MNIST:

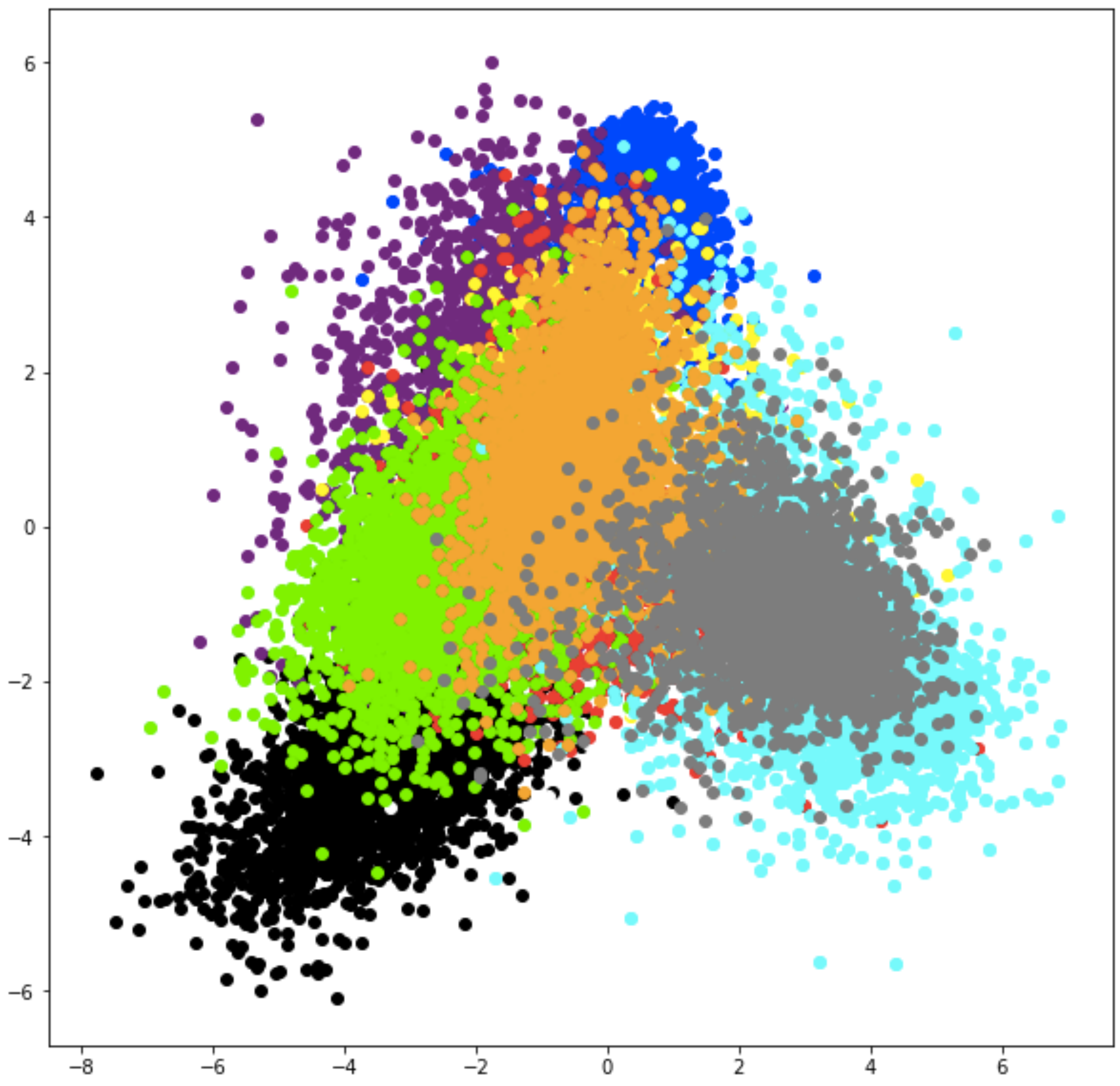
This dataset contains about 10 classes in it, each of the class is a hand written number.
So for each each hand written number we have a pixel information for with about 28*28 columns.

To Analyse this I did both LDA and PCA on the dataset and created two different data frames

In LDA when we look at the variance captured by 9 components is very less.

```
arrayarray([0.24158292, 0.200548   , 0.17738265,
0.10631457, 0.09465535,
       0.06893702, 0.04948719, 0.03455329, 0.02653901])
```

It captures about 100% of the variance in LDA. Below is the way in which data is captured.

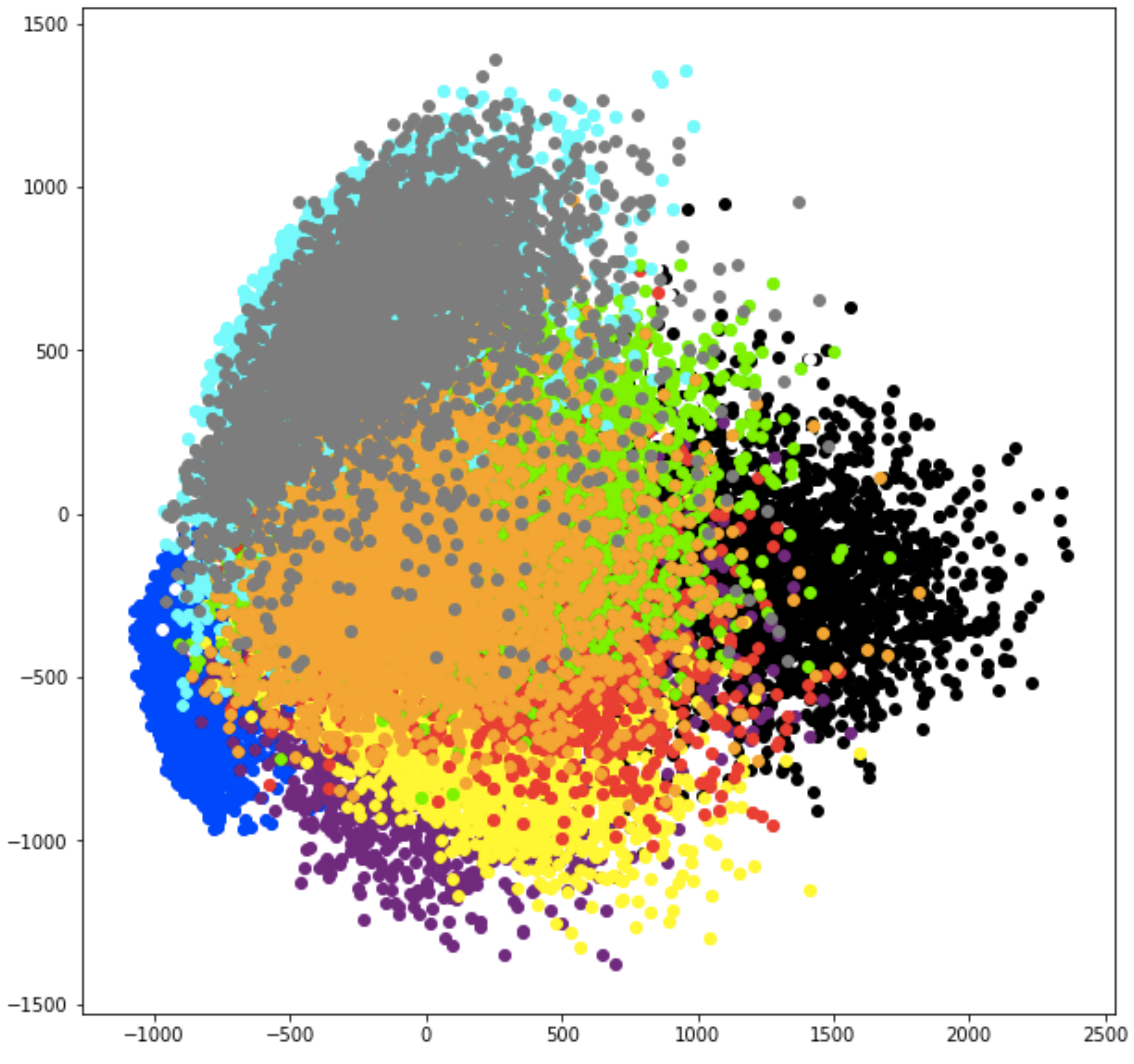


Lets look at the PCA:
When we look at the PCA Variance of the 9 components.

```
Explained variation per principal component: [0.09694147
0.07117981 0.06152888 0.05384314 0.04886116 0.04308593
 0.03283892 0.02875152 0.02745918]


Explained cumulative variation for 30 principal
components: [ 9.7 16.8 23.  28.4 33.3 37.6 40.9 43.8
46.5]
```

The Cumulative Sum is about 46% in the PCA.
THE PCA figure looks like



There is lot of overlap of the data when we look at PCA components.

MOdel 1:(LDA+Naive Bayes DIag Variance)
Lets fit a Bayesian Gaussian MIxture model with only Diagonal Variance. Lets set the LDA dataset for the first model. There is no much difference in the accuracy of training and test data set.

```
accuracy for LDA training data set 0.2833006993006993

accuracy for LDA testing data set 0.28
```

Model 2: (PCA+Naive Bayes DIag Variance):
In the second model the accuracy is pretty low it almost drops down to 1 percent on training dataset and increases to 9 percent of testing dataset.

```
accuracy for PCA training data set 0.01932867132867133
accuracy for PCA testing data set 0.09736082474226804
```

MOdel 3:(LDA+Naive Bayes FULL Variance)
When we look at the accuracy it is less than the model 1:

```
accuracy for LDA training data set 0.021622377622377623
accuracy for LDA testing data set 0.022762886597938143
```

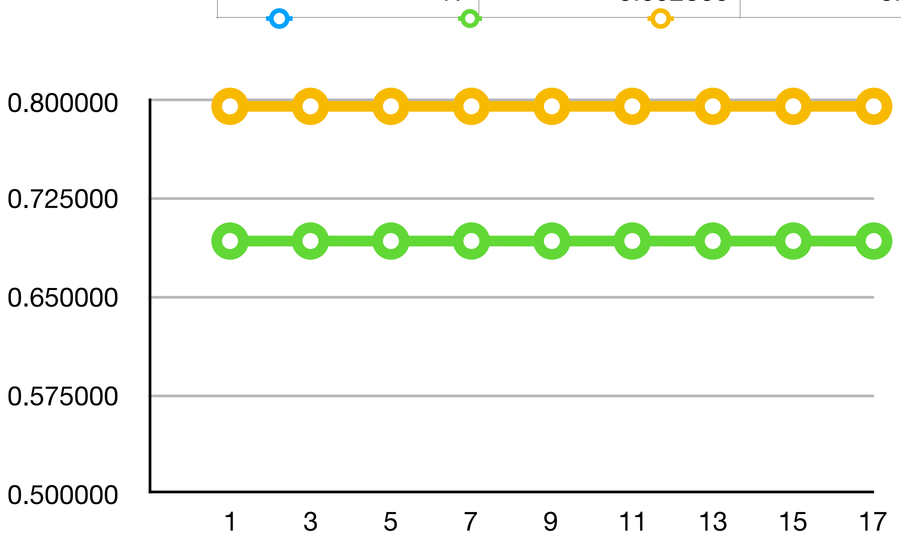
Model 4:(LDA+Naive Bayes FULL Variance)
When we look at the accuracy for the model4 the accuracy increase for the testing dataset but its same for training dataset.

```
accuracy for PCA training data set 0.08041958041958042
accuracy for PCA testing data set 0.09868041237113402
```

Model 5: KNN on PCA: In the below table the accuracy of the KNN algorithm is present. The train accuracy is about 79% and test accuracy is about 69%. Which is way more when compared to the Naive Bayes Gassian Mixture algorithm. The appropriate K value will be about 9 , to avoid complexity and overfitting as all are giving same accuracy.

KNN PCA

K	test_accuracy	train_accuracy
0	0	0
1	0.692866	0.796084
3	0.692907	0.796084
5	0.692866	0.796084
7	0.692907	0.796084
9	0.692866	0.796084
11	0.692907	0.796084
13	0.692866	0.796084
15	0.692907	0.796084
17	0.692866	0.796084

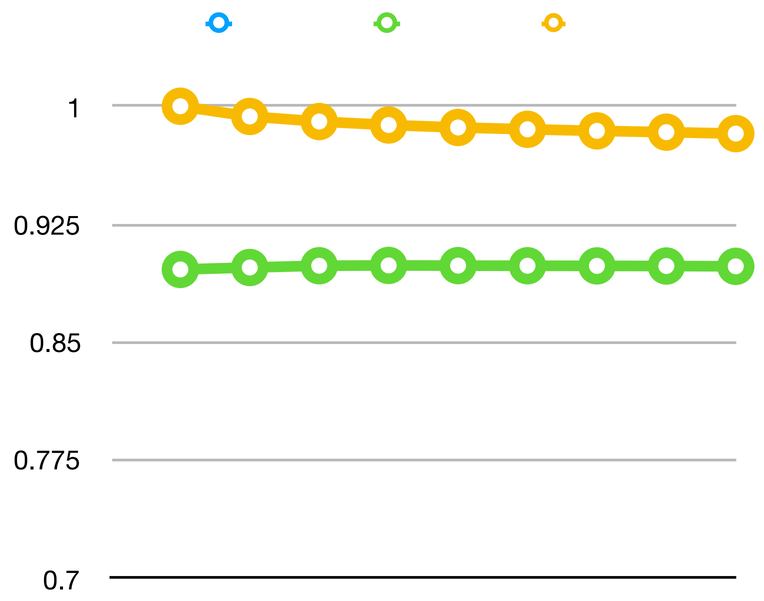


Model 6(KNN on LDA):

KNN- Accuracy

K	test_accuracy	train_accuracy
1.0	0.896371	1.000000
3.0	0.897608	0.993510
5.0	0.898722	0.990294
7.0	0.898928	0.988084
9.0	0.898804	0.986517
11.0	0.898722	0.985371
13.0	0.898639	0.984364
15.0	0.898557	0.983552
17.0	0.898351	0.982629

By far the LDA is giving the best accuracy and choosing K value will be safe side to not to increase the accuracy or overfitting.



Orange is training and green is testing

****

Kernal Density:

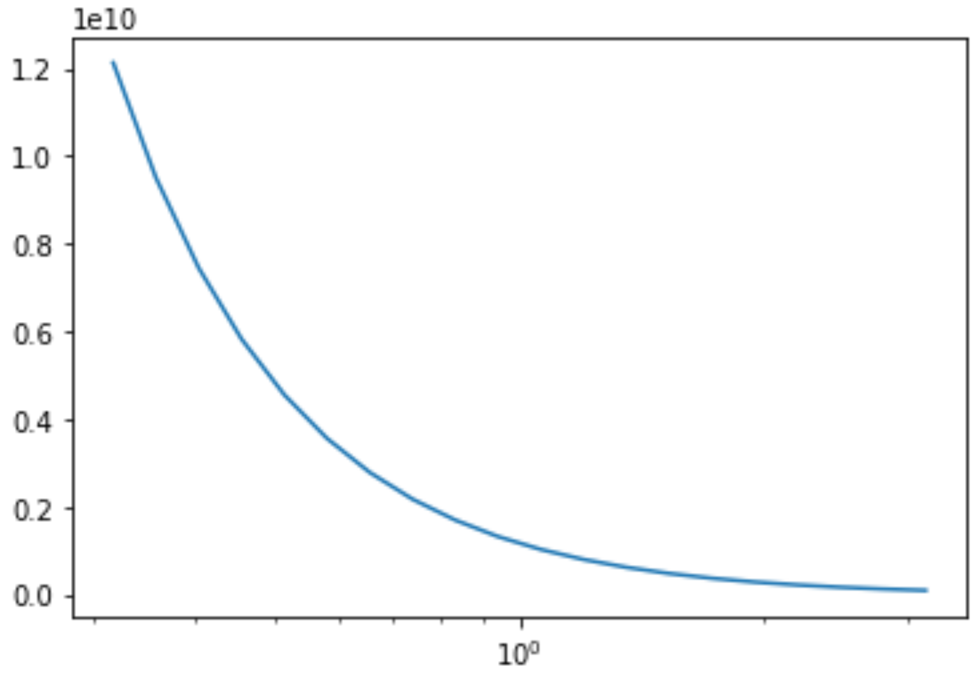
FOR PCA:

```
GridSearchCV(cv=5, error_score='raise',
       estimator=KernelDensity(algorithm='auto', atol=0, bandwidth=1.0,
breadth_first=True,
       kernel='gaussian', leaf_size=40, metric='euclidean',
       metric_params=None, rtol=0),
       fit_params=None, iid=True, n_jobs=1,
       param_grid={'bandwidth': array([0.31623, 0.35697, 0.40296, 0.45488, 0.51348,
0.57964, 0.65432,
       0.73862, 0.83378, 0.9412 , 1.06247, 1.19935, 1.35388, 1.52831,
       1.72521, 1.94748, 2.19839, 2.48163, 2.80136, 3.16228])},
       pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
       scoring=None, verbose=0)
```

Best Bandwidth is

```
KernelDensity(algorithm='auto', atol=0, bandwidth=3.1622776601683795,
       breadth_first=True, kernel='gaussian', leaf_size=40,
metric='euclidean', metric_params=None, rtol=0)
```

The plot between bandwidth and accuracy looks like this.
FOR pea



FOR LDA: