




Digits Recognition

Project-03

Outline

- 1. Objective
- 2. Methodology
- 3. Results Discussion
- 4. Appendix

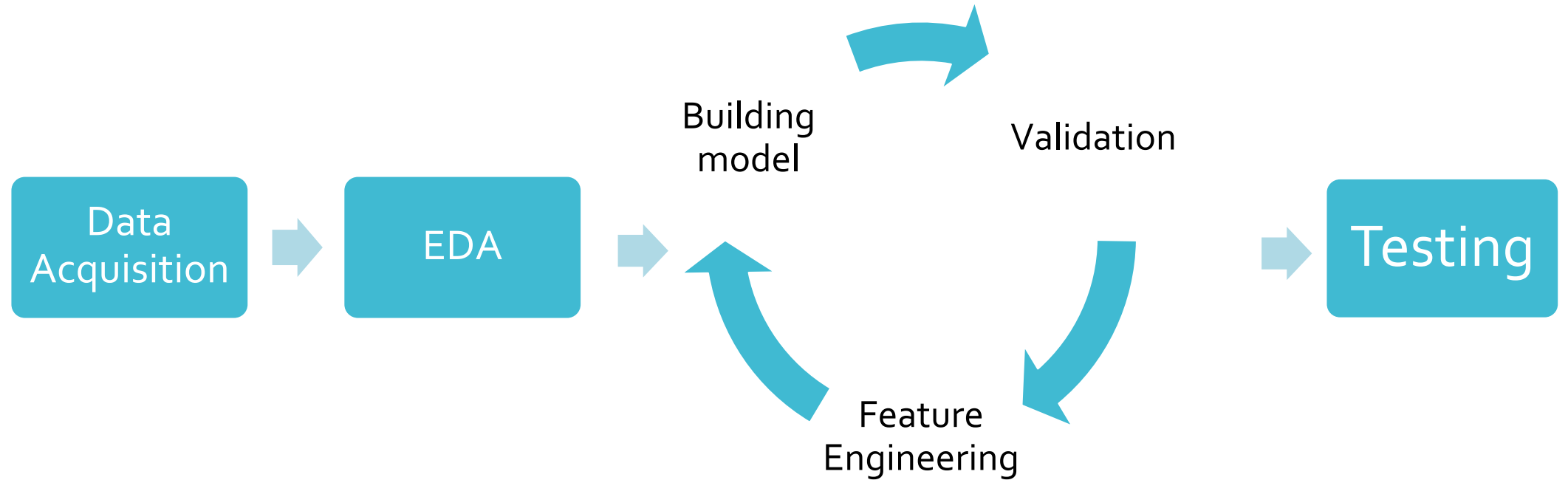


Deliver a machine
Learning model that
is able to classify
images of
handwritten digits

Objective &
expected outcome



Methodology

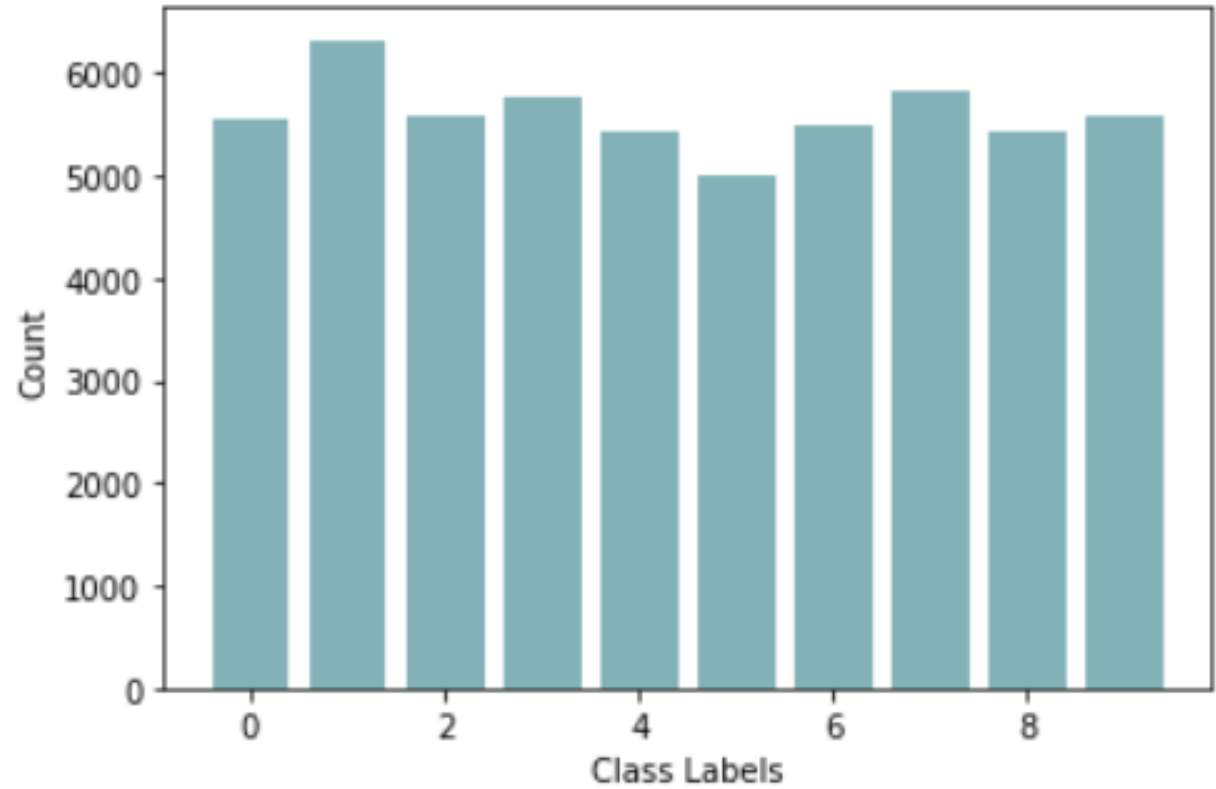


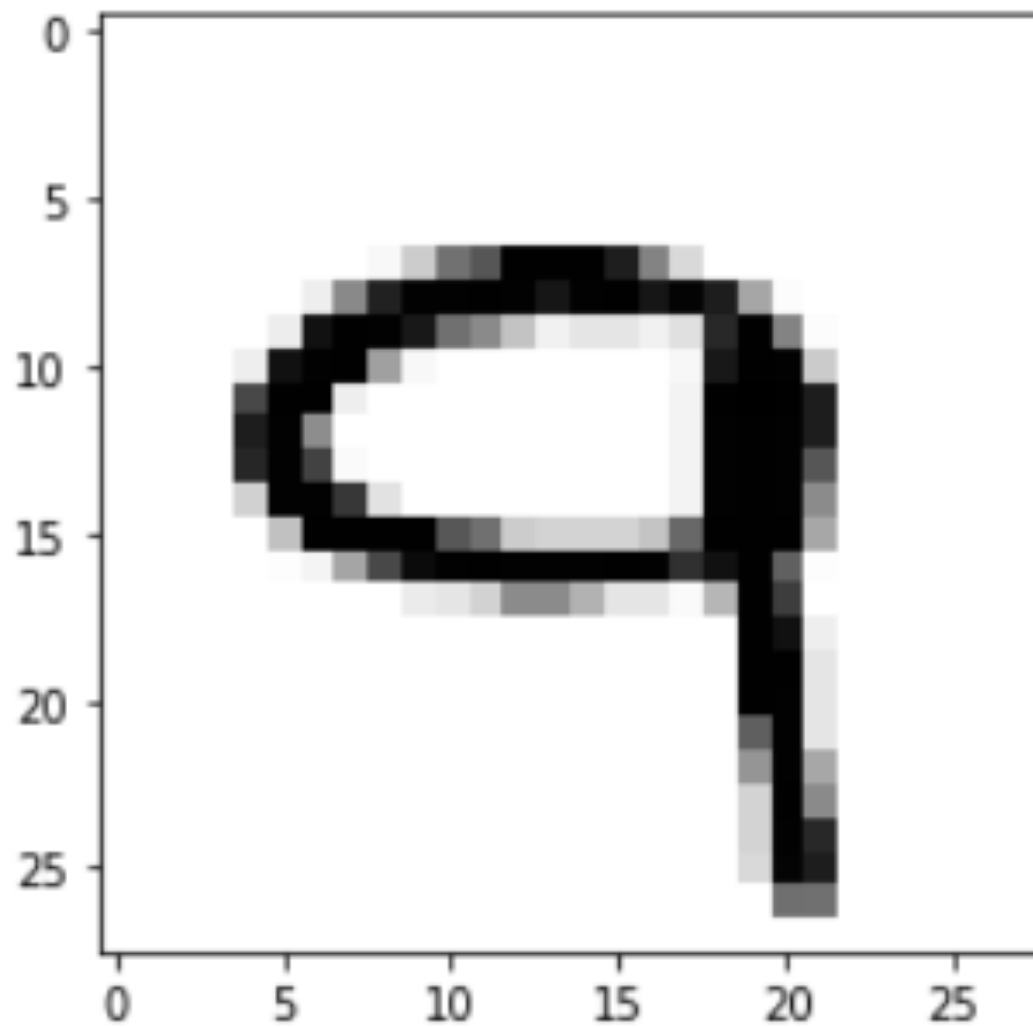
Data Acquisition and EDA

- Dataset: MNIST handwritten digits
- Features: 784 columns for each pixel
- Class: Digit value

[illegible]

Target Distribution





Pixels
plotted
using
matplotlib

Baseline

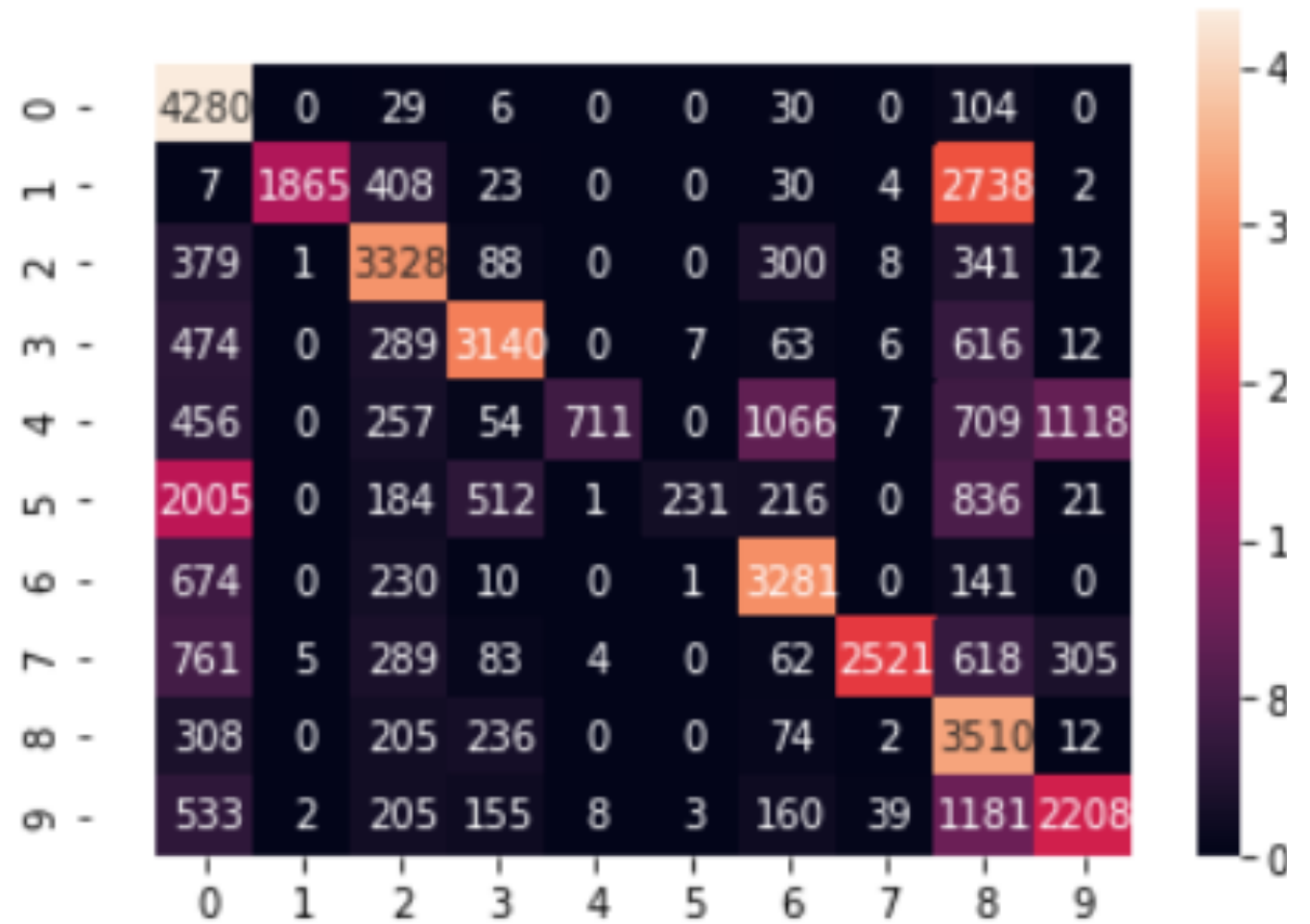
- KNN model.
- Cross validation accuracy score: **0.97** !
- Can you guess what the problem is?
- Bad time complexity!!

Custom Classifier

1. Take the features mean from every instance at each class.
2. Project the test samples through the template and calculate how much pixels made it through.
3. The more pixels pass through this template, the more likely this test sample is identical to this digit shape.



Confusion matrix for the custom classifier results.



	digit_0	digit_1	digit_2	digit_3	digit_4	digit_5	digit_6	digit_7	digit_8	digit_9
0	0.938766	0.150459	0.422304	0.464090	0.272487	0.508361	0.494927	0.352328	0.491920	0.396880
1	0.534857	0.158112	0.464102	0.369990	0.346009	0.405576	0.453420	0.322384	0.622958	0.429320
2	0.401613	0.294328	0.531838	0.380156	0.567754	0.366809	0.513486	0.513949	0.655019	0.593521

Feature Engineering

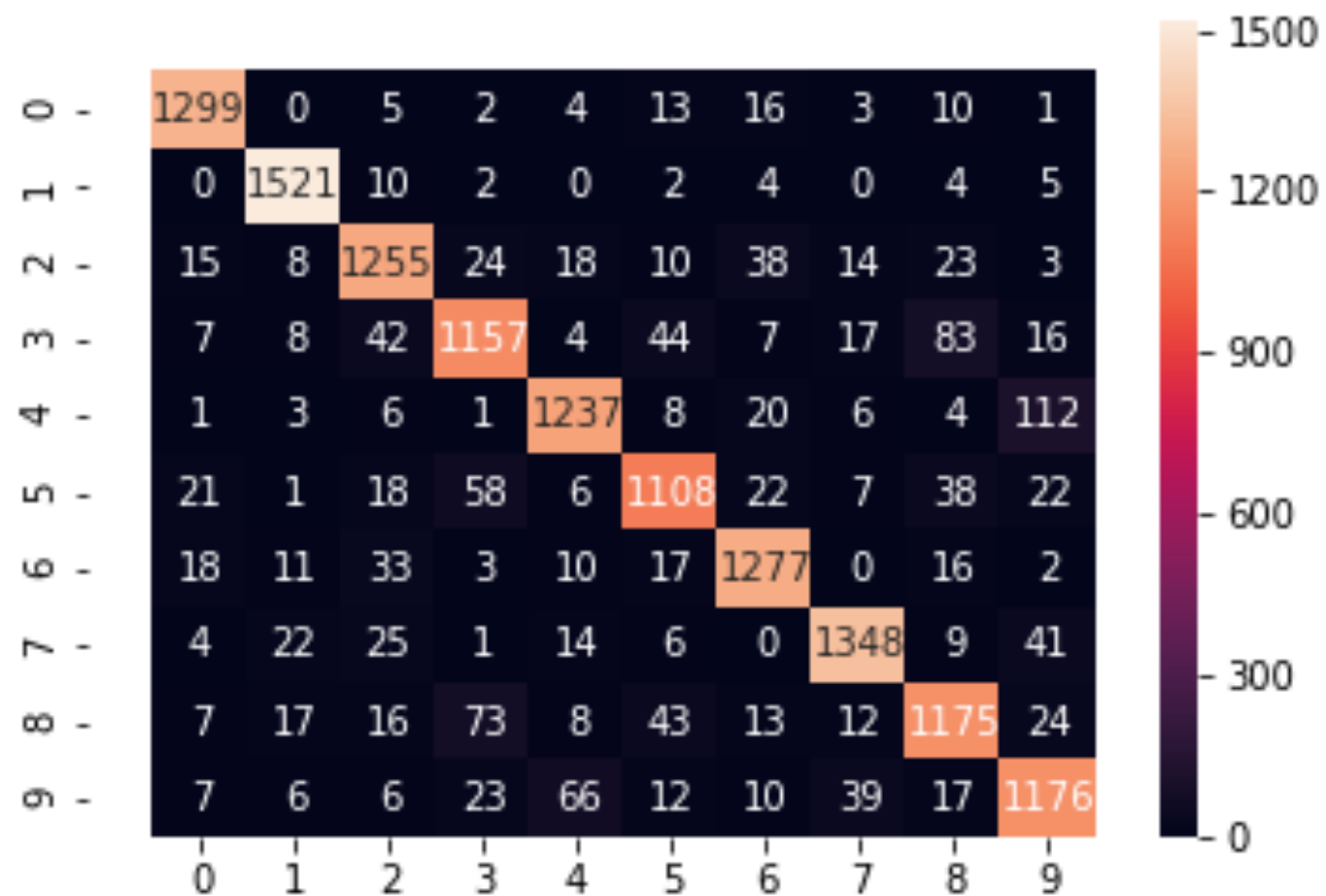
Experiment Results

Model	Training Score	Out-of-sample Score	Train time
KNN	0.9074	0.8695	10.8 s
Logistic Regression	0.7969	0.7830	5.46 s
SVM	0.7739	0.7753	3.87 s
Random Forest	1.0	0.8583	38.2 s
Naïve Bayes	0.5708	0.5120	45.3 ms

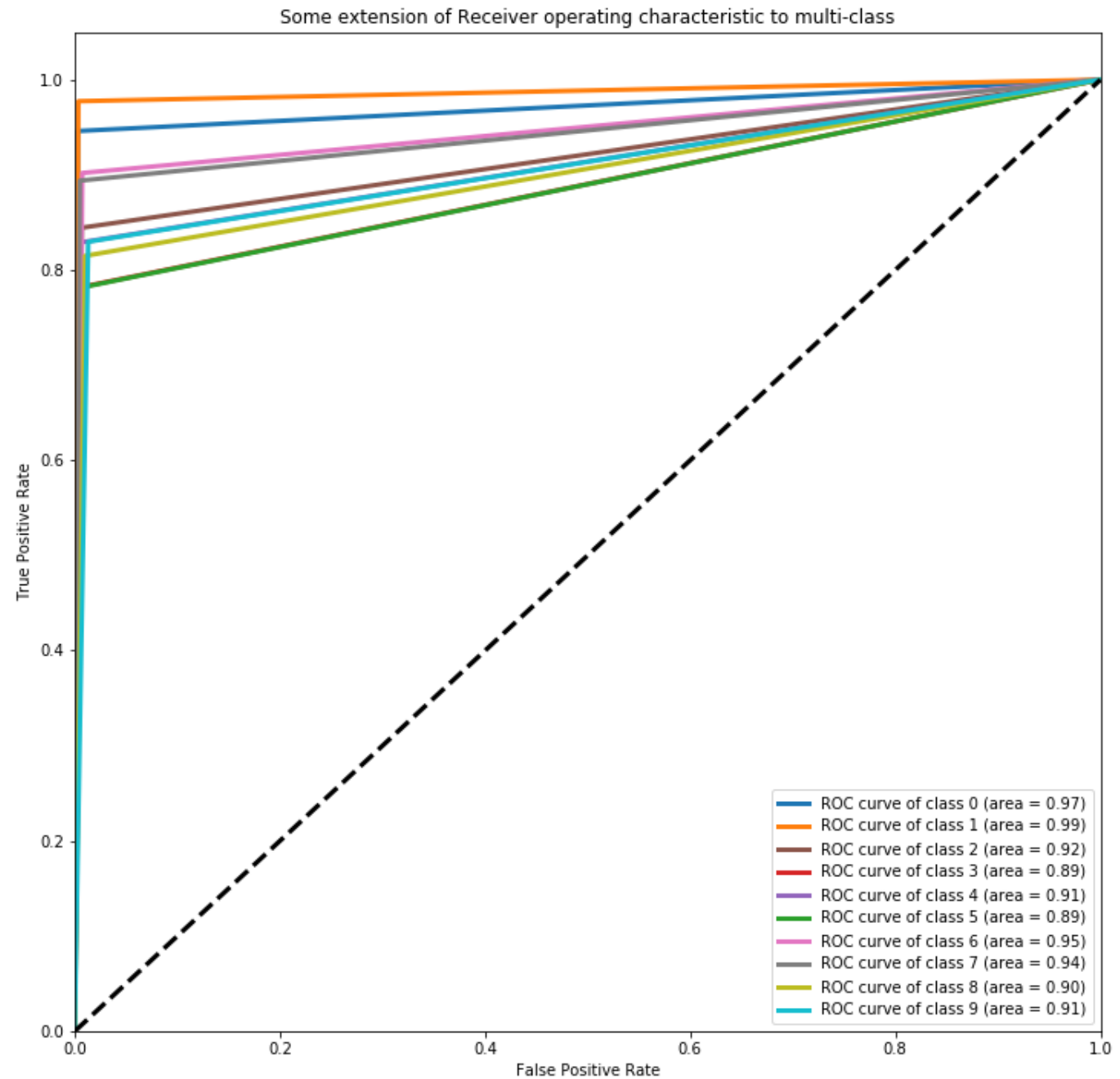
KNN Performance

Selected Model

Confusion Matrix



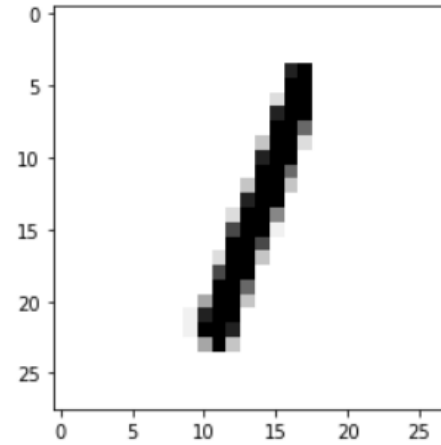
One-vs-Rest AUC



Sample Tests

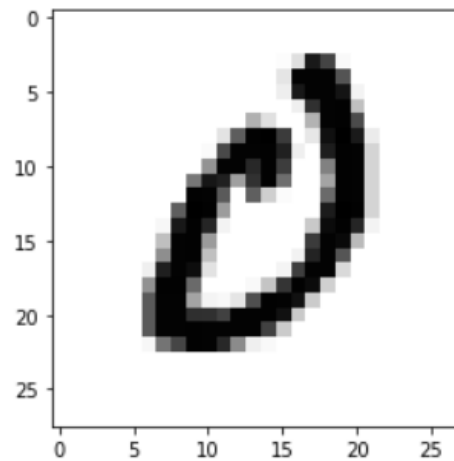
Predictid Label: 1.0

<matplotlib.image.AxesImage at 0x1498dec88>



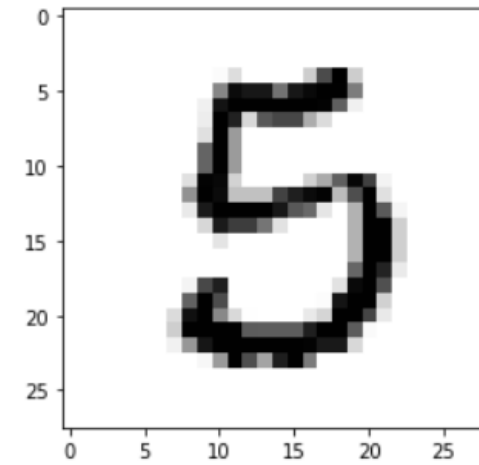
Predictid Label: 0.0

<matplotlib.image.AxesImage at 0x149604ac8>



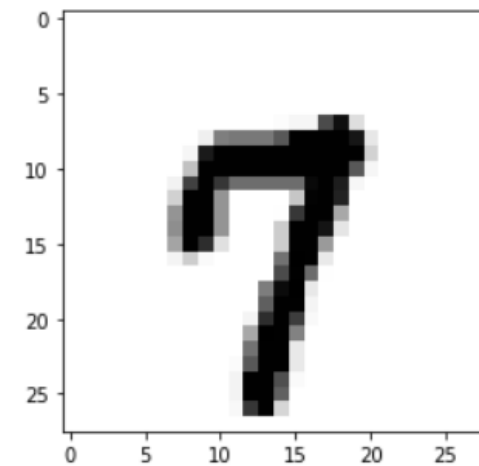
Predictid Label: 5.0

<matplotlib.image.AxesImage at 0x1494449b0>



Predictid Label: 7.0

<matplotlib.image.AxesImage at 0x1497edb38>



Questions

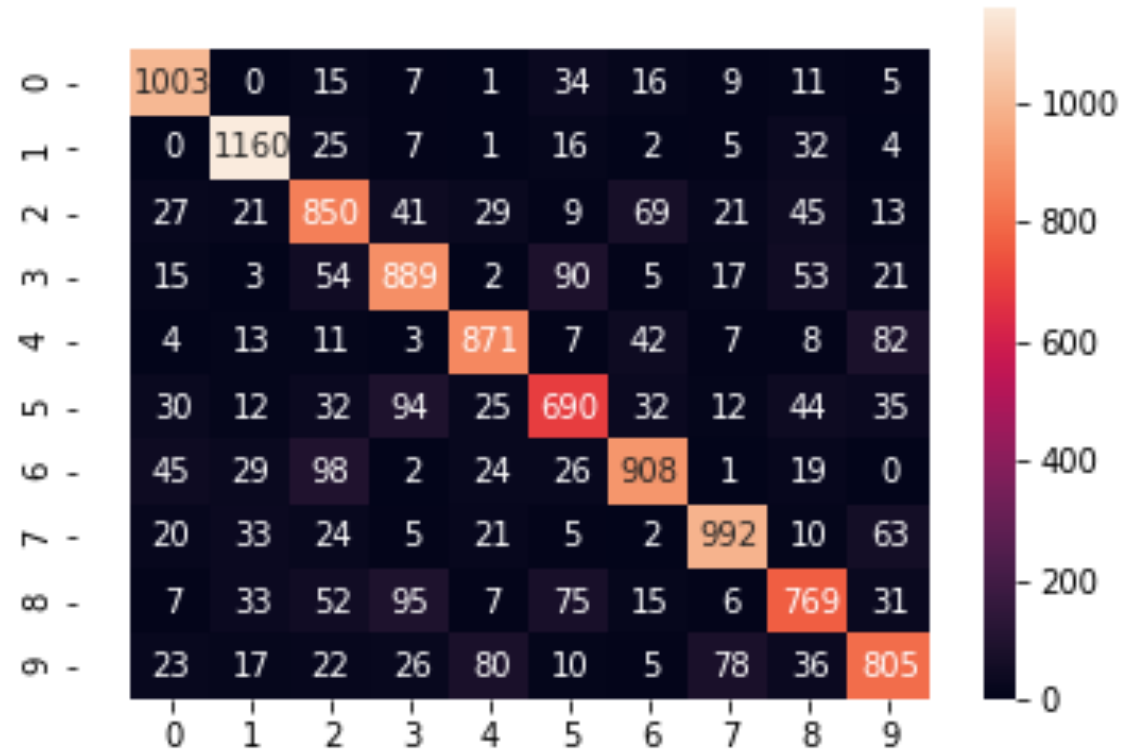


Appendix

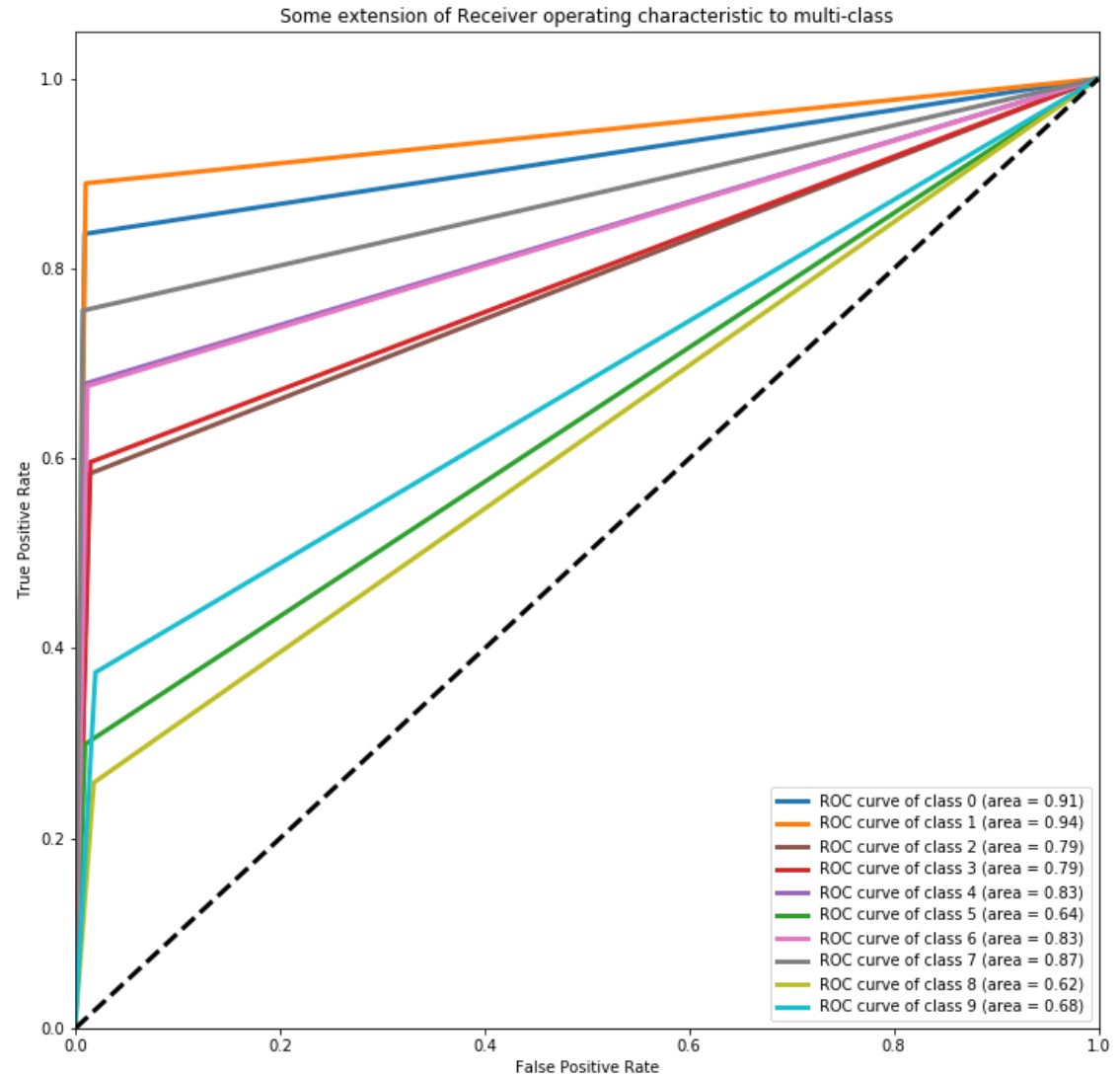


Logistic Regression

Logistic Regression Confusion Matrix

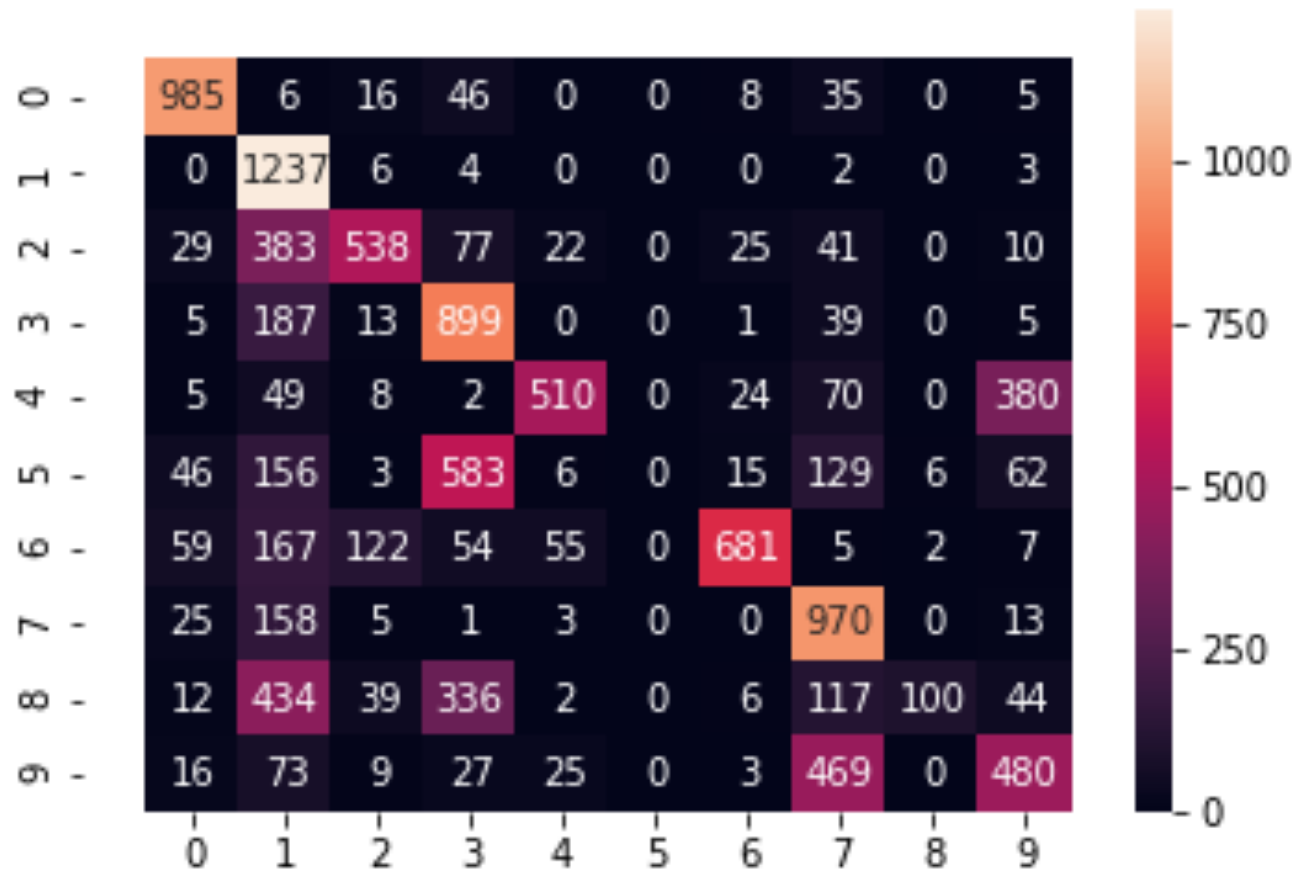


Logistic Regression One-vs-Rest AUC

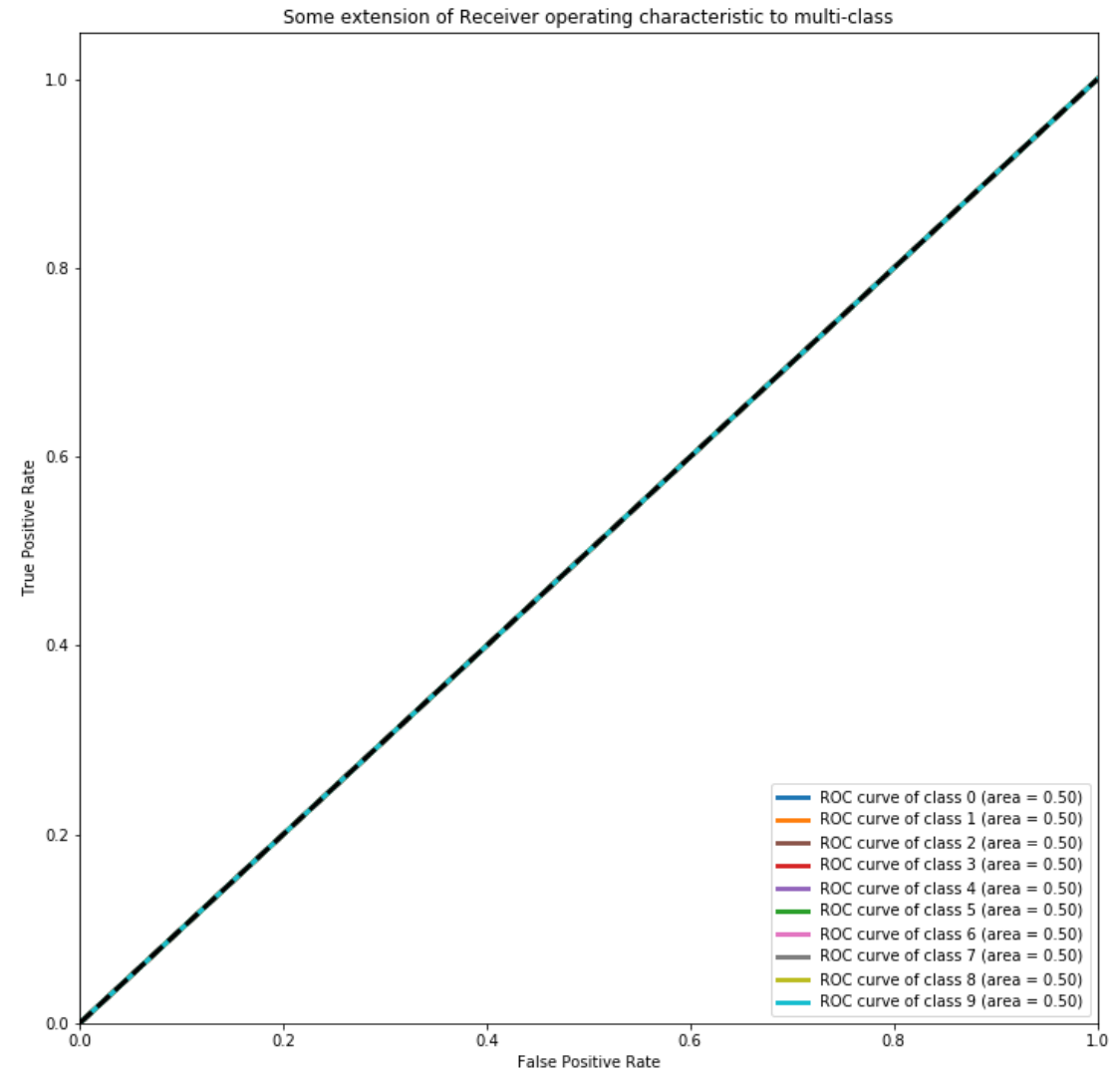


Naïve Bayes

Naïve Bayes Confusion Matrix

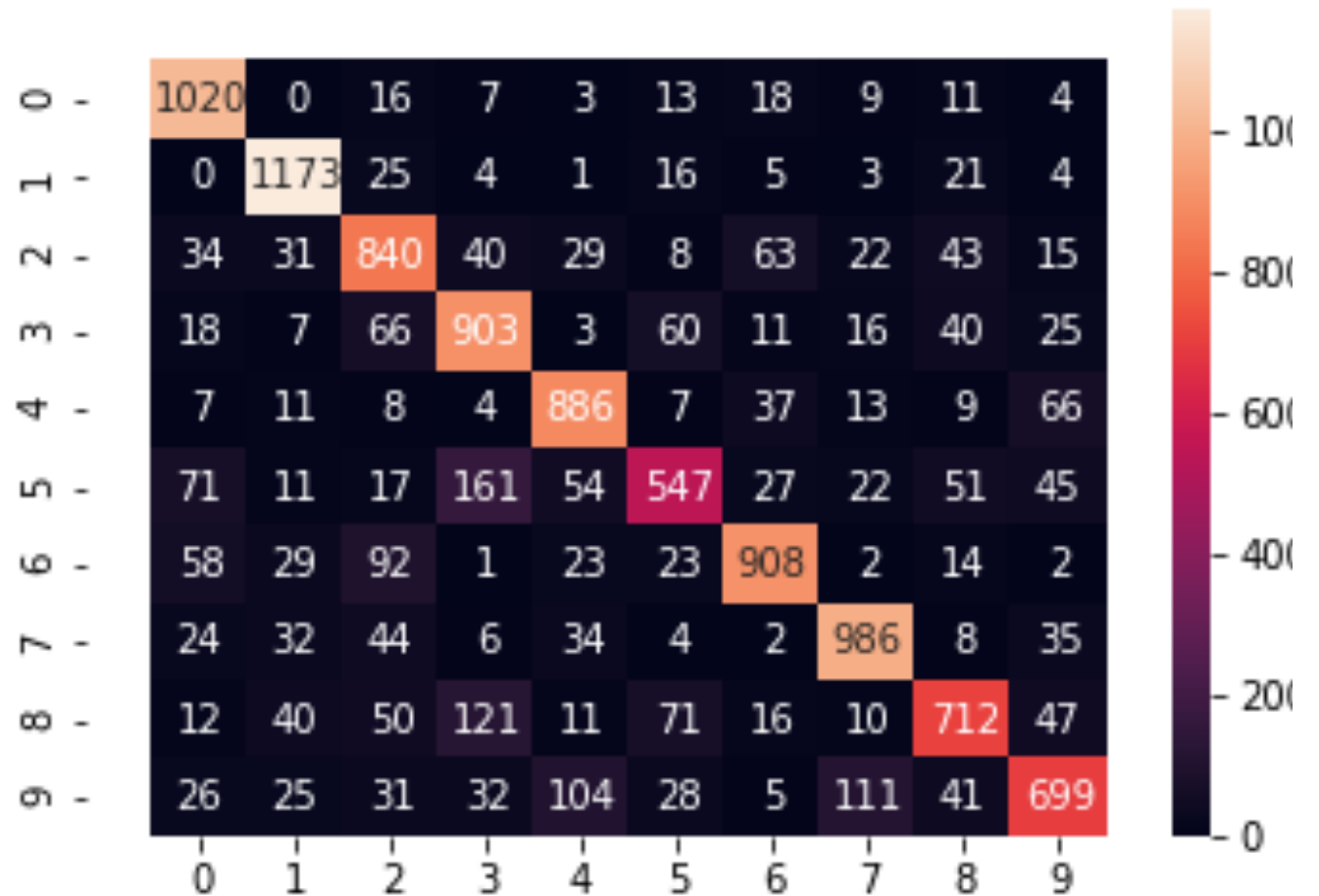


Naïve Bayes One-vs- Rest AUC

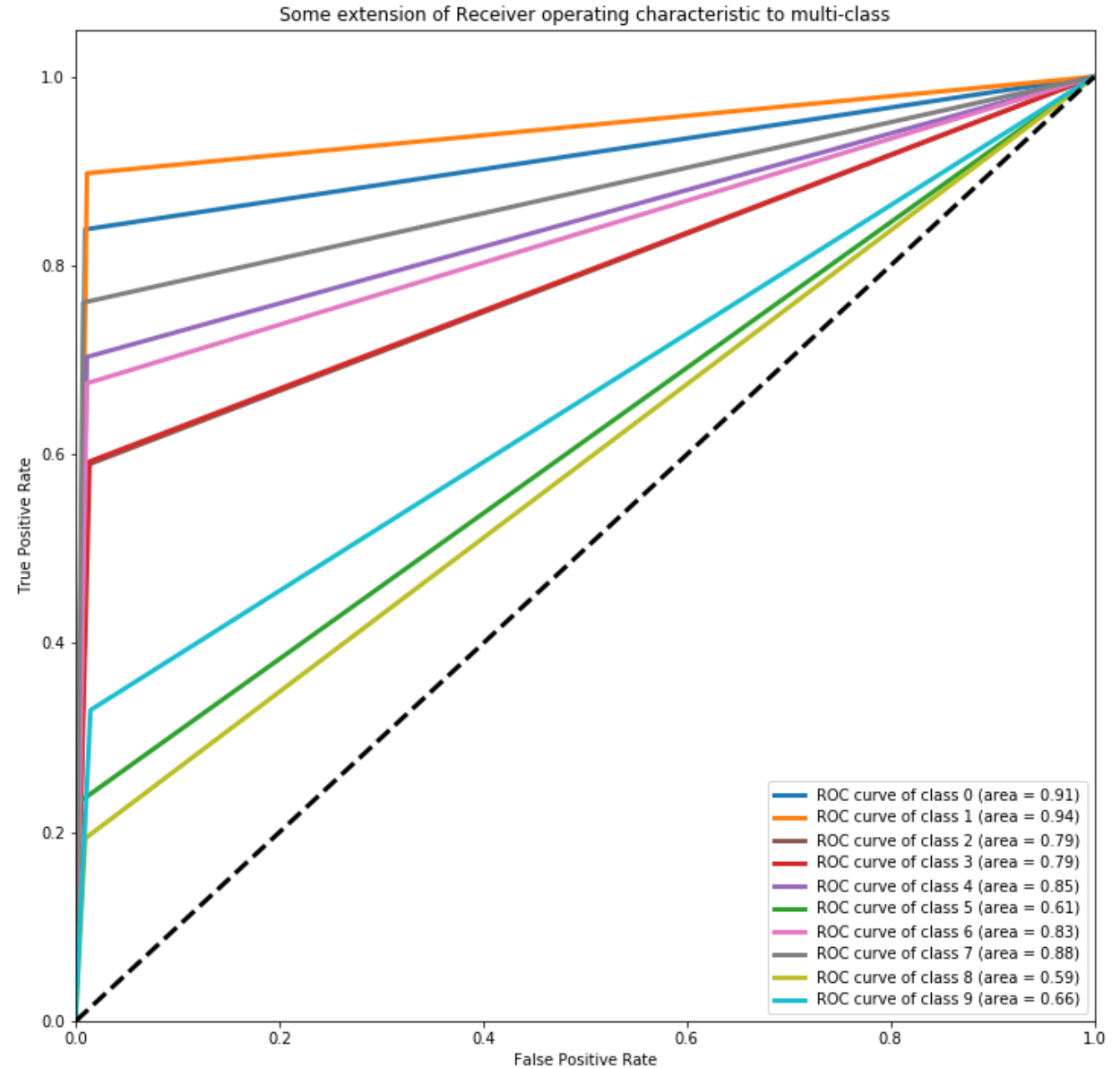


Support Vector Machine

SVM Confusion Matrix

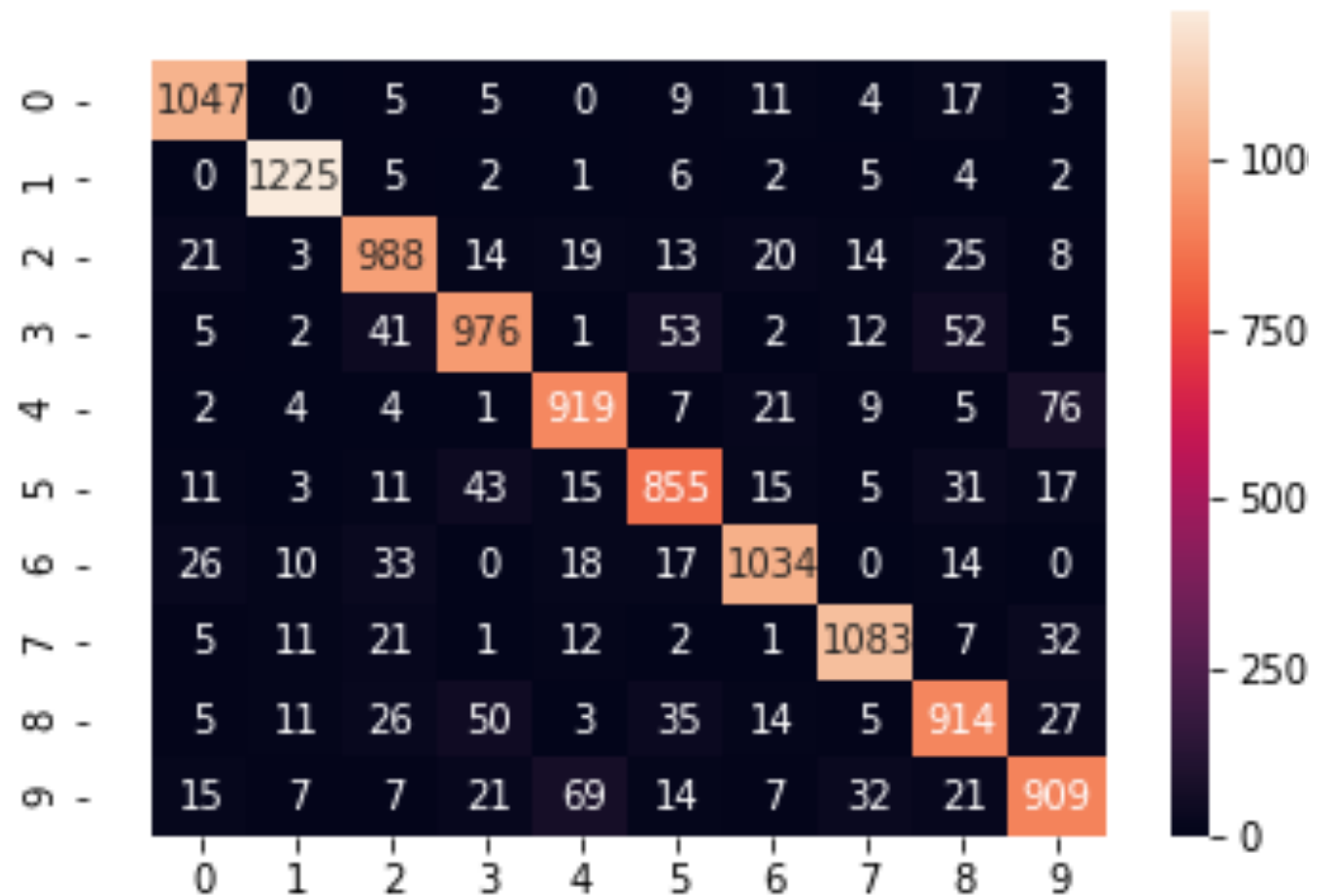


SVM One-vs- Rest AUC



Random Forest

Random Forest Confusion Matrix



Random Forest One-vs- Rest AUC

