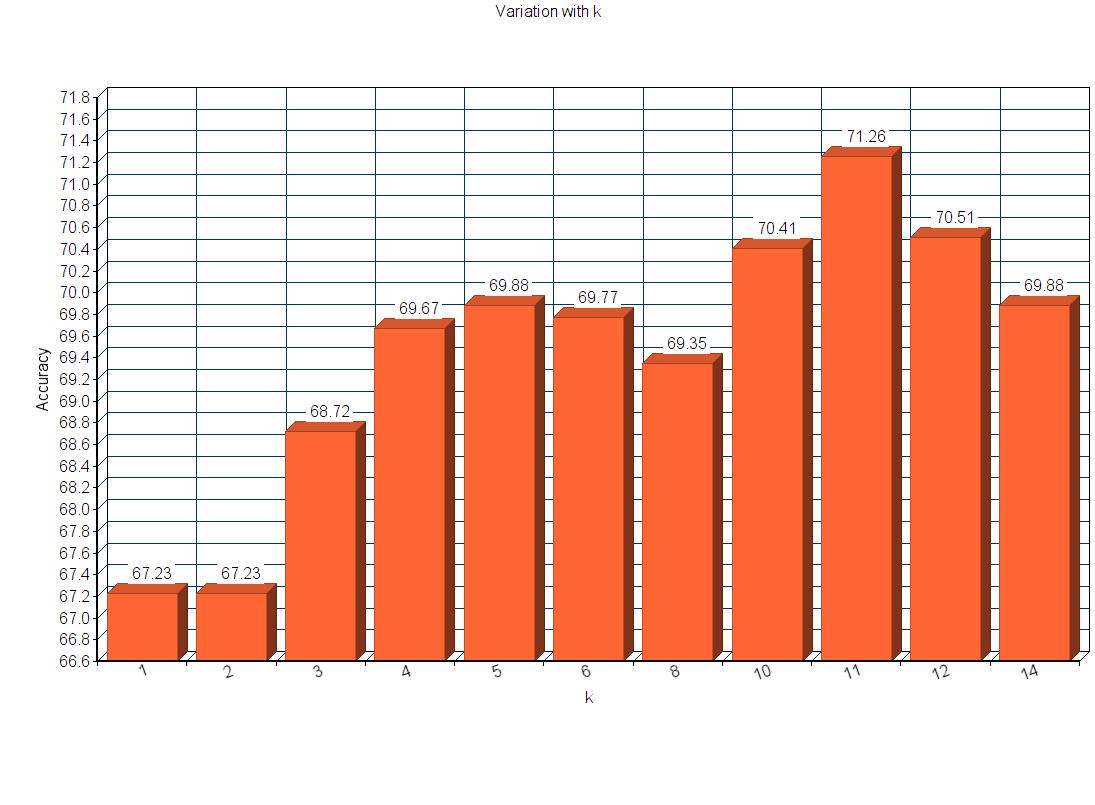
**Report:**

**K-Nearest Neighbour:**

When we train our model using K-Nearest Neighbour, we observe that, the accuracy increases with the value of ‘k’ until k = 11 and then decreases gradually. The maximum accuracy at k = 11 is 71.26%. Following are the values of accuracy for the respective values of ‘k’:

|  |  |
| --- | --- |
| **K** | **Accuracy (in percentage)** |
| 1 | 67.23 |
| 2 | 67.23 |
| 3 | 68.72 |
| 4 | 69.67 |
| 5 | 69.88 |
| 6 | 69.77 |
| 8 | 69.35 |
| 10 | 70.41 |
| 11 | 71.26 |
| 12 | 70.51 |
| 14 | 69.88 |



**AdaBoost:**

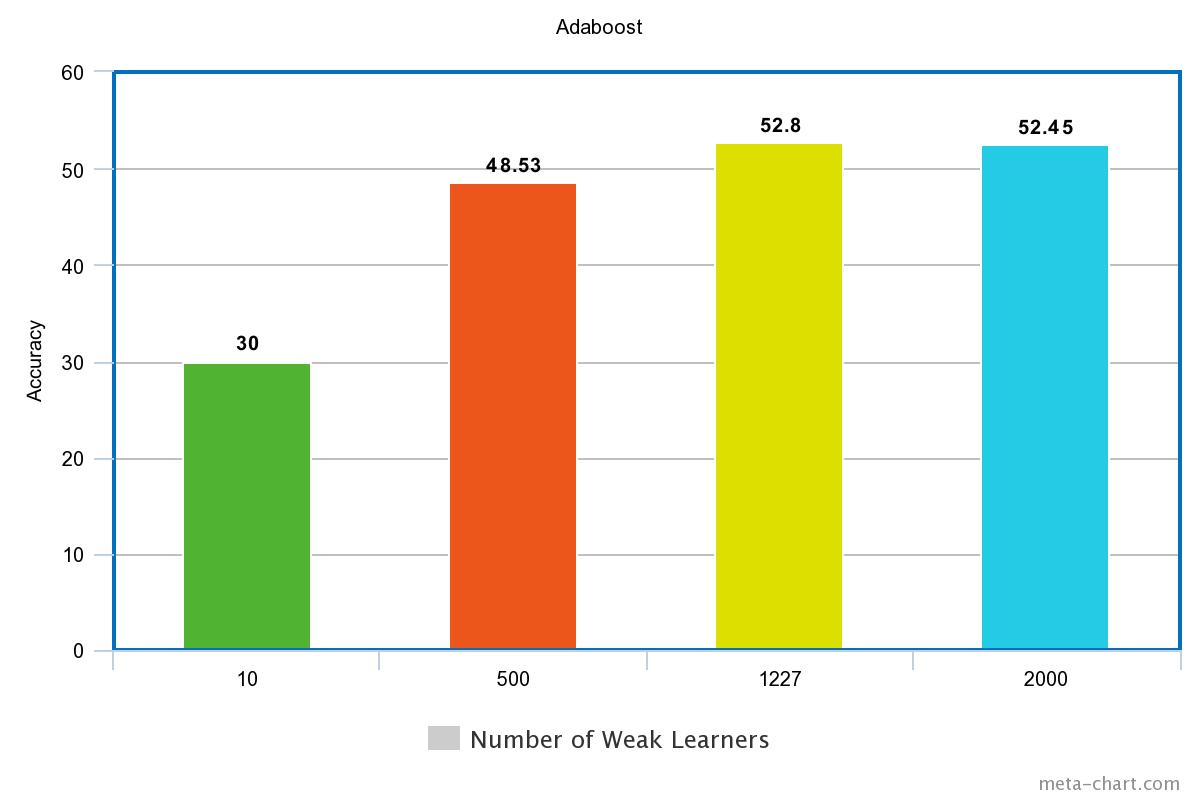
Here, we generate random pairs of column indices and subtract the pixel values at those columns together to get our hypotheses. We created 4 separate classifiers for each orientation (one vs all classifiers). If the value after subtraction is greater than 0, we label it as positive class. Otherwise, the label gets assigned as negative class.

Since the column indices are generated randomly, accuracy varies with each run even for same number of weak learners.

Below is a table with observations when we ran AdaBoost with varying number of weak learners. It was found that despite increasing the number of weak learners significantly beyond 1227, accuracy was not increasing by much. Also, the training time for a greater number of learners was too high as compared to the reward of accuracy.

We would recommend 1227 learners for this model

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.No | #.of Weak learners | Training time | Accuracy |
| 1 | 10 | 2 sec | 30.00% |
| 2 | 500 | 54 sec | 48.53% |
| 3 | 1227 | 6 min 3 sec | 52.80% |
| 4 | 2000 | 9 min 25 sec | 52.45% |



**Decision Forests:**

We are using “Gini Impurity” and information gain to find the best split across various features. We selecting random 100 features out of the total 192 features and building a decision tree of depth 6. This way we are generating 10 decision trees and then taking a vote of all these trees to classify an image.

We tried with various combinations of depth, number of features and number of decision trees and it resulted the below accuracies.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | No. of Decision trees | Depth | No. of features | Threshold | Accuracy | Train Time |
| 1 | 5 | 6 | 100 | 127 | 63.83% | 41s |
| 2 | 10 | 6 | 100 | 127 | 67.33% | 1m 15s |
| 3 | 10 | 6 | 150 | 127 | 63.72% | 2m 7s |
| 4 | 15 | 6 | 100 | 127 | 65.95% | 1m 53s |
| 5 | 10 | 6 | 80 | 127 | 63.73% | 1m |
| 6 | 10 | 4 | 100 | 127 | 63.79% | 2m |

**Best Model:**

Our best model so far has been KNN with accuracy of 71.26 with k=11