Final Project Report

Multi-Class Classification Analysis Based on Fashion MNIST

Name	NetID
Shuyuan Chen	sc109
Yifei Shu	yshu5
Peiwen Zhang	peiwen4

1. Project description and summary.

Fashion MNIST data is a standard method for benchmarking ML/AI algorithms. Our crucial goal in the project is not to make fine tuned deep learning models to achieve some high accuracy indicators but to work with different classification models and find a logical and feasible way to improve the model and finally use the F-MNIST dataset to verificate our idea.

- Understanding how different kinds of classification models work and which scenario is suitable for different models.
- Explain why we choose the desired models for this dataset based on the model principles and characteristics.
- Properly tune model parameters and provide some model accuracy indicators (such as classification error) to demonstrate that our inference is correct.
- Conduct an integration of different models and provide a creative point of our report.

Some important approaches are utilized in our project. First, we use statistical descriptive analysis to have a basic understanding of the data in section three. Then in section four, we utilized Principal Component Analysis and t-SNE plot as our unsupervised learning methods before conducting multiclass classification. For t-SNE, it can help us to dig out the data structure behind the original data and visualize high-dimensional data. For PCA, it seeks to maximize variance and preserves large pairwise distances. In that scenario, things that are different end up far apart, and high-dimensional data can also be decreased.

Based on the findings and insights from unsupervised learning, several multi-class classification methods such as SVM and KNN are applied to build models based on training data and to predict the classification of labels. However, predictions from multi-class classification models are limited to certain types of data structures. These models are only suitable for specific cases and perform well under certain conditions. Therefore, we utilized majority voting ensemble models to combine several classification models and design our personalized models.

During the process of exploring the datasets and designing the models, some important conclusions are proposed:

- One of the important findings of unsupervised learning is that some labels such as label 8-bag could have some sublabels. This insight provides a hint to improve our classification models in section five, using sub-labeled data to conduct model training and prediction.
- Another important finding of unsupervised learning is that the suggested clusters for this F-MNIST data is six. This tells us that some classes are very similar to each other and this feature of the dataset may lead to a low classification accuracy of some labels.
- The optimal single model on multiclass classification is SVM model of 90.68% classification accuracy on the testing data without any ensemble models. And the performance of predicted labels is basically consistent between each model. For example the accuracy of the 6^{th} label shirt are the worst in all methods.
- The ensemble classification model finally achieved 92.11% classification accuracy on testing data by using the majority voting ensemble machine learning models.

2. Literature review.

F-MINST datasets are the advanced version of MNIST datasets, it is always utilized to verify the accuracy of newly developed and proposed algorithms (benchmark). Moreover, for the AI/ML developers, even if their algorithms passed the accuracy test of F-MNIST datasets, there is no guarantee that their algorithms are efficient, that is the F-MNIST dataset is only the first step of benchmark. **Fine-Tuning DARTS** (Differential Architecture Search), which is a method of Neural Architecture Search, provides the best accuracy of 96.91% on Fashion-MNIST datasets.

Majority literatures on Fashion MNIST datasets focus on deep learning models such as CNN (Convolutional Neural Network), which is a common deep learning model for computer vision. In the first literature,[2] Hyper-Parameter Optimization (HPO) is utilized to combine with CNN method and provide the best classification accuracy of 93.99%. For example, when CNN works with image classification, the CNN method takes an input image's raw pixel data and learns how to extract key features from the image by assigning weight/importance to each feature, then it will output a classification label of this image. Therefore, in this paper, the HPO method is applied to optimize the parameters in the CNN model. This idea is also useful for our classification models, Hyper-Parameter Optimization can also be applied to improve the model accuracy of our models.

Besides, many researches are completed based on classic machine learning models such as multiclass SVM. The second literature [2] utilized Histogram of Oriented Gradient (HOG) features to extract crucial features from original image data and then use SVM classifier to make classification. Classical multiclass SVM model works well with image processing and classification, while HOG is an efficient feature descriptor which also has a fairly great performance with image data. The methods in this paper works in a fast and efficient way, HOG is used to divide the input image into small cells and the size of cells should be determined by parameter tuning using classifiers. This is when the SVM classifier will be imported, it serves as a classification tool of HOG features. Finally the accuracy results of the HOG+SVM model is 86.53%.

The classification accuracy indicators of these two methods are not the best among all emerging methods, but the methodology of these two articles are valuable and significant for our project. In the first article, the idea of applying the hyper parameter optimization method is a great idea not only for deep learning models but also for other classic classification models. While in the second method, making a combination of classic models and other optimizing methods is a good idea to both improve the performance of the ensemble model and keep the advantage of model stability of the classic model.

Based on the relevant research and the insights from these literatures, we conducted our own research on this dataset.

3. Descriptive Statistical Analysis

3.1 Summary Statistics

After checking the training dataset, we find that there are 10 labels and every label has 6000 rows. Besides, there is no missing data here.

Table 1: Frequency table of training data

Label	T-shirt	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot
Frequency	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000
Relative Frequency	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

As for the testing data, we find that there are also 10 labels and every label has 1000 rows, and no missing data either.

Tabel 2: Frequency table of testing data

Label	T-shirt	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot
Frequency	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
Relative Frequency	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

3.2 Data processing

Data normalization is an important step which ensures that each input parameter (pixel, in this case) has a similar data distribution. This makes convergence faster while training the network. For image inputs we need the pixel numbers to be positive, so we might choose to scale the normalized data in the range [0,1].

4. Unsupervised learning

4.1 t-SNE plot

The t-SNE plot on training data provided the relative distance between observations:

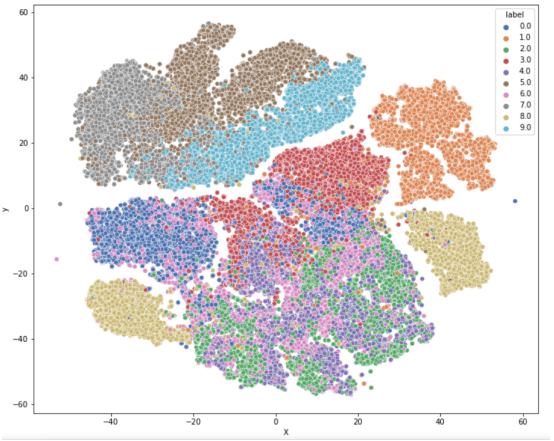


Figure 1 — T-SNE plot based on training data

The graph suggests 6 main clusters in the training dataset. What's more, we can find that some clusters have subclusters, because there are obvious gaps between them. For example, labels 0, 5 and 8 have 2 sub clusters because of the big gap.

Based on the Figure 2, we can see that some labels seem to have subclusters, like label 0, 1, 3, 8, and 9. After visualizing the centroids by K-means, we found that the difference of centroids are lightness, not shape, except for label 8, and label 3. For label 2, 4, 6, their shapes are very similar with each

other, so it is hard to class them correctly even with human eyes. That's why these labels are mixed with each other in the t-SNE plot.

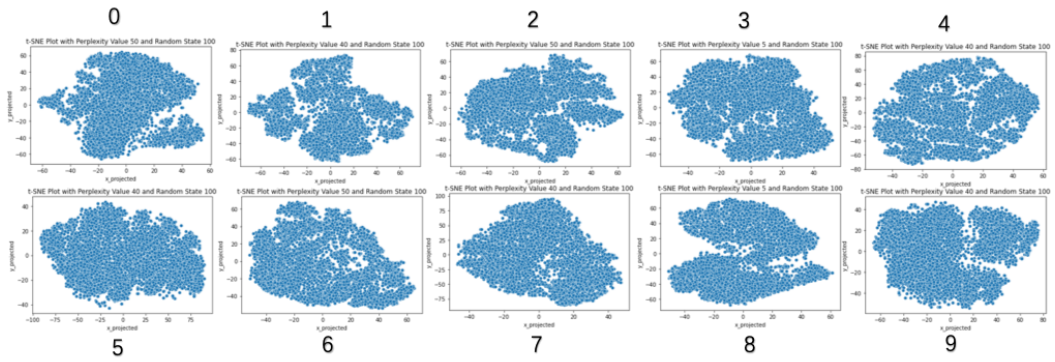


Figure 2 — T-SNE plot of every single label data

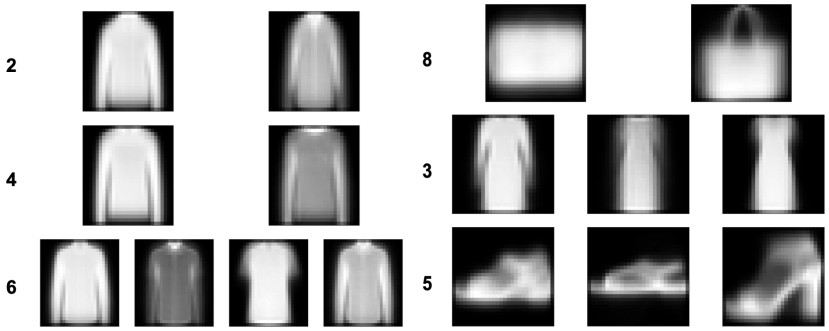


Figure 3 — Visualization of centroids of label 2, 4, 6 and 8, 3, 5

We can see that the difference between 2 subclusters of label 8 is the shape. Label 8 is a bag, and some bags have hadel, some don't. For label 5, we can see that the difference between centroids is the height of sandel's heel, some are flat, some are kitten, and some are high. For label 3, we can see that the difference between centroids is the length of the dress's sleeves, some are long, some are short, and some are none sleeves.

4.2 PCA algorithm

Based on the observation of the t-SNE plot, we applied Principal Component Analysis to have a look at the principal components of 10 label images in training dataset.

4.2.1 PCA result

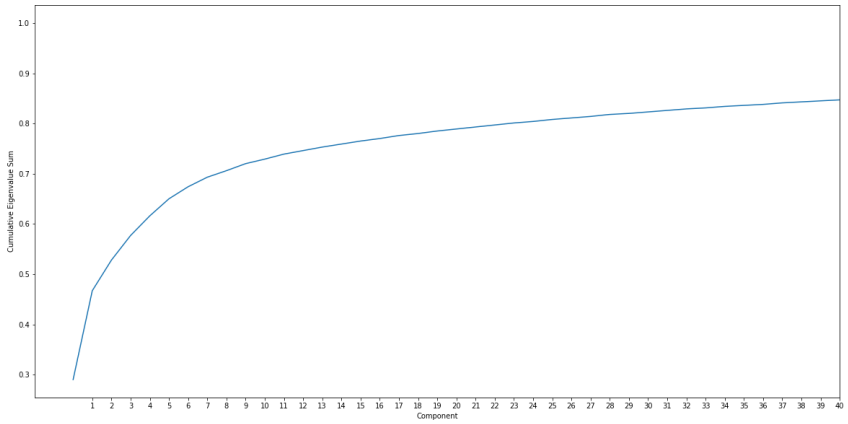


Figure 4 Cumulative Eigenvalue sum based on different number of components

Based on Figure 4, 25 components is good enough to represent the dataset with a explained variance of 80%.

4.2.2 Association between the first 2 principal components and the labels

Here is the relationship between first and second principal component boxplot with each label:

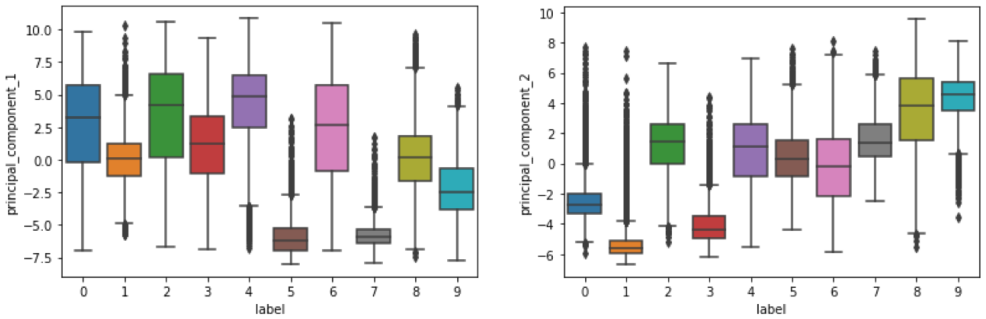


Figure5: Different label's attributes on the first 2 PCs

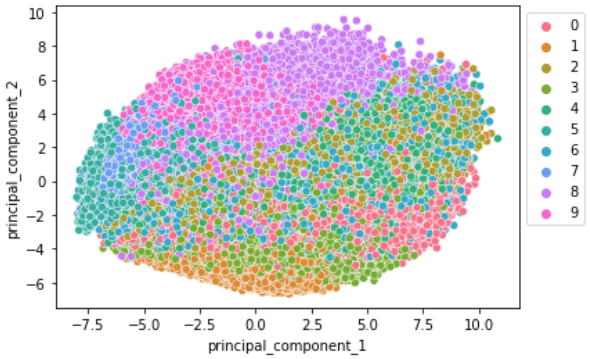


Figure 6 Principal component 1 and 2 values for 10 labels

Set components equals 25, and the cumulative eigenvalue sum is around 0.8. In that scenario, let's see the different label's attributes on the first 2 PCs. For label 9, its main area is (-5.0, 2.5) on PC1, and (4, 8) on PC2, so PC1 and PC2 all dominate it. For label 1, its main area is (-6, -4) on PC2, and (-5, 3) on PC1, so PC2 dominates it which corresponds to Figure 30. For label 5, its main area is (-8, 10) on PC2, and (-4, 4) on PC1. For label 8, its main area is (4, 8) on PC2, and (0, 5) on PC1, so PC1 and PC2 all dominate it. For label 3, its main area is (-5, -1) on PC2, and (-2, 7) on PC1, so PC2 dominates it which corresponds to Figure 30. For label 0, its main area is (-5, -1) on PC2, and (-2, 7) on PC1, so PC1 dominates it which corresponds to Figure 30. For label 2, 4, 6, 7, we can see that they are mixed together, there is no well separated cluster based on this plot. In other words, both of 2 PCs dominate them.

4.3 Summary of Unsupervised Learning

We applied t-SNE and PCA methods to analyze the data structure in training data. From the t-SNE plot based on whole training data, it seems that there are 6 main clusters. Then we applied t-SNE and K-means to see the data structure of one single label, and visualize each subcluster's centroid. For label 8, 5, they do have 2 subclusters. For other labels, they have no well-separated subclusters.

In the PCA part, we analyze the relationship between the first 2 principal components and the labels. And when n_components equals to 25, after checking 10 labels' attributes on the first 2 PCs, we found what is the dominating y label in each cluster. Besides, there are 6 main clusters based on the attributes of different clusters on the first 2 PCs, which corresponds to the t-SNE part.

5. Multi-class Classification Model

5.1 SVM Classification Model

5.1.1 Model Definition

Supported Vector Machine method is selected as one of our multi-class classification models for the below reasons:

a. SVM is effective in high dimensional spaces and has the advantage of avoiding overfitting problems in training data, which is helpful for our image datasets.
b. SVM supports both binary classification and multi-class classification. Although in most cases SVM is applied on binary classification, it could be extended to solve multi-class case problems.
c. Moreover, SVM has been successfully applied in the field of pattern recognition.

Therefore SVM classification model is finally selected as one of our multi-class classification models.

5.1.2 Model Tuning

When fitting the SVM classification model, several kinds of parameters could be tuned based on the measure of their performance. By making many attempts on changing the Kernel, gamma value and penalty C value, an optimal model is finally achieved. The model tuning is completed using GridSearchCV with randomly choiced 5000 sample data from training datasets.

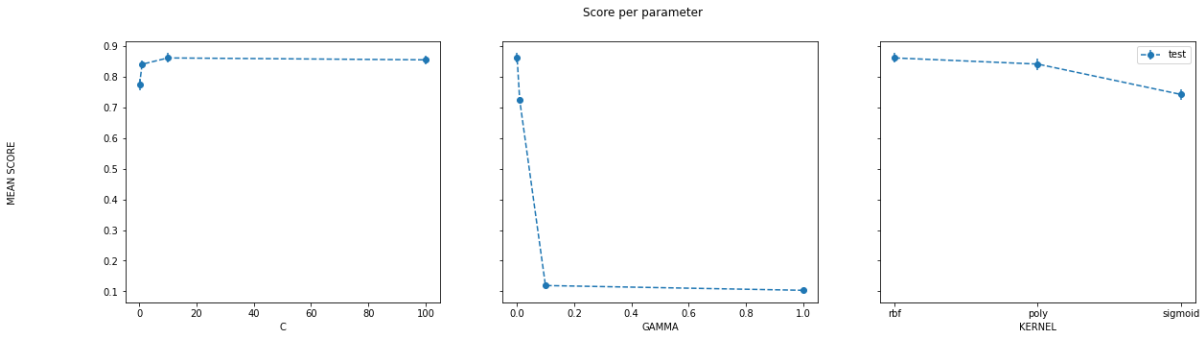


Figure 7 SVM model parameter tuning

Model tuning results show that the optimal model is set with the penalty C value of 10, gamma value of 0.001 and the kernel of radial basis function.

5.1.3 Optimal Model Reporting

a. Model Accuracy Indicators

With the best parameters obtained from the last step, the optimal model has the following performance:

Tabel 3 SVM classification model accuracy report

Training Accuracy	Testing Accuracy
97.37%	90.68%

b. Model Results Visualization

The confusion matrix and the decision boundary projection on 2 dimensional space of the optimal SVM classification model is visualized as follow:

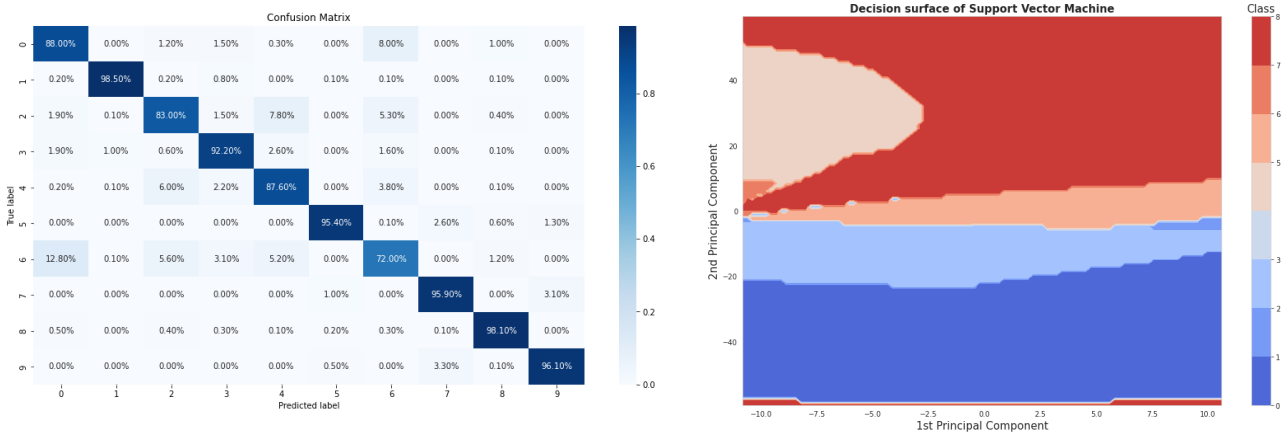


Figure 8 SVM model results visualization (Confusion matrix & Decision boundary)

- The prediction accuracy is considerably high for each label other than the 6^{th} label - Sneaker with the lowest accuracy 72.0%.
- When projecting the classification decision boundary to the first two principal components, the decision boundary could be visualized in a 2 dimensional graph.

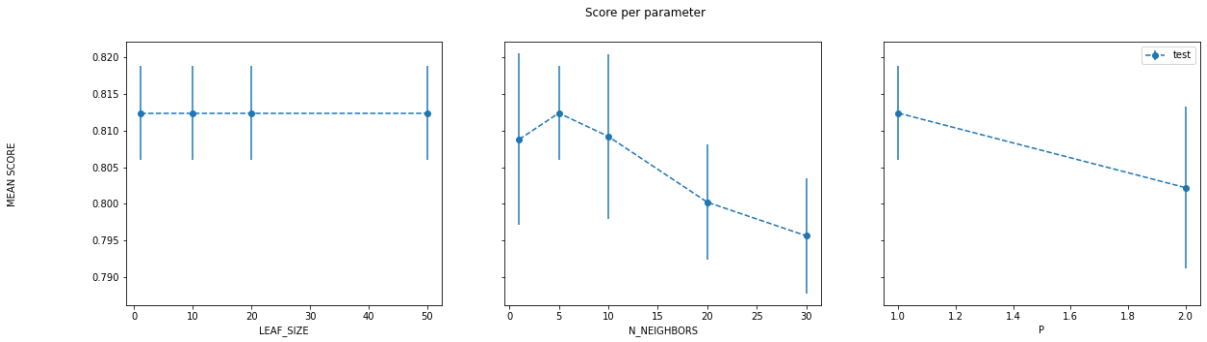
5.2 KNN Classification Model

5.2.1 Model Definition

The K-Nearest Neighbor algorithm is an excellent non-parameter classification model and it is simple and easy to execute.

5.2.2 Model Tuning

In KNN model, the parameters that need to be tuned are k nearest neighbors and the leaf size.



Based on the parameter tuning results, the optimal model is achieved when k nearest neighbors is set to five. The choice of leaf_size and p value does not have significant influence on the model performance.

5.2.3 Optimal Model Reporting

The classification accuracy on training data is 90.13%, while its accuracy on testing data is 86.13%.

Tabel 4 SVM classification model accuracy report

Training Accuracy	Testing Accuracy
90.13%	86.12%

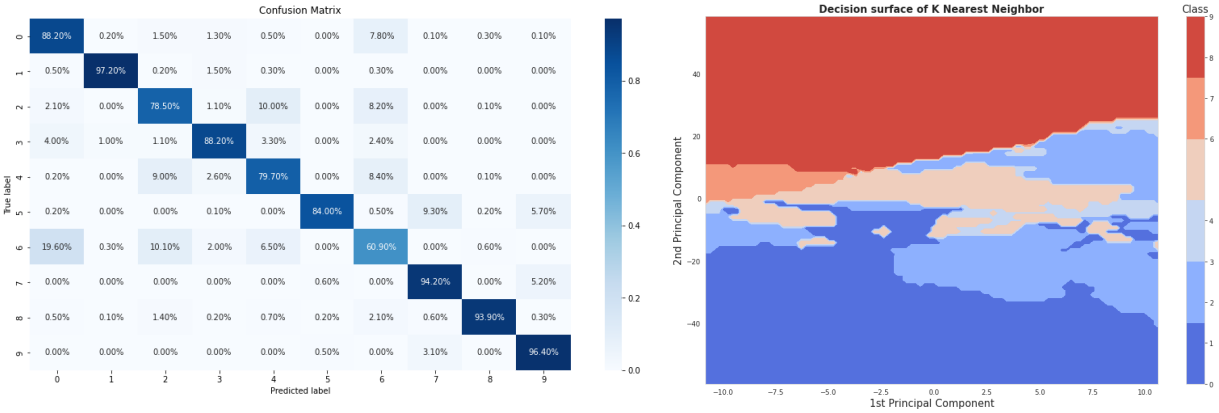


Figure 9 KNN model results visualization (Confusion matrix & Decision boundary)

According to the visualized results of KNN classification, the testing accuracy of each label is worse than the SVM model. The decision boundary of the KNN model is also less clear than the SVM model. Therefore, the classification performance of KNN is worse than the SVM model.

5.3 Extension from Binary Logistic Classification to Multi-Class Classification

5.3.1 Method Framework

Basically, there are two kinds of methods to perform multi-class classification based on binary classification, one vs one method and one vs all method. The first method is implemented by splitting the multi-class problem into multiple binary classification problems and fitting a binary logistic regression model to each problem, while the second one splits a multi-class classification into one binary classification for each class. The approach we chose here to extend the binary logistic model is **one vs all** method. Its framework could be basically describe as follow:

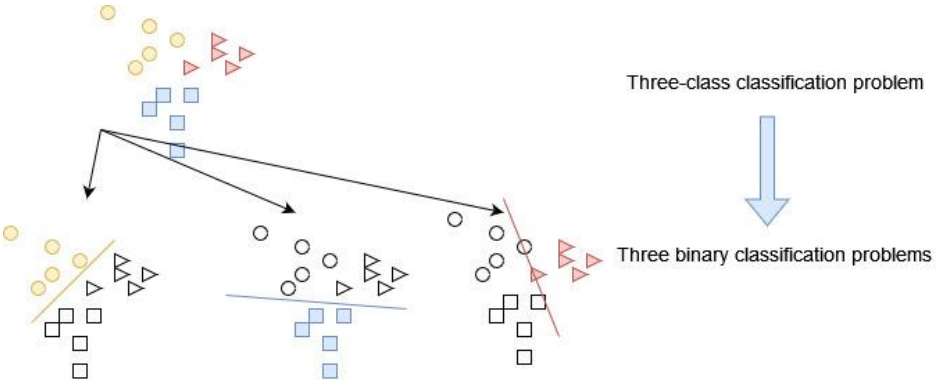


Figure 10 Method framework of extended logistic classification

In this method, the original y label column is processed to a 10 columns data, while for each column in the new y data, we will implement a binary classification. For example, when working with label 2 column, it will return 1 for the original label 2 and return 0 for the rest labels. Also, the gradient

descent method is used to optimize the cost function in the logistic classification model. The cost function describes how far the prediction from the original output: [1]

$$C = -\frac{1}{m}(\sum y_{original}{}^{i} logy^{i}_{pred} + (1 - y_{original})log(1 - y^{i}_{pred}))$$

where the m is the number of training data. Based on the cost function, the optimizing parameter in each iteration is below:

$$\theta = \theta - \alpha\sum(h^{i} - h^{j})X^{j}_{i}$$

5.3.2 Model Reporting and Comparison

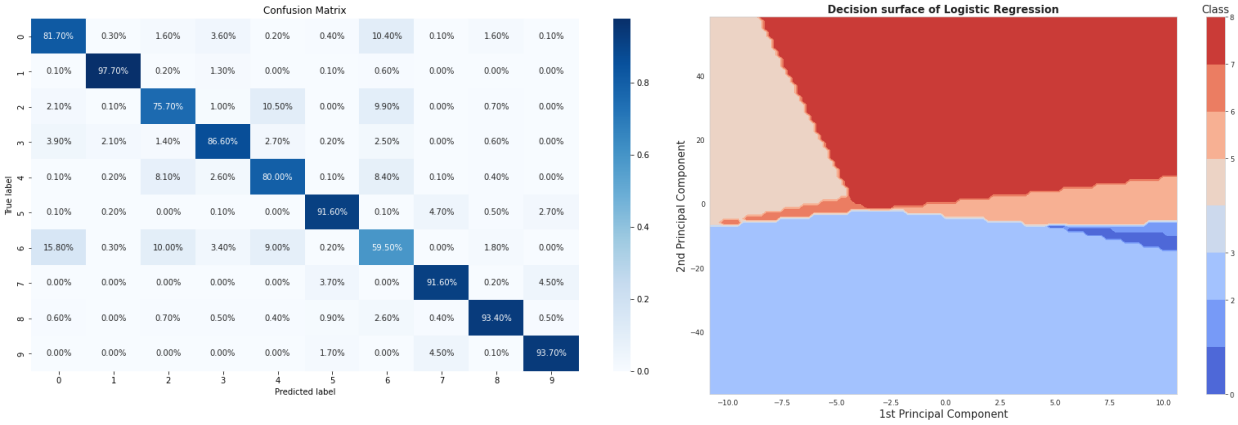


Figure 11 Extended logistic classification model visualization

According to the visualized results of the extended logistic classification

Tabel 5 SVM classification model accuracy report

	Training Accuracy	Testing Accuracy
SVM optimal model	97.37%	90.68%
KNN optimal model	90.13%	86.12%
Extended Logistic Classification	87.66%	85.15%

6. Ensemble Model and Feature Engineering

Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. There are many kinds of ensemble methods, like Max-voting, Stacking, Blending. We choose Max-voting as our ensemble model.

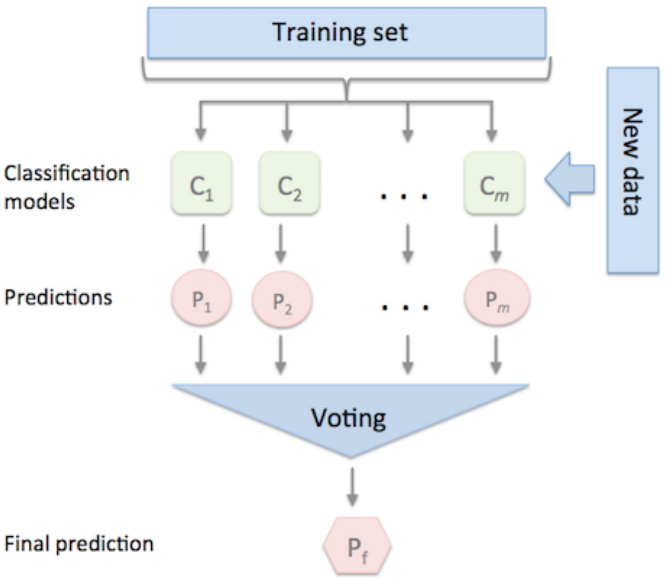


Figure 12 Max-Voting flowchart

The Max-Voting process is as follows:

1. Train base learners and record their prediction labels and testing accuracies.
2. If two of the three labels are the same, it will be the final predicted label.
3. If all three labels are different, we choose the label predicted by the model with the highest accuracy as our final predicted label.

The key of Max-voting is to get high performance base learners. If the performance of base learners is not good, the ensemble result will not be good as well. In order to get a good final result, we choose 3 models that perform well on the training set, which are SVM, XGBoost and LightGBM.

Take Lightgbm as an example:

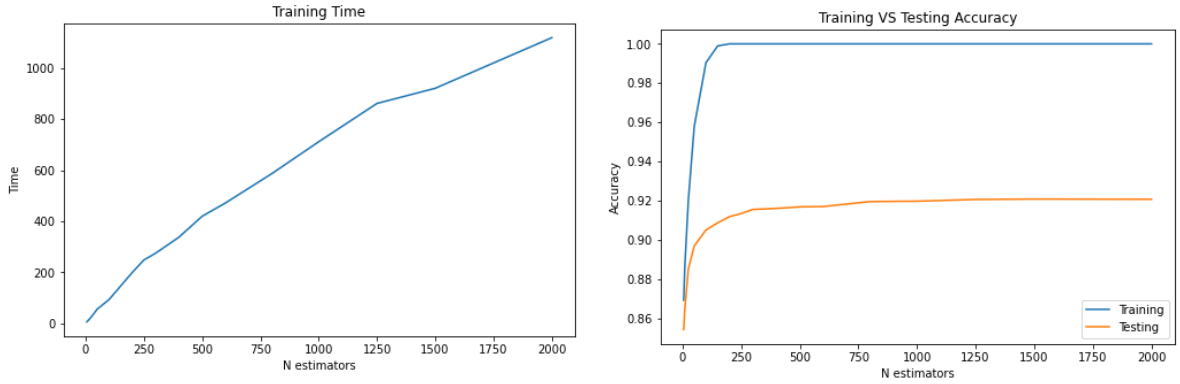


Figure 13 Lightgbm model parameter tuning

There are so many hyper parameters in the Lightgbm model. Some parameters are important,like num_leaves, max_depth, n_estimators, since these parameters can directly influence the numbers or complexity of the base learner. Theoretically, we can use grid search or bayes optimization to tune the parameters, expecting to get the best combination of parameters. Since tuning multiple parameters will cost a lot of time, to simplify the task, I only choose to adjust one of the most important parameters-n_estimators, which stands for the boosting rounds.

As we can see from the left graph, as the number of estimators becomes larger, the training time becomes larger.

As we can see from the right graph, as the number of estimators becomes larger, the training accuracy immediately reaches 1 and the testing accuracy immediately reaches 0.9, then gradually increases. To be specific, when N estimators = 1250, testing accuracy is 0.9206; when N estimators = 1500, testing accuracy is 0.9208; When N estimators = 2000, testing accuracy is 0.9207. So, we choose N estimators = 1500, since it achieves the highest testing accuracy, anything beyond that is a little bit overfitting.

After training the three sub models, we concatenate their outputs as the input for the second stage. The first stage output is as follows and it is pretty intuitive.

Tabel 6 First stage output

Data	SVM	XGBoost	LightGBM
0	0	0	0
1	1	1	1
2	4	3	3
3	0	6	6

Each row represents the predicted labels of svm,xgboost and lightgbm. For example, If predicted labels are “0,0,0”, the final ensemble model will output 0; If predicted labels are “4, 3, 3”, the final ensemble model will output 3. The final results are as follows:

Tabel 7 Model results comparison

	Training Accuracy	Testing Accuracy
XGBoost optimal model	100%	91.79%
SVM optimal model	97.37%	90.68%
LightGBM optimal model	100%	92.08%
Ensemble model	**100%**	**92.12%**

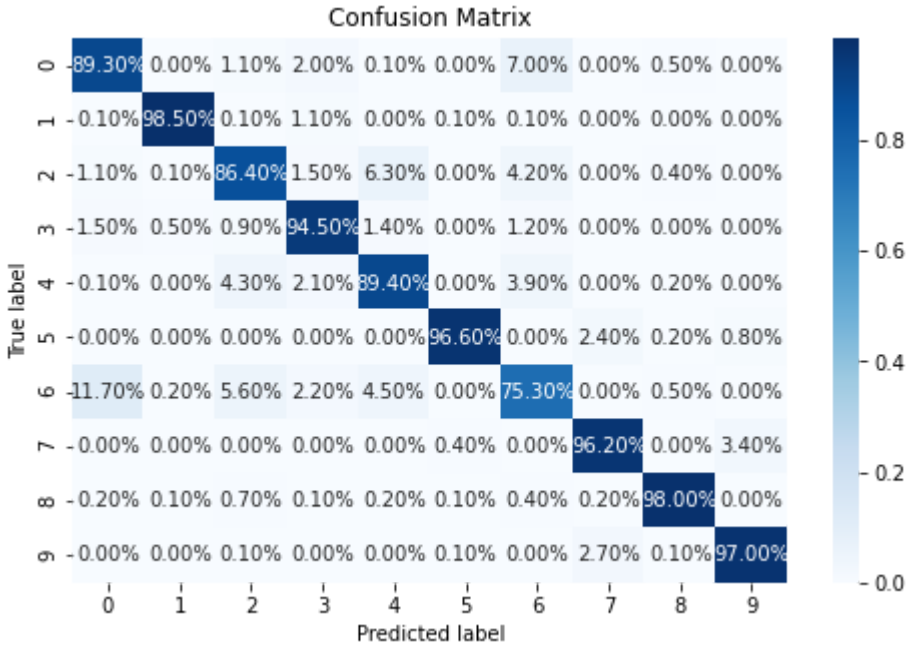


Figure 14 Confusion matrix of ensemble model

As we see from the confusion table, most of the labels are predicted well. But accuracy is not so high for labels 2,4 and 6. This is not strange, since in the t-SNE plot, labels 0, 2 , 4 and 6 are mixed with each other, which tells us it's difficult to predict them very accurately.

Reference:

[1]https://towardsdatascience.com/multiclass-classification-algorithm-from-scratch-with-a-project-in-python-step-by-step-guide-485a83c79992

[2]Greeshma, K. V., and K. Sreekumar. "Hyperparameter optimization and regularization on fashion-MNIST classification." *International Journal of Recent Technology and Engineering (IJRTE)* 8.2 (2019): 3713-3719.

[3]Greeshma, K. V., and K. Sreekumar. "Fashion-MNIST classification based on HOG feature descriptor using SVM." *International Journal of Innovative Technology and Exploring Engineering* 8.5 (2019): 960-962.