

Evaluating machine learning models for classification



Hossein Hosseini

EC Utbildning

Machine learning assignment-2 report

202503

Abstract

In this report multiple machine learning models have taken for handwritten digits classification using MNIST dataset. This report goes through three different machine learning models which are Support Vector Machine (SVM), Random Forest Classifier (RFC) and Logistic Regression (LR) to predict handwritten digits. We will see the models performance and how to improve their performance after optimizing via hyperparameter tuning. The models will be trained with a subset of 30000 samples of MNIST data set from sklearn datasets. After all processes the best performing model will be selected based on accuracy. At the end the models will be tested to predict test dataset as new dataset. For ease of use of these models to predict new data in this project, a Streamlit application will be implemented. By this the users will be able to interact and draw and even upload handwritten digit image to be predicted.

Contents

Abstract	2
1 Introduction.....	4
2 Theory.....	5
2.1 Support Vector Machine (SVM)	5
2.1.1 Gamma in SVM	5
2.1.2 Regularization in SVMs	5
2.2 Random Forest Classifier (RFC)	6
2.2.1 Number of Trees.....	6
2.2.2 Maximum Depth.....	6
3 Method.....	7
3.1 Preprocessing the data set.....	7
3.2 Hyperparameter tuning	7
3.2.1 Choosing hyperparameters for SVM	7
3.2.2 Choosing hyperparameters for RFC.....	8
3.3 Select the best model	8
3.4 Model deployment	8
3.5 Structure of the project	9
3.6 Challenges and Limitations	9
4 Results and discussion	9
5 Conclusion	10
6 Teoretiska frågor	11
7 Självutvärdering.....	13
Appendix A	14
Källförteckning.....	15

1 Introduction

Handwritten digit recognition is an essential field within computer vision and machine learning, widely applied in areas such as automated postal sorting and reading forms by machine etc. With advancements in machine learning techniques, the ability to accurately classify handwritten digits has significantly improved. The MNIST dataset is a well-established benchmark in machine learning. It is a good data set for testing various classification models.

The purpose of this report is to analyse and compare three machine learning models for handwritten digit classification and to determine the most suitable model based on performance metrics. In this study Support Vector Machine (SVM), Random Forest Classifier and Logistic Regression are being considered to determine their effectiveness in handwritten digit classification.

How do different machine learning models perform in terms of accuracy when classifying handwritten digits?

Why do we focus on accuracy score in this study?

Why is hyperparameter tuning important?

How does feature scaling impact model performance?

2 Theory

2.1 Support Vector Machine (SVM)

Support Vector Machines (SVMs) are supervised learning models used for classification tasks. Via SVM we can find an optimal boundary with margin between different classes. There are different Kernel functions that can be used for different problems and decision functions. Since high-dimensional dataset MNIST which has 784 features is used in this work the best choice is RBF Kernel. The other Kernels are suitable for datasets with less complexity like binary classification.

2.1.1 Gamma in SVM

Gamma is a hyperparameter in SVM that controls the effect of every single point when classifying the points. Small Gamma results in a larger radius which means each data point affects a larger area. We will have smoother and more generalized decision boundaries. When we get a larger Gamma the points affect on a smaller area nearby the point which creates a complex decision boundary (overfitting).

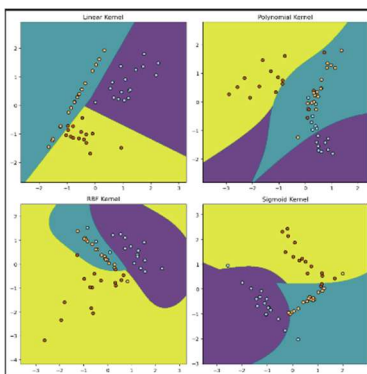


Figure 2.1 different Kernels in SVM

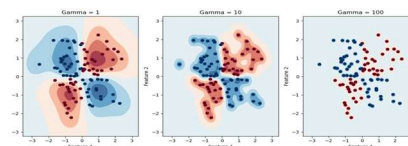


Figure 2.2 Gamma in SVM

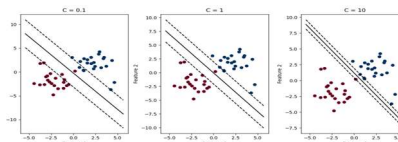


Figure 2.3 C in SVM

2.1.2 Regularization in SVMs

Regularization in machine learning helps to prevent overfitting and helps to generalize the model to have good performance with unseen data. In SVM, regularization plays a key role in controlling the trade-off between having a large margin and minimizing the classification errors. In the real world data sets have noise, outliers or overlapping classes. This makes challenges to find a proper boundary to separate the classes. In SVM, the regularization parameter is C. Having large C can lead to classify all training data correctly that results to overfitting and vice versa.

2.2 Random Forest Classifier (RFC)

Random Forest is a machine learning method widely used for classification and regression problems. It works by creating multiple decision trees on random subsets of the data and combining their predictions to make the final decision. This ensemble approach reduces the risk of overfitting that can occur with individual decision trees and improves the model's overall performance and robustness. Random Forest is especially effective for high-dimensional datasets like MNIST, which contains many features (pixels), making it a powerful tool for complex classification problems.

2.2.1 Number of Trees

The number of trees (estimators) refers to how many decision trees are built. More trees generally can cause to better accuracy, but too many can take longer to train without much improvement.

2.2.2 Maximum Depth

The depth of a tree decides how much detail it learns. A deep tree can take a lot of details but may only work well on training data, not on new data. A simpler tree avoids this problem and works better on data it hasn't seen before.

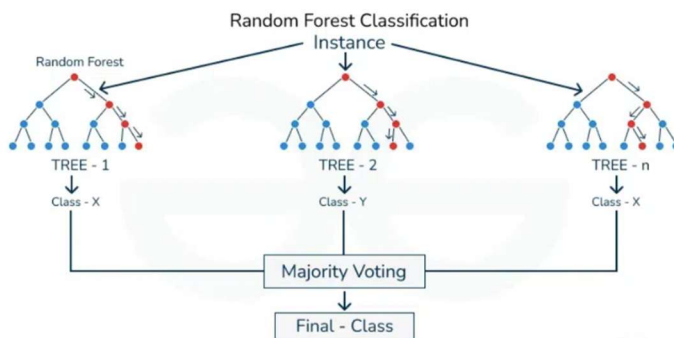


Figure 2.4 RFC

3 Method

3.1 Preprocessing the data set

Firstly, it is needed to fetch the data set from sklearn datasets. Define x features and y as target from the data set. In this work only 30000 rows of the dataset are taken because the whole data set takes long time to train the model with limitation of processor I have. The data needs to be split more into train, validation and test set. The train set will be used to train and fit the model then the model will be validated by validation set and finally the model will be tested with test set. Now it is time to scale or standardize the data sets in order to make sure that all features are on the same scale. This helps models perform better by preventing some features from dominating others.

$z = \frac{x - \mu}{\sigma}$ μ is the mean of the feature, x is the original value of the feature and σ is the standard deviation of the feature.

3.2 Hyperparameter tuning

By hyperparameter tuning the performance of a model can be improved. Because there are different parameters that impact the performance of the model and there are different problems in real world. Grid search and cross validation are the greatest examples for tools to find the best parameters. This process is done via giving a part of the data set to every combination of different value for hyperparameters and fit the model. In this work grid search is taken to iterate through different hyperparameters with different values. I used kind of the same hyperparameter tuning approach for all models in this work but of course with related hyperparameters for each model.

3.2.1 Choosing hyperparameters for SVM

By searching common values for regularization and gamma in Scikit-learn.org I found the range for each parameter for this SVM model. In SVM model, regularization is managed by the C parameter. This parameter determines how much margin the model takes. The larger margin (softer margin) the more misclassifications the model does. That means that we will get a model that is more tolerant to errors in the training and it decreases getting overfit. By large C we will get smaller margin but a model that is overfitted and weaker to predict unseen data. I chose 1, 10 and 20 for C values in grid search.

As mentioned above RBF Kernel is the best Kernel for MNIST data set. I used grid search as a tool to find the best values for hyperparameters. After preparing grid search I fitted the model with the train set and then found the best estimator.

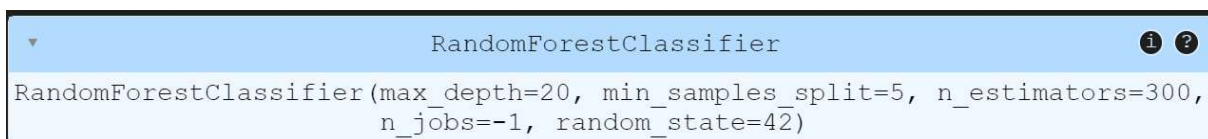
I did a prediction on validation set and obtain the accuracy score with comparing target value with the predicted values by the model. In order to have this model in the application (streamlit app) which I will discuss about later in this report, I saved the model with joblib.

3.2.2 Choosing hyperparameters for RFC

I searched for those common values for n_estimators in scikit-learn web site, so I got 100, 200 and 300. I chose none, 1, 5, 10, 20 for max_depth as hyperparameters. I used the same process to find the best estimator which is grid search. The same process to validate the model via validation data set is done for this model as well.

3.3 Select the best model

After all steps above training and validation it is time to choose the best model. The best model was therefore tested on the test set as new data that each model has not seen before to assess their generalization capabilities. Each model's performance was measured using classification metric accuracy. According to my setup the final best model was Random Forest Classifier with the following hyperparameters.

A screenshot of a Jupyter Notebook cell. The cell has a blue header bar with the text 'RandomForestClassifier' and two icons (an 'i' and a '?') on the right. Below the header, the cell contains the following Python code:

```
RandomForestClassifier(max_depth=20, min_samples_split=5, n_estimators=300, n_jobs=-1, random_state=42)
```

3.4 Model deployment

I created a app.py file in order to deploy the models to allow users to interactively write, draw and upload the handwritten digits and see the prediction by the models. I have saved all models including the best model to have a overview of their prediction in the application at the same time. I studied different resources including Youtube to be able to create the Streamlit application.

I have two different sections in the application, the first one is a section there users can draw or write a digit by hand and the other section there users can upload an image of handwritten digit.

For ease of use I have added some test data (images) in a folder named "samples" in the project.

3.5 Structure of the project

In this project I have tried to separate the files that does different jobs. For instance I created a file named functions.py to have all necessary functions in one place. I have created app.py file to have Streamlit application in it. All models are saved in the root path in the project.

In this project I have added a requirements.txt file to list all libraries that are needed for the project and a readme.md file in order to users can follow the instructions to be able run the project.

3.6 Challenges and Limitations

During the implementation of this project, I encountered some challenges including hyperparameter tuning and training the model with a large data set with high dimension.

For hyperparameter tuning I had to train the models with different values which was time consuming. I referred to different resources to find the best values of parameters for my data set.

Training the model was also time consuming due to my limitation on my computer (processor with 8 cores). When I was pushing my project to github I got also issue with large files (the models) so I compressed those files as zip files. I have included instruction how to unzip them in the readme file.

4 Results and discussion

In this project as we can see in the table below, RFC and SVM achieved the highest accuracy but required significantly longer training times which I think can make it expensive for the real time applications. On the other hand Logistic Regression model takes less time to be fitted but unfortunately with the lowest accuracy among the three models. Overall, I think the best model for this problem is Random Forest Classifier with highest accuracy and relatively less expensive in terms of computation compared to Support Vector Machine.

I think to decrease the cost of the computation time we could have a better preprocessing and hyperparameter tuning so Grid search functionality has less case to compute. In addition we will achieve a better model that has balance between bias and overfitting.

Accuracy score for the models	
RFC	96,1 %
SVM	96 %
LR	92 %

Here we can see the confusion matrix obtained by predicting the test data set for RFC model. The numbers along the diagonal line shows that the predicted value is quite high.

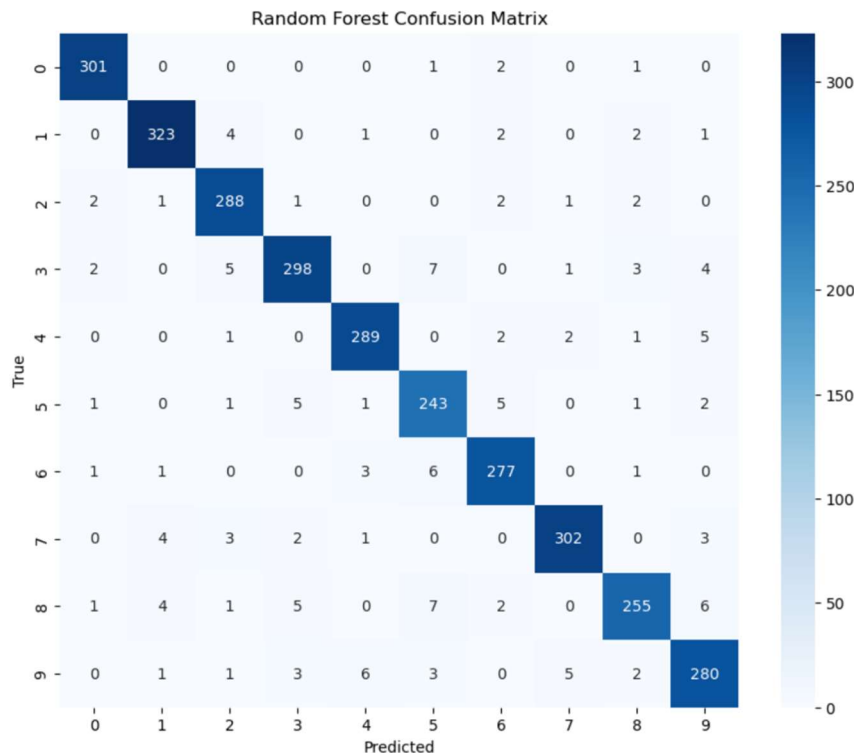


Figure 4.1 RFC model matrix

5 Conclusion

This project aimed to evaluate and compare the performance of SVM, RFC and LR machine learning models for classifying handwritten digits from the MNIST dataset. The results showed that RFC achieved the highest accuracy. LR training was lighter than the others but had the lowest accuracy, making it less suitable for this kind of data set.

We know that accuracy is a key metric in this kind of classification problem when the dataset is balanced. Since misclassification can lead to incorrect predictions so high accuracy ensures reliable predictions.

hyperparameter tuning has a important tole in model performance. In SVM, tuning parameters like C and gamma affects the decision boundaries and model prediction on new data. In RFC, choosing the right number of trees and depth can give us better model to predict new data. Via the functionality of Grid search we computed all those different values for the parameters in each model and achieved the best estimator.

6 Teoretiska frågor

1. The main dataset is typically split into three subsets: a training set (70%), a validation set (15%), and a test set (15%).
Training data is used to train the machine learning algorithms like SVM, Linear regression and Random forest etc. The model **learns relationships** between features (input) and labels (output).
Validation set is used to find the best performing hyperparameters of the model. This part is taken as a part of the training of the model. The model use this data only for evaluation and does not learn from it. The model uses this data which is not training data and is not seen before.
After all training and validation data set the model is ready to use test data set for final evaluation. By test data set the model will be tested how well the model can predict new data. We are able somehow to estimate real world performance before deployment.
2. When there is no explicit validation set the best approach is to use k-fold cross-validation before fitting the model to evaluate the performance of each model. With applying that it will be possible to compute the average performance metric like RMSE for each model.
3. A regression problem is referred to a type of supervised machine learning technique. Via this type of supervised machine learning a continuous numerical value dependent on one or more features will be predicted. In this type we have dependent variable which is called target and independent variables called features.
Output or target values are real or continuous value such as price, weight etc.
There are many different models of this type:
Linear regression, polynomial regression, Lasso regression, Random forest regression etc.
Some potential application areas are as follows:
Predicting stock prices, loan defaults and investment returns in finance.
Estimating property values or rental prices.
Predicting patient outcomes, disease progression in healthcare.
Forecasting sales, demand and revenue and estimating the impact of advertising on sales.
4. RMSE (Root Mean Squared Error) is a highly used metric in machine learning to measure the performance of regression models. It calculates the difference between the predicted values and the actual values in a regression model. It is in fact the root of average of the error between predicted values and actual values. The lower RMSE the model has the better performance and closer to the actual values results the model will have.
5. A classification problem is a supervised machine learning algorithm which predicts variables of type discrete (in contrast with Regression which is continuous). The model learns by getting training data and then predicts which category a new item belongs to.
It actually after learning creates a decision boundary that separates the classes in the feature space.

Examples of classification problems: image classification like handwriting recognition, medical diagnosis by getting patient symptoms and test results and predicting if it is disease or not.

Some widely used models for this problem are:

Support Vector Machines (SVM): finds the optimal boundary to separate classes in the feature space.

Random Forest: a group of decision trees that improves accuracy and reduces overfitting.

6. K-means model is a unsupervised machine learning algorithm used for clustering. It groups similar data points based on their features similarity. The algorithm starts with choosing randomly k centroids and then calculates the distance between each data point and the centroids to determine which data point belongs to which centroid.

The new centroids will be determined and again the above calculation will be repeated until the centroids no longer changes significantly.

Applying areas:

In marketing the companies can group the customers based on their shop behavior.

In social media the platforms can cluster users based on their interactions and interests.

7. **Ordinal encoding** is used when the variables have natural order like satisfaction levels which can be “Not at all Satisfied”, “Partly Satisfied”, “Satisfied”, “More than Satisfied” and “Very Satisfied” and can be mapped to integer values 1, 2, 3, 4 and 5.

Another example for this, is education levels high school, bachelor's degree, master and Phd.

In One-hot encoding the categories can be converted into binary vectors with separate columns for each category. It is useful when there is no natural order between the categories. Colors is a good example for this method as there is no order between blue, red, green and yellow. Another example is car brands like Volvo, BMW and Toyota.

Dummy variable encoding is similar to one-hot encoding but it removes one category to prevent redundancy as this category can be predicted by seeing the others. This removal helps to have better model.

8. The interpretation of data as nominal or ordinal can depend on which context we are talking about. In the normal situation the colors do not have any inherent order but, if we define a order to the colors like Julia mentioned that the color red is the prettiest. In this way we ourselves defined the context so we can see a order for the colors. I think Julia is correct.
9. Streamlit is python library that can help us to build a web application for machine learning and visualization of data. We can create a web application as an interface where the user can interact with the application. This library is easy to use instead of creating a web application which requires knowledge about html, css and javascript. We can also upload our machine learning models in the application.

7 Självutvärdering

1. Utmaningar du haft under arbetet samt hur du hanterat dem.
Testing and training the model tooks long time (about 3 hours)
In addition I had to run the code several time in different times
2. Vilket betyg du anser att du skall ha och varför.
VG because I think I have done all the mentioned points
3. Något du vill lyfta fram till Antonio?

Appendix A

Källförteckning

[AntonioPrigmet/ds24_ml](#)

[scikit-learn: machine learning in Python — scikit-learn 1.6.1 documentation](#)

[IBM - United States](#)

[Chapter 2 : SVM \(Support Vector Machine\) — Theory | by Savan Patel | Machine Learning 101 | Medium](#)

[Random Forest Classifier using Scikit-learn - GeeksforGeeks](#)

[آموزش یادگیری ماشین — جامع و با مفاهیم کلیدی | فرادرس](#) (in persian)