1. K-Nearest-Neighbors and Naive Bayes

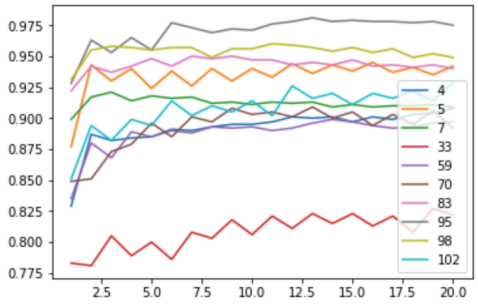
a. Classification with kNN

Finding the value of k for each label was done by sweeping across all labels we were concerned with and all values of k from 1 to 20. Training (on the train set) and predicted (on the validation set) based on each label and then scoring the predictions. This yielded the matrix shown here:

	4	5	7	33	59	70	83	95	98	102
1	0.829	0.877	0.899	0.783	0.835	0.849	0.922	0.928	0.931	0.851
2	0.887	0.943	0.917	0.781	0.88	0.851	0.942	0.963	0.955	0.894
3	0.882	0.93	0.921	0.805	0.868	0.873	0.937	0.953	0.958	0.882
4	0.884	0.94	0.914	0.789	0.889	0.879	0.942	0.965	0.957	0.899
5	0.885	0.924	0.918	0.8	0.885	0.896	0.948	0.955	0.955	0.894
6	0.891	0.938	0.916	0.786	0.89	0.885	0.942	0.977	0.957	0.914
7	0.89	0.926	0.917	0.808	0.888	0.901	0.95	0.973	0.957	0.902
8	0.893	0.94	0.912	0.803	0.893	0.897	0.948	0.969	0.949	0.91
9	0.895	0.93	0.913	0.818	0.892	0.908	0.95	0.972	0.956	0.905
10	0.895	0.94	0.911	0.806	0.893	0.903	0.947	0.971	0.956	0.914
11	0.897	0.933	0.913	0.821	0.89	0.905	0.947	0.976	0.96	0.902
12	0.901	0.944	0.912	0.811	0.892	0.901	0.943	0.978	0.959	0.926
13	0.9	0.936	0.913	0.823	0.896	0.909	0.945	0.981	0.957	0.916
14	0.901	0.943	0.909	0.815	0.899	0.9	0.943	0.978	0.954	0.92
15	0.897	0.938	0.911	0.823	0.897	0.905	0.947	0.979	0.957	0.911
16	0.901	0.945	0.909	0.813	0.894	0.894	0.942	0.978	0.953	0.92
17	0.899	0.937	0.91	0.821	0.892	0.903	0.943	0.978	0.956	0.916
18	0.903	0.941	0.91	0.808	0.893	0.895	0.941	0.977	0.949	0.922
19	0.904	0.935	0.911	0.827	0.896	0.906	0.943	0.978	0.952	0.914
20	0.908	0.942	0.909	0.822	0.897	0.892	0.94	0.975	0.949	0.929

This matrix shows the score (0 – 1 with 0 being all predictions are wrong and 1 being all predictions are correct) of each label with each value of k. (the rows represent values of k and the columns represent the id of the labels).

For each label, a graph of the score vs the value of k was then produced using df.plot().



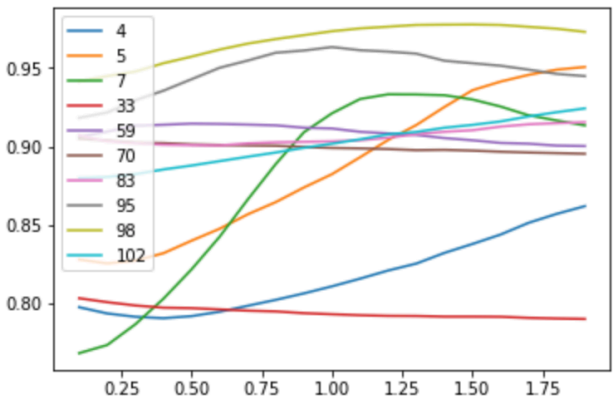
From this graph and table, an appropriate value of k was found for each class. These were then recorded in the 'ks.txt' file.

b. Classification with NB

For the Nayve Bayse Bernoulli Classification, the parameter alpha was scanned for. Performing a similar test as in the kNN Classification. All labels were scanned across for values of alpha ranging from 0.1 to 1.9 in increments of 0.1. The compiled scores for each model are found in the table below:

	0.1	0.2	0.30000000000000004	0.4	0.5	0.6	0.7000000000000001	0.8	0.9	1.0
4	0.79758	0.793579	0.791479	0.790579	0.791879	0.794679	0.79848	0.80228	0.806381	0.810881
5	0.827983	0.825483	0.827283	0.831983	0.839984	0.847585	0.856786	0.864586	0.873887	0.882388
7	0.768277	0.773477	0.786679	0.80298	0.821682	0.842484	0.866187	0.888889	0.909091	0.921192
33	0.80328	0.80078	0.79868	0.79728	0.79698	0.79618	0.79538	0.794879	0.793779	0.793079
59	0.906091	0.909491	0.913291	0.913891	0.914591	0.914391	0.913991	0.913491	0.911991	0.911491
70	0.905091	0.90389	0.90249	0.90219	0.90139	0.90099	0.90069	0.90049	0.89969	0.89909
83	0.907191	0.90329	0.90199	0.90119	0.90089	0.90069	0.90199	0.90269	0.90299	0.90329
95	0.918292	0.921692	0.929493	0.935594	0.942894	0.950195	0.954795	0.959796	0.961196	0.963396
98	0.941594	0.944894	0.947795	0.953095	0.957296	0.961796	0.965597	0.968497	0.970997	0.973497
102	0.879888	0.880588	0.882388	0.885389	0.887889	0.890689	0.893489	0.89619	0.89919	0.90179

1.1	1.2000000000000002	1.3000000000000003	1.4000000000000001	1.5000000000000002	1.6	1.7000000000000002	1.8000000000000003	1.9000000000000001
0.815782	0.820982	0.825383	0.832183	0.837884	0.843884	0.851385	0.857086	0.861986
0.893189	0.90419	0.913691	0.925193	0.935894	0.941394	0.945695	0.948995	0.950895
0.930293	0.933293	0.933193	0.932693	0.929993	0.925493	0.919892	0.916592	0.913491
0.792479	0.792079	0.791979	0.791579	0.791579	0.791479	0.790779	0.790379	0.790079
0.909391	0.908391	0.907391	0.905391	0.90399	0.90229	0.90179	0.90059	0.90029
0.89879	0.89839	0.89769	0.89789	0.89739	0.89669	0.89619	0.89579	0.89529
0.90399	0.905591	0.907891	0.909391	0.910391	0.912691	0.914191	0.914891	0.915492
0.961396	0.960496	0.959196	0.954695	0.953095	0.951495	0.948995	0.946295	0.944894
0.975298	0.976398	0.977498	0.977698	0.977798	0.977498	0.976298	0.975098	0.973097
0.90459	0.907791	0.909291	0.911991	0.913791	0.915992	0.919192	0.921892	0.924292



This graph shows the relationship between the score as the value of alpha goes from 0.1->1.9. This clearly shows that there can be found a “best value” of alpha at roughly 1.25. This is therefore what was taken for the default value of alpha in the Naïve Bayes Bernoulli classifier.

c. Data Analysis

Based on the parameter estimations done in the previous two sections it can be clearly seen that the label with id 33 is the hardest to classify with both classifiers. This can be because the data points with label 33 are somewhat similar to other classes. (ie class 33 objects have more of the other classes in common).

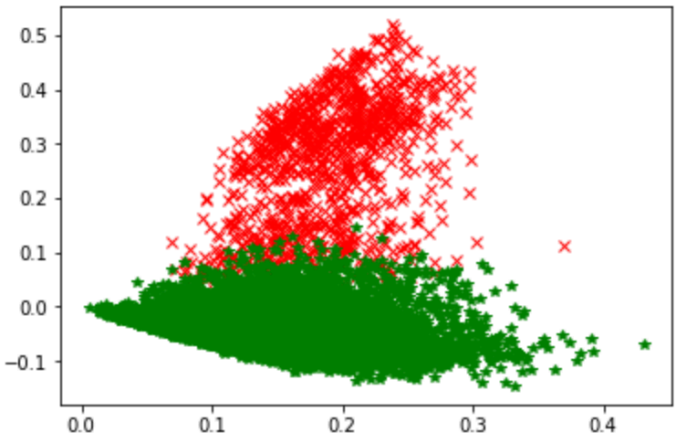
2. KMeans

a. Classification with k-Means

The KMeans classification was performed by taking the training data (when the train method is called), computing the centroid for the training set corresponding to the label and for the training set not corresponding to the label. This was done by fitting a new kmeans on each of the 2 data sets (true label and not label). These two centroids are then used as the centroids for the trained kmeans model.

b. Data Exploration for label = 33

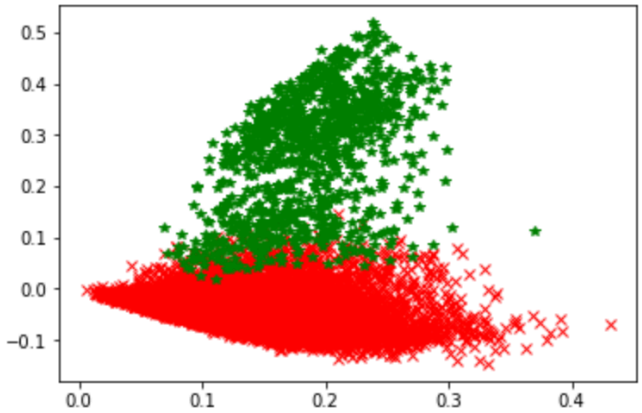
Using the Truncated SVD dimensionality reduction technique integrated in scikit-learn, a plot of the dimmensionaly-reduced data for the classification of label 33 is shown below:



In this graph, the green stars are not classified as the labels and the red xs are classified as the label.

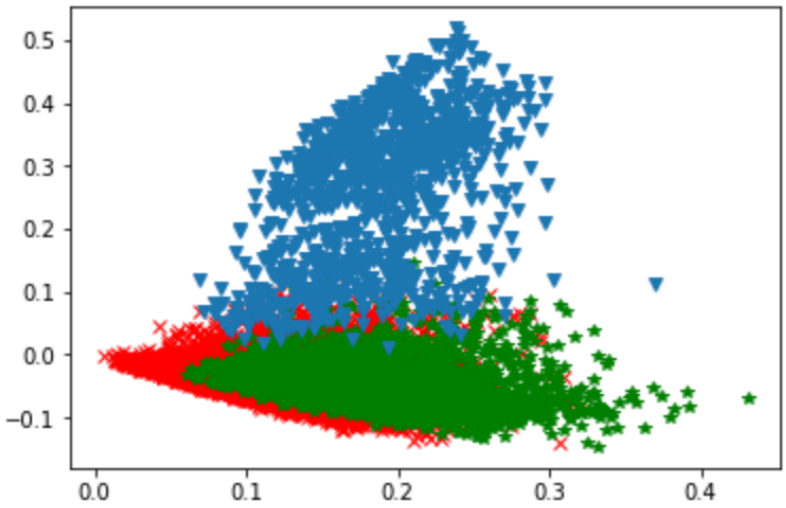
c. Data Exploration with random K-Means centroids

The following graph shows a random k-means centroid. It is transformed to be in the same coordinate space as the graph in part b.



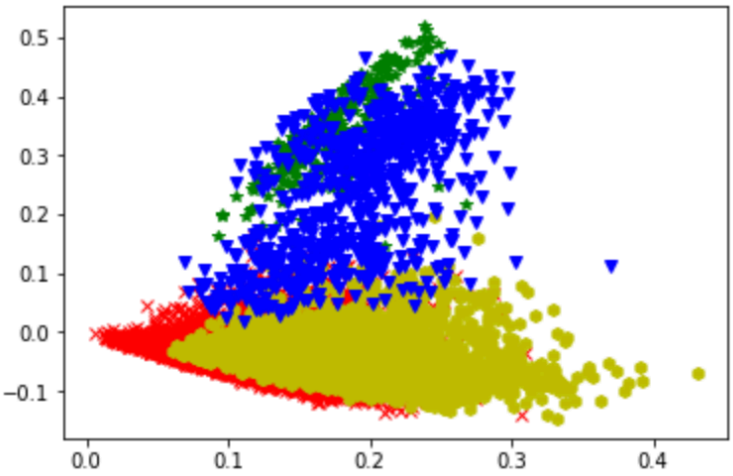
This graph (above) shows that, with 2 clusters and random centroids, we have very similar clusters, however, the ones that start with the defined centroids are more separated than the one with random centroids.

The following graph shows 3 cluster centers with random centroid selection.



This 3-cluster graph shows similarity to the 2-cluster graph. However there is more overlap then in the 2-cluster graphs. The red cluster is also the edge of the green cluster in this case. This shows that the dimensionality reduction used to separate label 33 from the others does actually work because the 2 main components that are used do work for differentiating 2 classes from 3.

The following graph shows 4 cluster centers with random centroid selection.



This 4-cluster graph, once again shows that the dimensionality reductions works fairly well as the blue class (true label) is fairly concentrated and apart from the others. The green

class however, shows that there may be a class that has fairly strong commonalities with the the data associated with label 33.