Programming assignment 2

Name: Zhuomin Li ID: 502313

Compare, analyze, and select a classification model for identifying letters in various fonts

1. **Introdcution**

**1.1 General assignment description**

This assignment is asking us to compare, analyze, and select a classification model for identifying letters in various fonts. Specifically, the relevant information shows that the objective is to identify each of a large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet. It will be intuitive to train different model to analyze a same dataset, and compare those methods from different angles. Indeed, there are severals factors could be considered as key factor for determining a classifier as the “best”. In my opinion, the accuracy of a classifier should be recognized as top priority. Then computational complexity, validation accuracy, and model interpretability might be considered as well. The motivation for me is that I used try to use YOLO model from deep neural network model for front identification, this time I can try different model to do the same prediction, and I am really thrilled.

**1.2 Specific assignmet description**

In this simplified question, we are mainingly focus on three pairs: H and K, M and Y, and L and Z. The reason is that L an Z is the first letter of my name, and more importantly, they are sharing underline, but have different upper structure. From my perspective, I predict H and K will be the most diffcult one to classify, and then M and Y or L and Z maybe at the same level. The standard of my prediction is that how many structures they are sharing, the distributed pixels of H and K make it become the hardest one to classify .

**1.3 Explanation for relevant questions**  
**1.3.1** The explanation for why dimension reduction is useful for this problem is that:

1) Some features have huge part of missing values, which cannot provide intuitive information with trainer

2) Low variance. Given that the number of instance is 20000, and number of Attributes is 17(Letter category and 16 numeric features), we can find that all integers are ranged from 0 to 15 after observation. Therefore, there is low variance for some features between each letter.

3) Domain expert judgement. From the ReadMe file, we know that there are 16 numeric features, but some of the features might or might not make sense for the model. For example, the feature width of box and height of box might less intuitive if all letters are place in planned box to present. Therefore, those factors make dimension reduction crucial.

**1.3.2** “For which methods are “better”, and what factors should be considered in determining a dimension reduction method as “good” or “bad”, we might need to figure out the variance between filter, wrapper, and embedded methods for feature selection.

1. Wrapper methods will create models with subsets of features, and select the features with the best model, and it measures the “usefulness” of features based on the classifier performance. It is optimizing the classifier performance, but with expensive computation.
2. The filter methods pick up the intrinsic properties of the features (i.e., the “relevance” of the features) measured via univariate statistics instead of cross-validation performance. For example, unsupervised approach keep informative or predictive pairs of features. In contrast, supervised approach will throw off a feature which is not predictive
3. The third class, embedded methods, are quite similar to wrapper methods since they are also used to optimize the objective function or performance of a learning algorithm or model.

It seems optimize model within model for each round.”1

**1.3.3** For what factors should be considered in determining a dimension reduction method as “good” or “bad”, we should have some conclusions of good dimension reduction method:

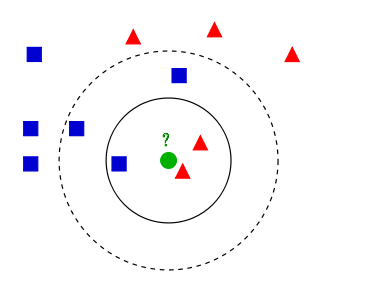
1. “Efficient. A good dimension reduction method always helps in compressing data, reducing the storage space required and shrinking time required for performing same computations. with a correct answer
2. Simplified model.It takes care of multi-collinearity that improves the model performance. It removes redundant features. For example: there is no point in storing a value in two different units (meters and inches).
3. Chaos reduction. It is helpful in noise removal also and as result of that we can improve the performance of models.”2

**2.Result**

**2.1 K-nearest neighbors model**

Description of the classifier and its advantages and disadvantages:

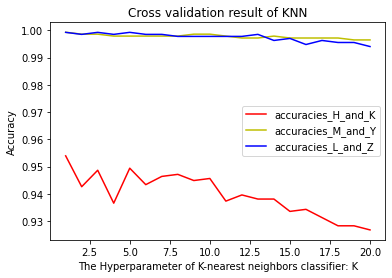
“K-nearest neighbor algorithm, could be explained as given a training data set, for a new input instance, find the K instances in the training data set that are closest to the instance, and the majority of these K instances belong to a certain class, then classify the input instance into this class. ”3



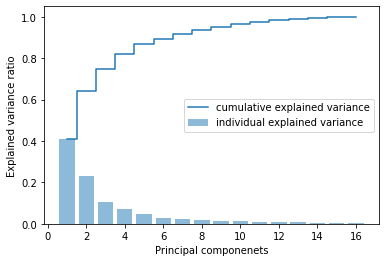
Pros: Very straightforward, sensitive for outliers, handy hyperparameter with precise model

Cons: Not suitable for huge amount of data, which make model confusion. Besides, it is time comsuming and extra space needed.

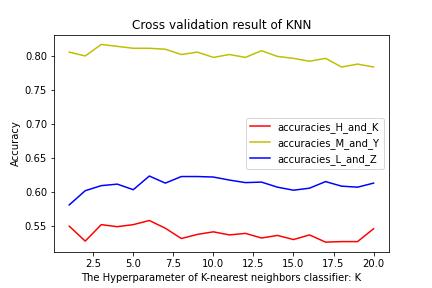
Hyperparameter: K



For the dimension reduction method, I used variance explained to shrink 16 features to 4 features. The term variance refers to a statistical measurement of the spread between numbers in a data set. More specifically, variance measures how far each number in the set is from the mean and thus from every other number in the set.13 The plot is:



Therefore, top 4th features take up 70% explained variance ratio, I will use them for training and testing data.



**2.2 Decision tree model4**

Description of the classifier and its advantages and disadvantages:

It’s one of the embedded methods. The decision tree algorithm uses a tree-like structure and uses layers of inference to achieve the final classification. The decision tree is composed of the following elements：

Root node: contains the full set of samples

Internal nodes: corresponding to feature attribute tests

Leaf nodes: represent the result of the decision

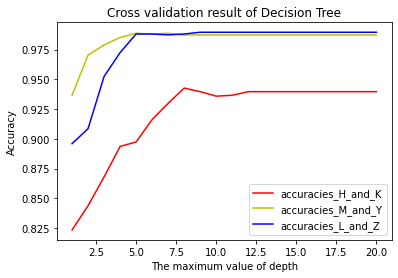
Pros:

* Decision trees are easy to understand and interpret, can be analyzed visually, and rules can be easily extracted.
* can handle both categorical and numerical data.
* being more suitable for handling samples with missing attributes.
* the ability to handle uncorrelated features.

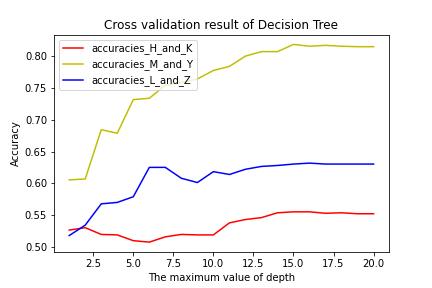
Cons:

* Prone to over-fitting
* Tend to ignore the interconnectedness of attributes in the dataset

Decision tree has some hyperparameters like: criterion, max depth and min\_samples\_split; fot this project, I choose max depth



Description of the dimension reduction methods used: I used variance explained to shrink 16 features to 4 features



**2.3 Random Forest model5**

Description of the classifier and its advantages and disadvantages:

A random forest is composed of many decision trees, and different decision trees are not associated with each other.

When we perform the classification task, new input samples enter and let each decision tree in the forest judge and classify them separately, each decision tree will get a classification result of its own, and which one of the classification results of the decision tree has the most classifications, then the random forest will take this result as the final result.

How can we build a random forest model? It could be created as followed:

1. Create a bootstrapped dataset
2. Only considering a random subset of variables at each step
3. repeat step 1 and step2, each node in the decision tree is split according to step 2 (it is easy to understand that if the next attribute selected by the node is the same attribute that was used when the parent node split, then the node has already reached the leaf node and does not need to split anymore). It continues until it is no longer possible to split. Note that no pruning is done during the entire decision tree formation process.
4. A large number of decision trees are created, and this constitutes a random forest.

Pros:

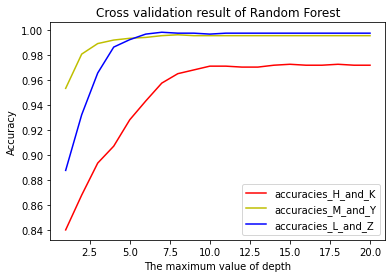
* It can come out with very high dimensional (many features) data, and without dimensionality reduction, no need to do feature selection
* It can judge the importance of features
* It can determine the interaction between different features
* It is not easy to over-fit
* The training speed is faster and it is easy to make parallel method
* It is relatively simple to implement
* It can balance the error for unbalanced data sets.
* If a large portion of the features are missing, the accuracy can still be maintained.

Cons:

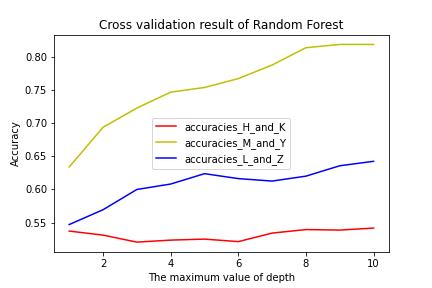
* Random forests have been shown to overfit on certain noisy classification or regression problems.
* For data with attributes that have different values, attributes with more divided values will have a greater impact on the random forest, so the random forest outputs on such data are not credible for attribute weights

Randomforest has some hyperparameters: n\_estimators, criterion, max\_depth, and for this project I choose max\_depth, and the max max\_depth is range from 1 to 20

min\_samples\_leaf



Description of the dimension reduction methods used: Use the most common four features for training new model, and do the final validation:



**2.4 SVM model6**

Description of the classifier and its advantages and disadvantages:

Support vector machines (SVMs) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. the SVM model is to represent examples as points in a space, mapped so that the examples of separate categories are divided by as wide a clear gap as possible. The new examples are then mapped to the same space and predicted to belong to a category based on which edge they fall on.

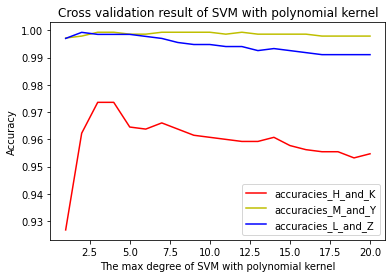
Pros7:

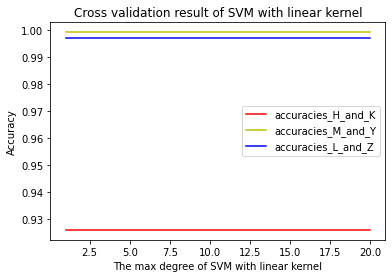
* the possibility of solving high-dimensional problems, i.e., large feature spaces.
* solving machine learning problems with small samples.
* Being able to deal with the interaction of nonlinear features.

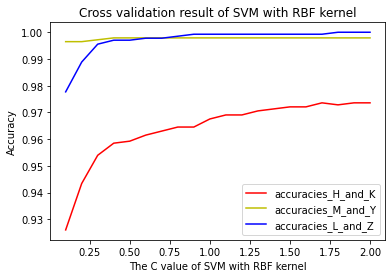
Cons:

* Not very efficient when the observed sample is large.
* There is no general solution for nonlinear problems and sometimes it is difficult to find a suitable kernel function.
* Sensitivity to missing data.

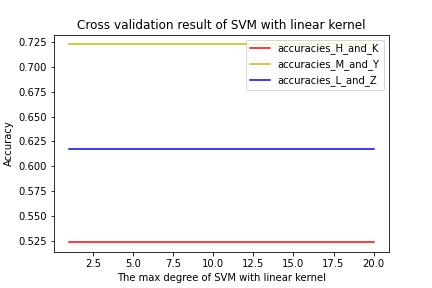
Random forest has three hyperparameters: c value, degree, gamma; I choose degree (from 1 to 20) for different kernel SVM model.

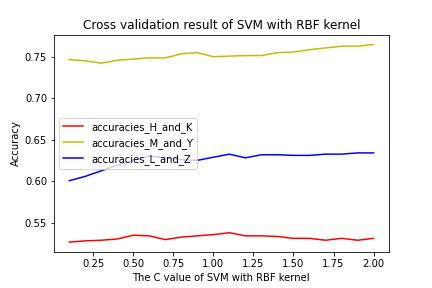






Description of the dimension reduction methods used: For the dimension reduction method, I used variance explained to shrink 16 features to 4 features:





**2.5 Artificial Neural Network model8**

Description of the classifier and its advantages and disadvantages:

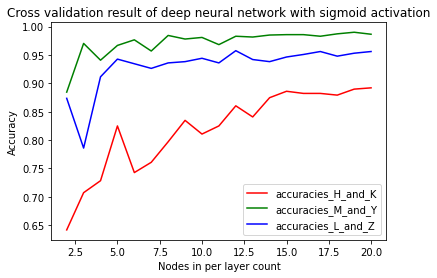
A neural network is an operational model that consists of a large number of nodes (or neurons) interconnected with each other. Each node represents a specific output function, called the activation function. Each connection between two nodes represents a weighted value for the signal passing through the connection, called a weight, which is equivalent to the memory of an artificial neural network. The output of the network varies depending on the connection method, the weight value and the activation function. The network itself is usually an approximation of some algorithm or function in nature, or it may be an expression of a logical strategy.

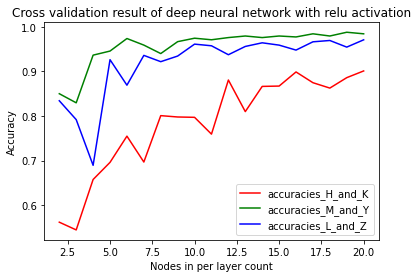
Pro:

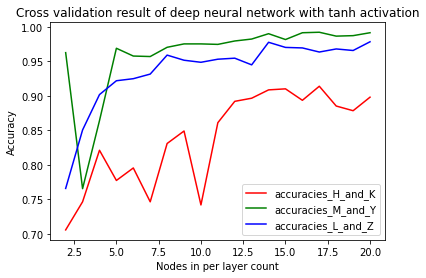
* Performs extremely well on tasks in speech, semantics, vision, and various games (e.g., Go)
* Algorithms can be quickly adapted to new problems

Cons:

* Requires large amounts of data for training
* Training requires high hardware configuration
* The model is in a "black box" state, making it difficult to understand the internal mechanisms
* Metaparameter and network topology selection is difficult.12

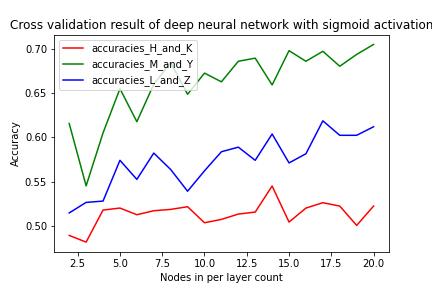


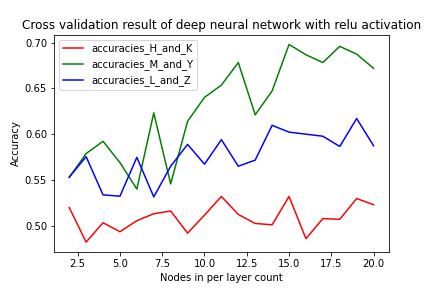


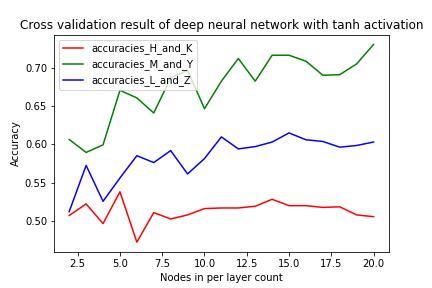


Description of the dimension reduction methods used: For the dimension reduction method, I used variance explained to shrink 16 features to 4 features. Besides, I used PCA to dimension reduction for NN with Sigmoid activation function. Principal component analysis (PCA) is the process of computing the principal components and using them to perform a change of basis on the data, sometimes using only the first few principal components and ignoring the rest. 14

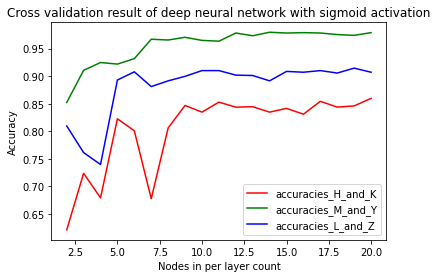
For the former method:







For PCA:



**Bonus:**

2.6 AdaBoost classifier

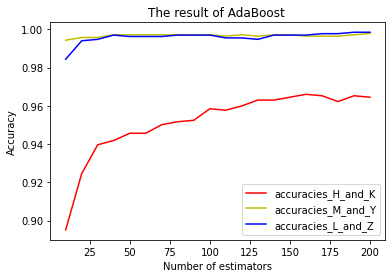
Description of the classifier and its advantages and disadvantages:

An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.[9]

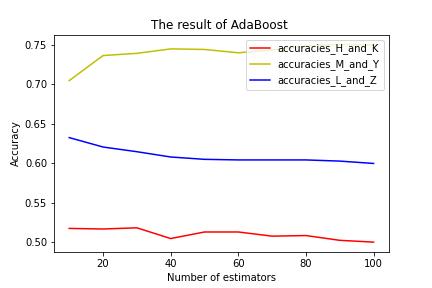
Pros:Adaboost is less prone to overfitting as the input parameters are not jointly optimized. The accuracy of weak classifiers can be improved by using Adaboost.

Cons: it needs a quality dataset. Noisy data and outliers have to be avoided before adopting an Adaboost algorithm.[10]

For hyperparameter, I choose number of estimators:



Description of the dimension reduction methods used: For the dimension reduction method, I used variance explained to shrink 16 features to 4 features



2.7 LogisticRegression Classifier with L1 regularization and LogisticRegression Classifier with L2 regularization

Logistic Regression is a machine learning method for solving binary (0 or 1) problems to estimate the likelihood of something. For example, the probability of a user buying a certain product, the probability of a patient suffering from a certain disease, or the probability of an advertisement being clicked by a user. Note that the term "likelihood" is used here, not "probability" in mathematics. The result of logisitc regression is not a probability value in the mathematical definition, and cannot be used directly as a probability value. The results are often weighted and summed with other eigenvalues, rather than multiplied directly.[11]

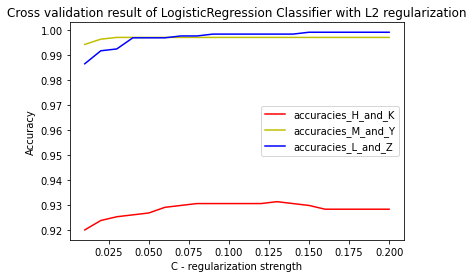
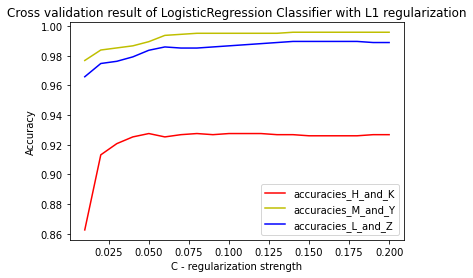
Pros:

* simplicity of implementation and wide application to industrial problems.
* very low computational effort for classification, high speed and low storage resources.
* Convenient observation of sample probability scores.
* multicollinearity is not a problem for logistic regression, which can be combined with L2 regularization to solve the problem.
* computationally inexpensive and easy to understand and implement.

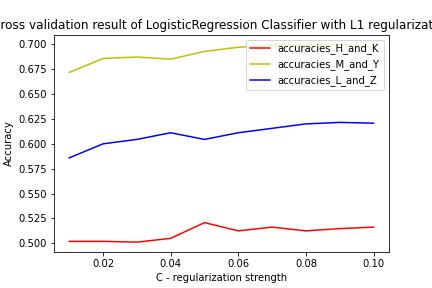
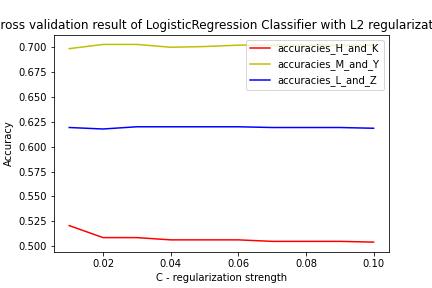
Cons:

* Logistic regression does not perform very well when the feature space is large.
* it prone to underfitting and generally not very accurate
* it cannot handle a large number of multi-class features or variables very well.
* it can only handle two classification problems (softmax derived from this can be used for multi-classification) and must be linearly separable.
* it requires transformation for non-linear features.

The hyperparater I choose is C value for model with L1 regularization and L2 regularization



For the dimension reduction method, I used variance explained to shrink 16 features to 4 features

**3.Discussion**

**3.1 Compare the performance and run time of the different classifiers on the final validation sets with either a table or a figure.**

For pair H and K

|  |  |  |  |
| --- | --- | --- | --- |
| name | model training time | model operation time | validation\_accuracy |
| KNN | 0.996112823 | 12.95733452 | 0.966216216 |
| Decision Tree | 4.982709885 | 0 | 0.972972973 |
| Random Forest | 355.8168411 | 18.44835281 | 0.986486486 |
| SVM with Polynomial kernal | 29.89792824 | 0.997781754 | 0.97972973 |
| SVM with linear kernal | 44.19016838 | 2.030611038 | 0.945945946 |
| SVM with RBF kernal | 21.89159393 | 6.067276001 | 0.972972973 |
| AdaBoost | 579.9789429 | 77.74019241 | 0.939189189 |
| LogisticRegression Classifier with L1 regularization | 37.8742218 | 0.999689102 | 0.952702703 |
| LogisticRegression Classifier with L2 regularization | 72.75366783 | 0 | 0.945945946 |
| sigmoid\_model\_HK | 2251.956701 | 195.5041885 | 0.959459484 |
| relu\_model\_HK | 2118.168116 | 191.2839413 | 0.932432413 |
| tanh\_model\_HK | 2210.450649 | 190.1471615 | 0.979729712 |

For pair M and Y

|  |  |  |  |
| --- | --- | --- | --- |
| name | model training time | model operation time | validation\_accuracy |
| KNN | 0 | 7.973909378 | 1 |
| Decision Tree | 1.993179321 | 1.000642776 | 0.98 |
| Random Forest | 249.1500378 | 17.9412365 | 1 |
| SVM with Polynomial kernal | 5.979537964 | 0.997543335 | 1 |
| SVM with linear kernal | 6.975650787 | 0 | 1 |
| SVM with RBF kernal | 7.980585098 | 1.992940903 | 1 |
| AdaBoost | 541.2352085 | 39.81900215 | 0.993333333 |
| LogisticRegression Classifier with L1 regularization | 23.9200592 | 0 | 0.98 |
| LogisticRegression Classifier with L2 regularization | 31.89539909 | 0 | 0.993333333 |
| sigmoid\_model\_LZ | 2058.696747 | 176.2385368 | 0.973333359 |
| relu\_model\_LZ | 2166.791439 | 175.9791374 | 0.980000019 |
| tanh\_model\_LZ | 2159.459114 | 3330.012798 | 0.99333334 |

For pair L and Z:

|  |  |  |  |
| --- | --- | --- | --- |
| name | model training time | model operation time | validation\_accuracy |
| KNN | 0.996589661 | 11.96050644 | 1 |
| Decision Tree | 3.985643387 | 0 | 0.962025316 |
| Random Forest | 383.7223053 | 20.92909813 | 0.993670886 |
| SVM with Polynomial kernal | 8.973121643 | 0.997304916 | 1 |
| SVM with linear kernal | 5.981206894 | 0 | 0.993670886 |
| SVM with RBF kernal | 9.959936142 | 1.992940903 | 0.993670886 |
| AdaBoost | 549.1683483 | 39.86692429 | 0.993670886 |
| LogisticRegression Classifier with L1 regularization | 17.93956757 | 0 | 0.987341772 |
| LogisticRegression Classifier with L2 regularization | 26.91054344 | 0 | 0.993670886 |
| sigmoid\_model\_MY | 2162.162304 | 185.7979298 | 0.987341762 |
| relu\_model\_MY | 2116.169691 | 186.2127781 | 0.993670881 |
| tanh\_model\_MY | 2200.676203 | 196.9976425 | 0.987341762 |

3.2 Compare the performance and run time of the different classifiers after dimension reduction on the final validation sets with either a table or a figure.

For pair H and K

|  |  |  |  |
| --- | --- | --- | --- |
| name | model training time | model operation time | validation\_accuracy |
| KNN | 2.029180527 | 6.002426147 | 0.540540541 |
| Decision Tree | 0.938177109 | 0.997781754 | 0.554054054 |
| Random Forest | 176.4581203 | 10.96463203 | 0.506756757 |
| SVM with Polynomial kernal | 159.4631672 | 4.994392395 | 0.513513514 |
| SVM with linear kernal | 78.73272896 | 4.975557327 | 0.540540541 |
| SVM with RBF kernal | 73.71068001 | 17.01903343 | 0.567567568 |
| AdaBoost | 215.2593136 | 16.93797112 | 0.506756757 |
| LogisticRegression Classifier with L1 regularization | 4.010915756 | 0 | 0.540540541 |
| LogisticRegression Classifier with L2 regularization | 6.903409958 | 0 | 0.513513514 |
| sigmoid\_model\_HK | 1180.747032 | 97.76329994 | 0.533783793 |
| relu\_model\_HK | 1098.54722 | 105.4255962 | 0.554054081 |
| tanh\_model\_HK | 1128.396988 | 100.9693146 | 0.445945948 |

For M and Y:

|  |  |  |  |
| --- | --- | --- | --- |
| name | model training time | model operation time | validation\_accuracy |
| KNN | 0.997066498 | 2.989768982 | 0.816455696 |
| Decision Tree | 0.996589661 | 0.996828079 | 0.740506329 |
| Random Forest | 153.4893513 | 9.965181351 | 0.816455696 |
| SVM with Polynomial kernal | 149.5456696 | 2.990722656 | 0.810126582 |
| SVM with linear kernal | 84.71870422 | 2.988815308 | 0.689873418 |
| SVM with RBF kernal | 63.74979019 | 12.99548149 | 0.727848101 |
| AdaBoost | 291.0265923 | 20.90144157 | 0.721518987 |
| LogisticRegression Classifier with L1 regularization | 9.995937347 | 0 | 0.702531646 |
| LogisticRegression Classifier with L2 regularization | 7.976055145 | 0 | 0.702531646 |
| sigmoid\_model\_MY | 1141.195774 | 103.8358212 | 0.696202517 |
| relu\_model\_MY | 1181.042194 | 99.37286377 | 0.708860755 |
| tanh\_model\_MY | 1130.025387 | 98.67143631 | 0.696202517 |

For L and Z:

|  |  |  |  |
| --- | --- | --- | --- |
| name | model training time | model operation time | validation\_accuracy |
| KNN | 2.003192902 | 3.98850441 | 0.706666667 |
| Decision Tree | 0.995635986 | 1.008987427 | 0.673333333 |
| Random Forest | 147.4385262 | 10.08796692 | 0.673333333 |
| SVM with Polynomial kernal | 72.68023491 | 4.007816315 | 0.64 |
| SVM with linear kernal | 68.77040863 | 4.96172905 | 0.613333333 |
| SVM with RBF kernal | 68.77040863 | 16.89934731 | 0.673333333 |
| AdaBoost | 266.1571503 | 23.91982079 | 0.6 |
| LogisticRegression Classifier with L1 regularization | 5.937337875 | 0 | 0.593333333 |
| LogisticRegression Classifier with L2 regularization | 4.981517792 | 0 | 0.593333333 |
| sigmoid\_model\_LZ | 1283.233404 | 110.6102467 | 0.626666665 |
| relu\_model\_LZ | 1162.169695 | 103.1761169 | 0.653333306 |
| tanh\_model\_LZ | 1218.73188 | 190.3655529 | 0.613333344 |

3.3Lessons learned: What model would you choose for this problem and

why? How did dimension reduction effect the accuracy and/or run times of the

different classifiers? What would you do differently if you were given this same

task for a new dataset? Anything else about this project that made you think?

I will choose SVM with RBF kernal for this problem. The reason is that: it keeps high accuary among other models with decent training and testing running time, especailly for the hardest pair H and K. Dimension reduction render most of model speed up with half lower accuary as the trade off. For the question what would you do differently if you were given this same task for a new dataset. I will try more hyperparameter for the most suitable model to get better model performance. Anything else about this project that made you think? I think ANN is really time consuming for the tiny dataset!

**4.Code explanation**

For the code, please make sure you have python 3.x and its standard library. All of the file is Jupeyter book so please run it from the top to bottom.

Reference:

1. <https://sebastianraschka.com/faq/docs/feature_sele_categories.html>
2. <https://www.analyticsvidhya.com/blog/2015/07/dimension-reduction-methods/>
3. <https://zhuanlan.zhihu.com/p/25994179>
4. <https://easyai.tech/ai-definition/decision-tree/>
5. <https://easyai.tech/ai-definition/random-forest/>
6. <https://scikitlearn.org/stable/modules/svm.html#:~:text=Support%20vector%20machines%20(SVMs)%20are,than%20the%20number%20of%20samples.>
7. <https://easyai.tech/ai-definition/svm/#xiaobai>
8. <https://easyai.tech/ai-definition/ann/>
9. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html>
10. <https://www.analyticsvidhya.com/blog/2021/06/adaboost-a-brief-introduction-to-ensemble-learning/#:~:text=Advantages%20and%20disadvantages,-Coming%20to%20the&text=The%20accuracy%20of%20weak%20classifiers,it%20needs%20a%20quality%20dataset.>
11. <https://zhuanlan.zhihu.com/p/28408516>
12. <https://developer.aliyun.com/article/620031>
13. <https://www.google.com/search?q=variance+explained&rlz=1C1FKPE_enUS939US939&oq=variance+explained&aqs=chrome.0.69i59l2j0i512l5j69i60.889j0j4&sourceid=chrome&ie=UTF-8>
14. https://en.wikipedia.org/wiki/Principal\_component\_analysis#:~:text=Principal%20component%20analysis%20(PCA)%20is,components%20and%20ignoring%20the%20rest.