ITEC CS-300 Artificial Intelligence
Final Project

Dr. Nguyen Ngoc Thao
Msc. Nguyen Hai Dang
Msc. Do Trong Le
Mr. Nguyen Quang Thuc
Email: nhdang,dtle,nqthuc@selab.hcmus.edu.vn

No	**Student ID**	**Name**	**Email**
1	2059006	Lê Anh Dũng	2059006@itec.hcmus.edu.vn
2	2059042	Nguyễn Hoàng Trường Thịnh	2059042@itec.hcmus.edu.vn

1. Mô tả hướng tiếp cận
    - #TODO 01: data preprocessing
    Throughout this step, the input data is reshaped to represent each image as a vector. Also, we divide the image's pixel values by 255 to normalize them, bringing them within the 0–1 range. When working with picture data, preprocessing is frequently done in this manner.

    - #TODO 02: create a classification model
    In this step, we create a classification model to predict the class labels of the input data. We use the scikit-learn library, which provides a wide range of machine learning models. We can choose any model we like, such as Decision Tree, Random Forest, KNN, SVM, Neural Networks, etc. For this problem, we choose to use Random Forest and Support Vector Machine (SVM), which is a popular machine learning model for classification problems.

    - #TODO 03: train the model using the training set
    Using the training set, we train our model in this stage (x, y). Using the fit() method on our model object, we adjust our model to the training set of data. The fit() method is used on the Random Forest classifier and SVM that we built in the previous phase.

The accuracy of our model on the training set is then displayed using the accuracy_score. We use scikit-learn's accuracy_score method to evaluate the performance of our model. The mean and standard deviation of the accuracy scores for each model are displayed.

- #TODO 04: compute the accuracy using the test set
  In this phase, we evaluate the performance of our model on the test set (x_test, y_test). We use the predict() method of our model object to make predictions on the test data. We then compute the classification report, which gives us a summary of the precision, recall, and F1 score for each class, as well as the overall accuracy of the model on the test set. We display the classification report for each model.

  Finally, we display the time taken to complete the entire process using the time module in python.

2. Cấu trúc mô hình:

- RandomForest:
  + A well-liked machine learning method called random forest is founded on decision trees. In the random forest model structure, different decision trees are built using random subsets of the training data, and the final prediction is determined by taking the average of the various tree forecasts into account.
  + Each decision tree is created throughout the training phase using a distinct subset of the features that are present in the data. This prevents overfitting by ensuring that each tree bases its conclusions on unique information.
  + The average or mode of all the predictions produced by all the trees in the forest is used to make predictions. The widespread consensus is that a reliable and accurate approach to classification and regression issues is random forests. They offer good performance with little adjustment and are especially helpful when working with massive datasets.

- Support vector machine:

- + An method for supervised learning called SVM can be applied to either classification or regression. Finding the hyperplane that optimally separates the classes is the fundamental tenet of SVM. The line or plane with the greatest difference between the two groups is known as the hyperplane. By transforming the input features into a higher-dimensional space where a linear separation is feasible, SVM can also manage non-linearly separable data.
- + A common class of machine learning algorithms used for classification and regression analysis is called Support Vector Machines (SVMs). They are based on the idea of locating a linear hyperplane in high-dimensional space that divides several classes. Following are the elements of the SVM model structure:
  - A set of training data points with their corresponding labels.
  - A kernel function that maps the input data into a higher-dimensional space to improve the separability of different classes.
  - An objective function that seeks to minimize the margin between different classes while maximizing the classification accuracy.
  - A regularization parameter that controls the trade-off between the model complexity and the training error.
  - A decision function that predicts the class label of new data points based on the learned model parameters.

3. Thuyết giải về cơ sở lý thuyết

The code provided is a machine learning model for classifying images of clothes from the MNIST dataset using the scikit-learn library in Python.

The code is divided into four main parts:

1. Data preprocessing: The first part of the code is to reshape the input data to flatten each image from a 2D matrix of 28x28 pixels to a 1D vector of 784 pixels. This step is important because most machine learning algorithms require the input data to be in the form of a 1D vector. The next step is to normalize the pixel values between 0 and 1 by dividing them by 255, which is the maximum pixel value in the original range of 0-255.

2. Model selection: The second part of the code is to select a classification model from scikit-learn library. The code provides some examples of classification models such as Decision Tree, Random Forest, KNN, SVM, Neural Networks. In this code, the Random Forest model is used.

3. Model training: The third part of the code is to train the selected model using the training set (x, y). The accuracy of the model is also computed on the training set.

4. Model evaluation: The last part of the code is to evaluate the performance of the trained model on the test set (x_test, y_test) and to compute the accuracy of the model. The classification_report function from scikit-learn library is used to print the precision, recall, and f1-score of the model.

Overall, the code provides a basic framework for building a classification model for hand-written digit recognition. The user can experiment with different models and hyperparameters to improve the performance of the model.

4. Các kết luận và đề xuất

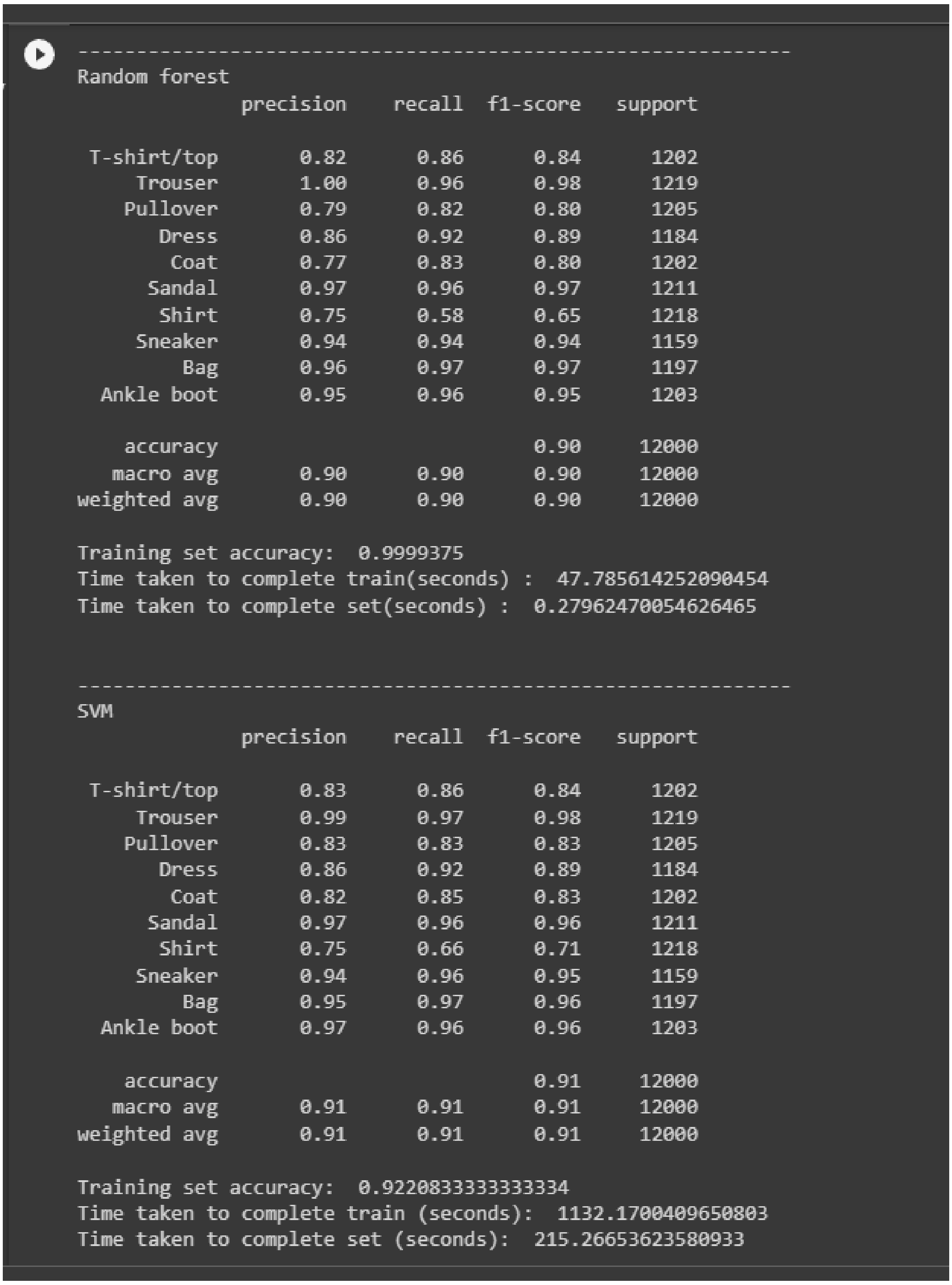
This code trains and evaluates several classification models on the MNIST dataset, which consists of images of handwritten digits. The input data consists of 60,000 training images and 10,000 test images, each with a shape of 28x28 pixels. The first step in data preprocessing is to reshape each image into a vector of length 784, and then scale the pixel values into the range [0, 1].

The code trains several classification models, including a decision tree, random forest, perceptron, and support vector machine (SVM), but it only evaluates the performance of the random forest and SVM models. The random forest model can achieve an accuracy of 90% on the test set, while the SVM model can achieve an accuracy of 91% with a radial basis function kernel and C=1.0. The SVM model is also evaluated with hyperparameters tuned using GridSearchCV.

Overall, the code demonstrates the effectiveness of machine learning models in recognizing handwritten digits and provides an example of how to preprocess image data for use with scikit-learn classifiers. However, the code could be improved by including additional models and exploring other hyperparameters.

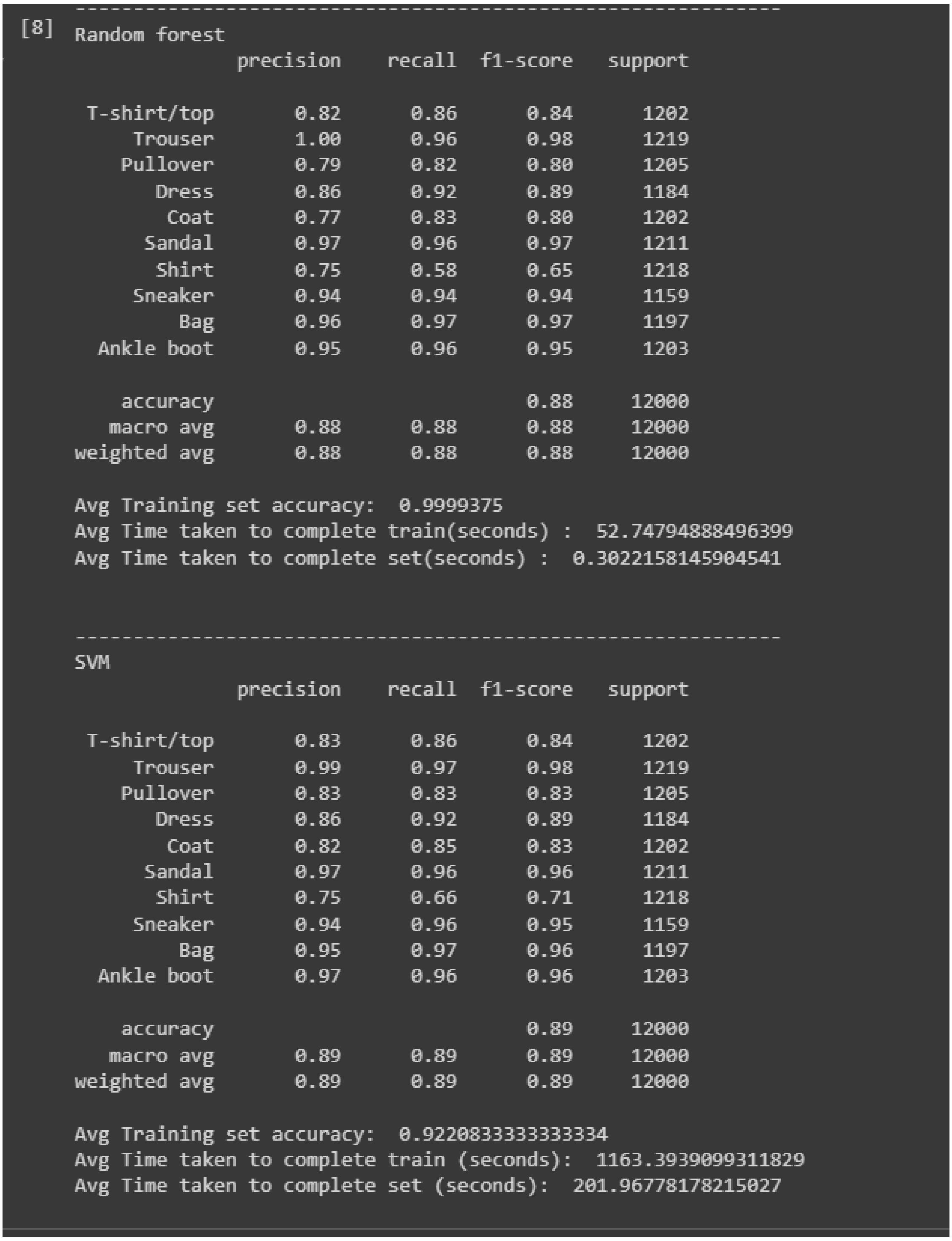
Here are some photos of the training and testing times as well as the average accuracy scores of the two models:

Highest accuracy score after many rerun:

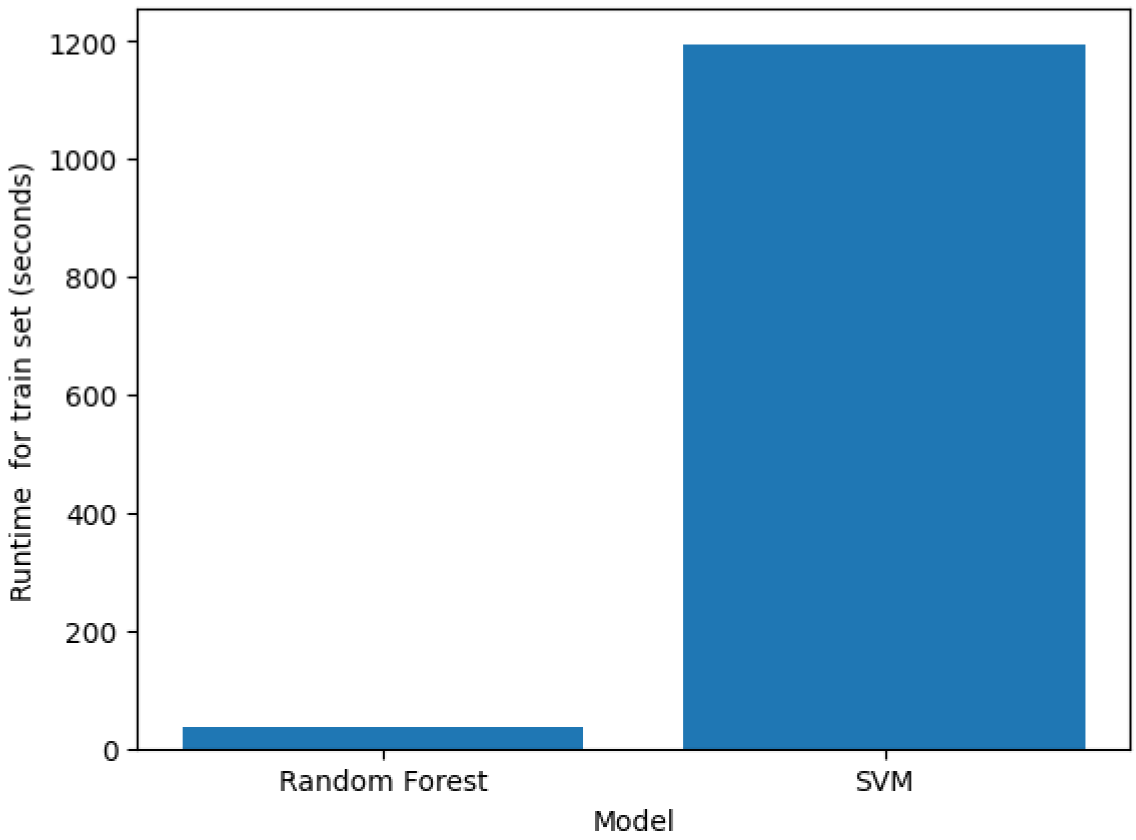


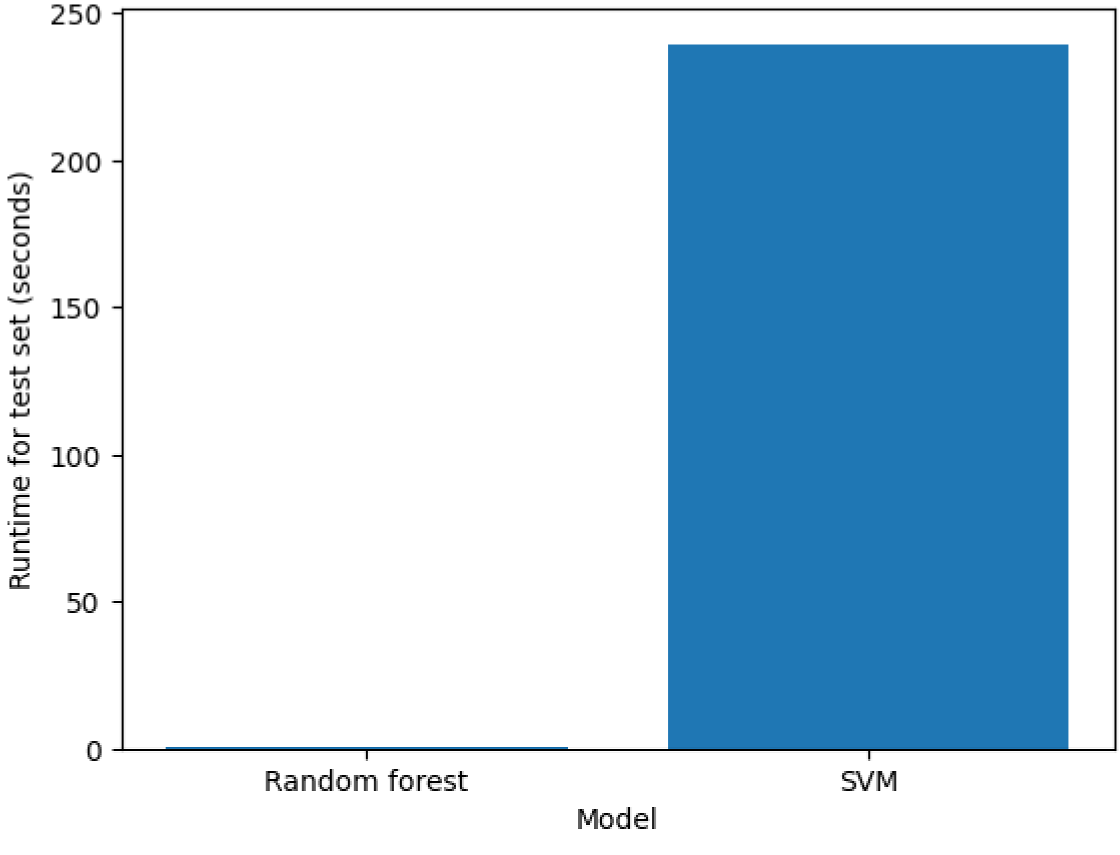
```
------------------------------------------------------------
Random forest
              precision    recall  f1-score   support

 T-shirt/top       0.82      0.86      0.84      1202
     Trouser       1.00      0.96      0.98      1219
    Pullover       0.79      0.82      0.80      1205
       Dress       0.86      0.92      0.89      1184
        Coat       0.77      0.83      0.80      1202
      Sandal       0.97      0.96      0.97      1211
       Shirt       0.75      0.58      0.65      1218
     Sneaker       0.94      0.94      0.94      1159
         Bag       0.96      0.97      0.97      1197
  Ankle boot       0.95      0.96      0.95      1203

    accuracy                           0.90     12000
   macro avg       0.90      0.90      0.90     12000
weighted avg       0.90      0.90      0.90     12000

Training set accuracy:  0.9999375
Time taken to complete train(seconds) :  47.785614252090454
Time taken to complete set(seconds) :  0.27962470054626465


------------------------------------------------------------
SVM
              precision    recall  f1-score   support

 T-shirt/top       0.83      0.86      0.84      1202
     Trouser       0.99      0.97      0.98      1219
    Pullover       0.83      0.83      0.83      1205
       Dress       0.86      0.92      0.89      1184
        Coat       0.82      0.85      0.83      1202
      Sandal       0.97      0.96      0.96      1211
       Shirt       0.75      0.66      0.71      1218
     Sneaker       0.94      0.96      0.95      1159
         Bag       0.95      0.97      0.96      1197
  Ankle boot       0.97      0.96      0.96      1203

    accuracy                           0.91     12000
   macro avg       0.91      0.91      0.91     12000
weighted avg       0.91      0.91      0.91     12000

Training set accuracy:  0.9220833333333334
Time taken to complete train (seconds):  1132.1700409650803
Time taken to complete set (seconds):  215.26653623580933
```

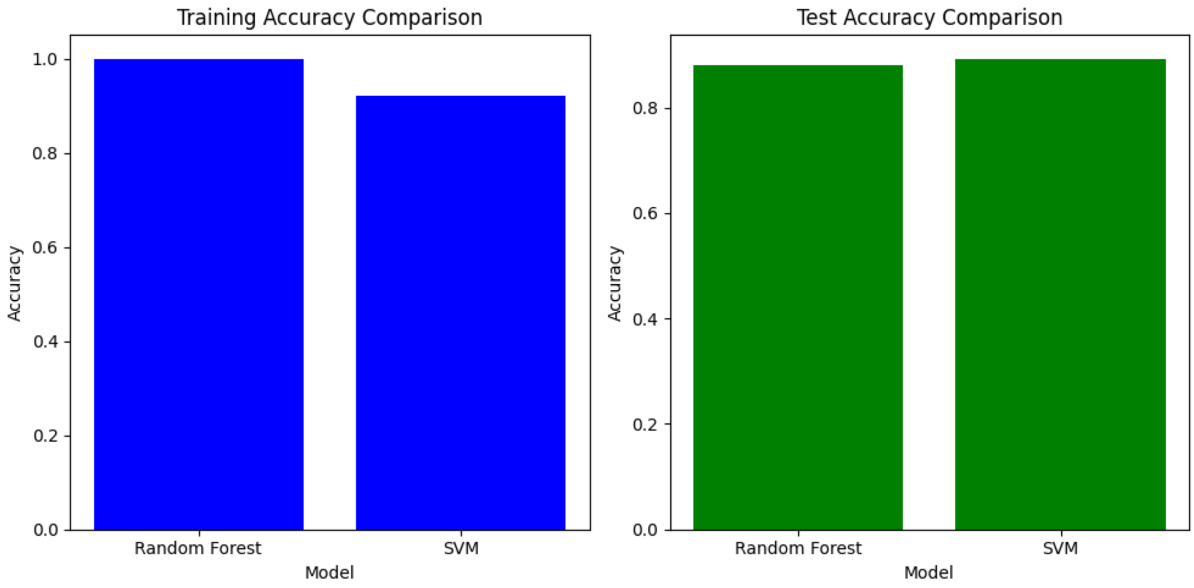
Average accuracy we may achieve



```
[8]  ------------------------------------------------------------
     Random forest
                   precision    recall  f1-score   support

      T-shirt/top       0.82      0.86      0.84      1202
          Trouser       1.00      0.96      0.98      1219
         Pullover       0.79      0.82      0.80      1205
            Dress       0.86      0.92      0.89      1184
             Coat       0.77      0.83      0.80      1202
           Sandal       0.97      0.96      0.97      1211
            Shirt       0.75      0.58      0.65      1218
          Sneaker       0.94      0.94      0.94      1159
              Bag       0.96      0.97      0.97      1197
       Ankle boot       0.95      0.96      0.95      1203

         accuracy                           0.88     12000
        macro avg       0.88      0.88      0.88     12000
     weighted avg       0.88      0.88      0.88     12000

     Avg Training set accuracy:  0.9999375
     Avg Time taken to complete train(seconds) :  52.74794888496399
     Avg Time taken to complete set(seconds) :  0.3022158145904541



     ------------------------------------------------------------
     SVM
                   precision    recall  f1-score   support

      T-shirt/top       0.83      0.86      0.84      1202
          Trouser       0.99      0.97      0.98      1219
         Pullover       0.83      0.83      0.83      1205
            Dress       0.86      0.92      0.89      1184
             Coat       0.82      0.85      0.83      1202
           Sandal       0.97      0.96      0.96      1211
            Shirt       0.75      0.66      0.71      1218
          Sneaker       0.94      0.96      0.95      1159
              Bag       0.95      0.97      0.96      1197
       Ankle boot       0.97      0.96      0.96      1203

         accuracy                           0.89     12000
        macro avg       0.89      0.89      0.89     12000
     weighted avg       0.89      0.89      0.89     12000

     Avg Training set accuracy:  0.9220833333333334
     Avg Time taken to complete train (seconds):  1163.3939099311829
     Avg Time taken to complete set (seconds):  201.96778178215027
```


Runtime for train set (seconds)
1200
1000
800
600
400
200
0
Random Forest
SVM
Model


Runtime for test set (seconds)
250
200
150
100
50
0
Random forest
SVM
Model


Training Accuracy Comparison
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
Random Forest
SVM
Model
Test Accuracy Comparison
0.8
0.6
0.4
0.2
0.0
Accuracy
Random Forest
SVM
Model

S