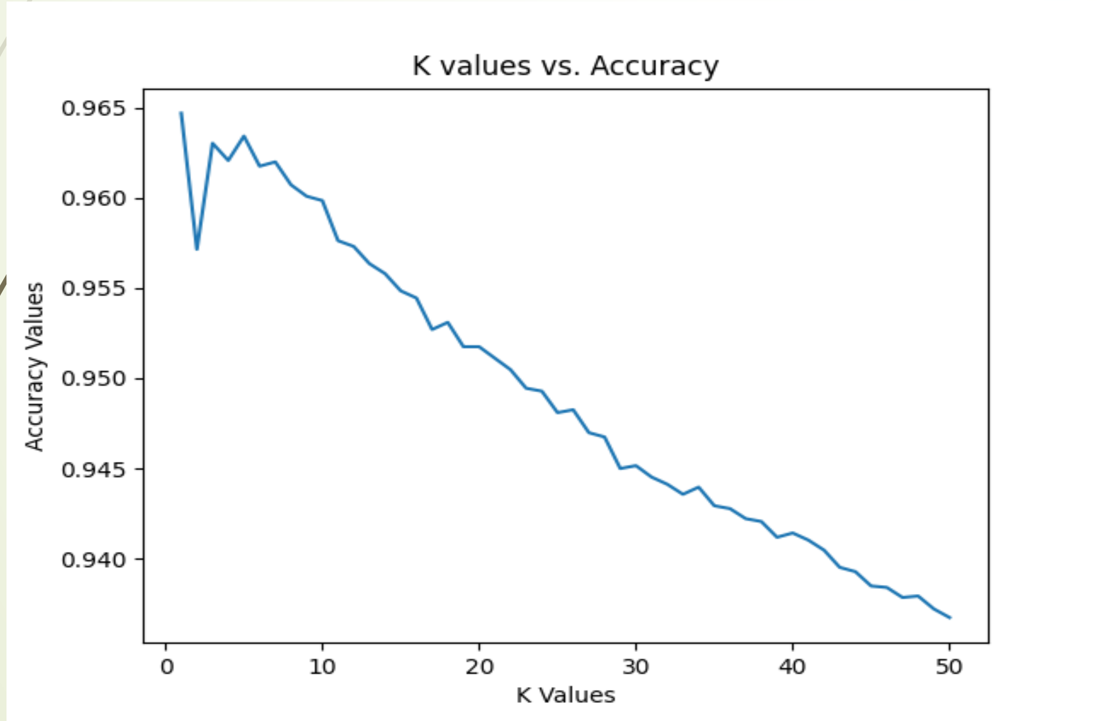


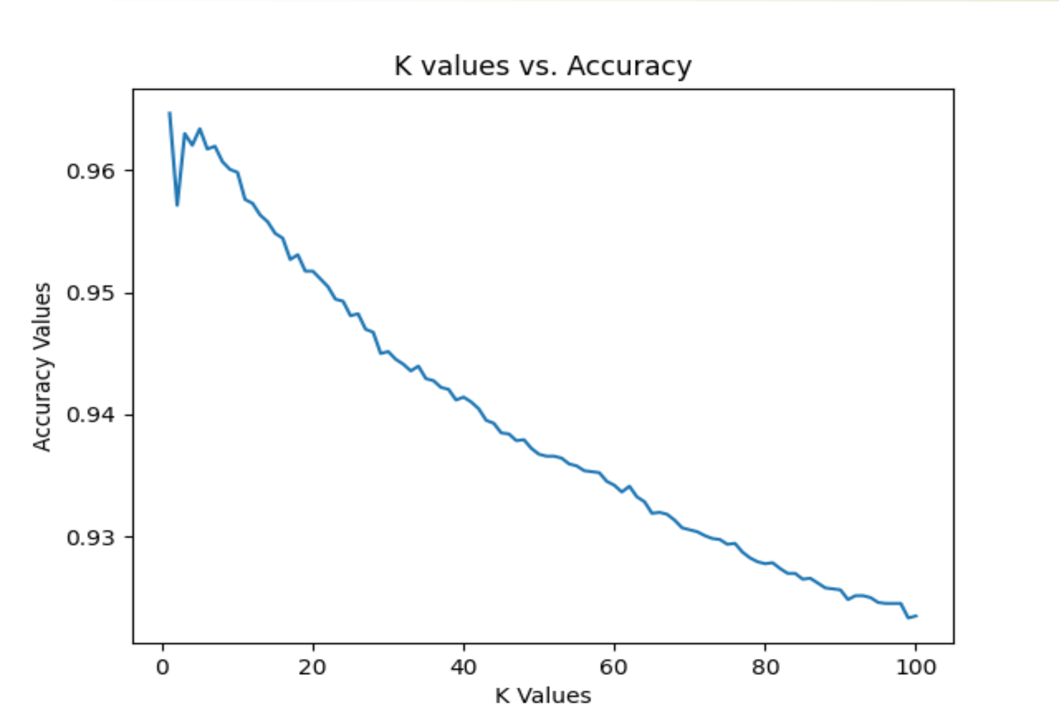
```
+---------+---------+
|   Digit |   Count |
|---------+---------|
|       0 |    4132 |
|       1 |    4684 |
|       2 |    4177 |
|       3 |    4351 |
|       4 |    4072 |
|       5 |    3795 |
|       6 |    4137 |
|       7 |    4401 |
|       8 |    4063 |
|       9 |    4188 |
+---------+---------+
```



Finding the best K value:

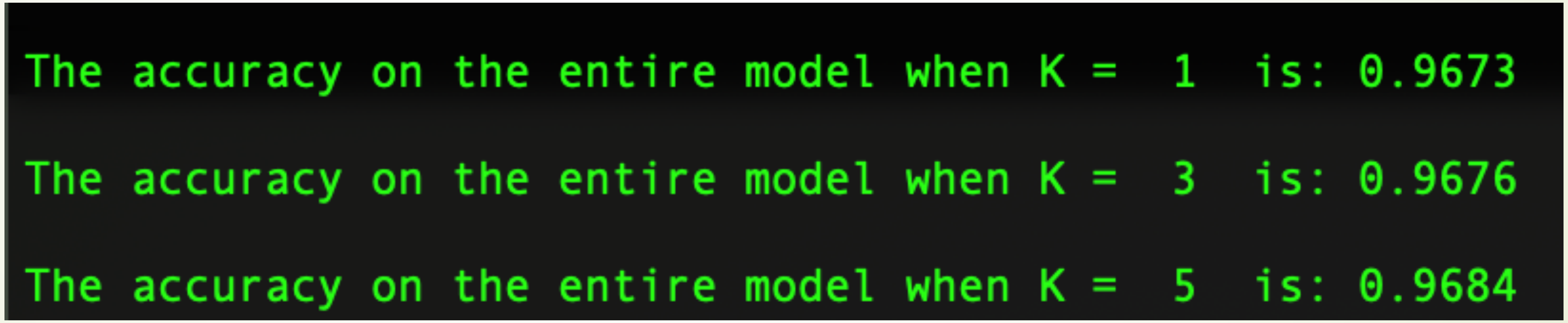
- The best K value for the KNN algorithm is found by randomly splitting the training data using train_test_split library from sklearn.
- The accuracies are found when K = 1 to 50 and K = 1 to 100. The results are stored in a CSV file. The three best accuracies are found when K = 1 (96.46%), the next best accuracy is found when K = 5 (96.34%) and finally when K = 3 (96.30%). These accuracies are very close to each other.





Training the model

- The model on the raw pixel intensities of the images in the training set and using the best K value.
- For this, a KNN classifier is created, the model is fit on X_train and Y_train. Once fitted, predictions are done on x_test. The predictions are then stored in y_pred. The actual label values are present in y_test.
- Accuracy is calculated on the training dataset by taking K = 1, the accuracy is 96.73%, when K = 3, the accuracy is 96.76% and when K = 5, the accuracy is 96.84%



```
The accuracy on the entire model when K =  1  is: 0.9673

The accuracy on the entire model when K =  3  is: 0.9676

The accuracy on the entire model when K =  5  is: 0.9684
```

Further plans with KNN

- I want to perform five-fold cross validation on the training data set and compute the average accuracy, precision, recall and F1 score over the five folds.
- For the next project checkpoint, I want to introduce my model to the testing dataset and be able to make prediction on the testing dataset.
- I want to find the accuracy of the model on the testing set
- I want to plot graphs to discuss my findings
- As a group, we want to compare the performance of different Machine Learning algorithms on the Kaggle handwritten digit recognition dataset and report our findings.

My GitHub link:

https://github.com/monicabernard/CAP-5610_Machine-Learning.git