Image Classifier

A classifier model that predicts handwritten digits.



Adrian Krsmanovic

EC Utbildning

Kunskapskontroll 2

2024–03

Abstract

The objective of this paper is to showcase the work behind developing a ML model that can classify images from the MNIST dataset as well as digits drawn by a user. This paper will include all the steps of the work that was done, from collecting data, testing different models and evaluating them, and at last using the best performing model in a streamlit application to predict on handwritten digits from the user. During the work some challenges appeared and will be discussed in this paper as well as further steps can be taken to develop this project even further. Overall, the project was successful, and the objectives were reached.

Abbreviations  
  
ML = Machine Learning

TP = True Positives

TN = True Negatives

FP= Fales Positives

FN= False Negatives

Innehåll

[Abstract 2](#_Toc161983001)

[1 Introduction 1](#_Toc161983002)

[1.1 Purpose 1](#_Toc161983003)

[1.2 Research Questions 1](#_Toc161983004)

[1.3 Limitations 1](#_Toc161983005)

[2 Theory 2](#_Toc161983006)

[2.1 Machine Learning 2](#_Toc161983007)

[2.2 Supervised Learning 2](#_Toc161983008)

[2.3 Evaluation metrics 2](#_Toc161983009)

[2.4 Data preprocessing 2](#_Toc161983010)

[2.5 Models 3](#_Toc161983011)

[2.5.1 Logistic Regression 3](#_Toc161983012)

[2.5.2 Support Vector Classifier 3](#_Toc161983013)

[2.5.3 RandomForestClassifier 3](#_Toc161983014)

[2.5.4 KnearestNeighbors 3](#_Toc161983015)

[2.6 Hyper parameters and GridSearchCV 4](#_Toc161983016)

[3 Method 5](#_Toc161983017)

[3.1 tools 5](#_Toc161983018)

[3.2 Data collection 5](#_Toc161983019)

[3.3 Exploratory data analysis (EDA) 5](#_Toc161983020)

[3.4 Data preprocessing 5](#_Toc161983021)

[3.5 Model training and testing 5](#_Toc161983022)

[3.6 Streamlit 5](#_Toc161983023)

[3.7 Image preprocessing 6](#_Toc161983024)

[4 Results and Discussion 7](#_Toc161983025)

[4.1 Results 7](#_Toc161983026)

[4.2 Discussion 7](#_Toc161983027)

[5. Conclusion 8](#_Toc161983028)

[5.1 Conclusion 8](#_Toc161983029)

[5.2 Further steps 8](#_Toc161983030)

[5.3 Challenges 8](#_Toc161983031)

[Teoretiska frågor 9](#_Toc161983032)

[Självutvärdering 11](#_Toc161983033)

[Appendix A 12](#_Toc161983034)

[Källförteckning 13](#_Toc161983035)

# 1 Introduction

In modern times many things are made simpler due to technologic advances. Machine learning is a big part of these advances and are integrated in many things we take for granted. For example, facial recognition which is standard on nearly every newer phone. In this project we will develop an image classifier using ML methods. To create a ML model the most important asset to have is data. Fortunately, I have access to the MNIST dataset which includes 70000 images of handwritten digits. MNIST is a dataset commonly used for benchmarking different models. Since this is an image classifier the possibilities are many. image classifiers are very common and can be used for medical purposes, traffic purposes and much more. Although I will not develop such an advanced image classifier, I will learn how to create a simpler image classifier that gives a basis on how to create image classifiers and ML models overall.

## 1.1 Purpose

The objective of this project is to develop a ML model that is capable of predicting handwritten digits with high accuracy, including those within the MNIST dataset as well as digits handwritten by users. To achieve this purpose, I will answer the following questions.

## 1.2 Research Questions

1: Can we predict MNIST data with close to 100% accuracy?

2: Does the model work on handwritten digits that are not part of the MNIST data?

## 1.3 Limitations

Given that this is an academic endeavor there is a time constraint. Therefore it is important to note that several steps, which could prove to significantly enhance this project’s capabilities was not performed. Despite the limitations this project aims to achieve satisfactory results. This leaves the project open for further development.

# 2 Theory

This chapter will address the theoretical concepts necessary to understand this work.

## 2.1 Machine Learning

“Machine Learning is the science (and art) of programming computers so they can

learn from data.” (Géron, 2019, p. 2). In simpler terms ML is the ability for a computer to learn without being explicitly programmed. This is the foundation which ML is based on and its important to understand before continuing with this paper.

## 2.2 Supervised Learning

In ML there are different types of tasks. This project tackles a supervised learning task.

“In supervised learning, the training set you feed to the algorithm includes the desired

solutions, called labels” (Géron, 2019, p. 8). In this case the labels are the digits and the features are the images of the handwritten digits.

## 2.3 Evaluation Metrics

When working with machine learning you often have different models to choose from. To evaluate different models you need to have a metric used to compare the models with. There are different metrics that can be used and below, 3 commonly used ones will be explained.

“**Recall** measures the proportion of actual positive cases correctly identified by the model. It answers the question: ‘Of all the actual positive instances, how many were correctly predicted by the model?’” (Bonnet, 2023, para. 6). Recall is useful when you want to make sure the model accurately predicts most of the positive class, even if you predict some false positives as well. The calculation of recall is shown on the bottom of the page.

“**Precision**, often referred to as the positive predictive value, quantifies the proportion of true positive predictions among all positive predictions made by the model” (Bonnet, 2023, para. 5). Precision is useful when there is a demand for the model to be accurate in its positive predictions. The calculation of precision is shown on the bottom of the page.

“**Accuracy** represents the ratio of correctly predicted instances to the total number of instances in the dataset” (Bonnet, 2023, para. 4) Accuracy is a useful metric when you want the model to perform well overall. The calculation of accuracy is shown on the bottom of the page.

## 2.4 Data preprocessing

**Standard Scaler** is a method from the scikit-learn library which standardizes features by removing the mean and scaling it to unit variance. This is used to ensure that the data have a balanced scale. (Scikit-learn documentation)

**PCA (**Principal Component Analysis) is a dimension reducing algorithm which according to Géron, is the most popular one (2019, 219). It identifies a hyper plane closest to the data and projects the data onto it. Using a PCA can be done, either by choosing the amount of dimensions to keep or the variance that is desired to be kept.

*2.4 -images before and after PCA*

En bild som visar text, handskrift, Teckensnitt, typografi

Automatiskt genererad beskrivning

## 2.5 Models

In this chapter I will dive into the 4 different models that were tried in the project.

### 2.5.1 Logistic Regression

**Logistic Regression** is a classification model that outputs a number between 0-1 which can be interpreted as the probability of an instance belonging to a class. Since this is a multiclass classification there are different methods to use, in this case I used OvR (One versus Rest), “It involves splitting the multi-class dataset into multiple binary classification problems. A binary classifier is then trained on each binary classification problem and predictions are made using the model that is the most confident.” (Brownlee, 2019, para. 4)

### 2.5.2 Support Vector Classifier

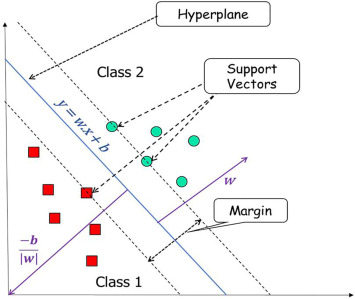
**SVC (Support Vector Classifier)** is a support vector machine for non-linear data. Support vector machines try to create, a as broad as possible separation between data and uses support vectors to separate. “They distinguish between two classes by finding the optimal hyperplane that maximizes the margin between the closest data points of opposite classes” (IBM, 2019, para. 1). If the data is not linearly separable a “kernel trick” can be used to make the data separable.

### 2.5.3 RandomForestClassifier

**RandomForestClassifier** is an ensemble of Decision Trees. The predictions from each decision tree are computed and a final decision is provided. How the decisions are computed can variate but the standard way is that the majority of predictions wins and it is the final prediction.

### 2.5.4 KnearestNeighbors

**“KNearestNeighbors**is a Machine Learning algorithm that uses the similarity between our data to makeclassifications” (Miguel, 2021, para. 1) In simpler terms, the KnearestNeighbors classifier classifies instances based on their closest neighbors.

*2.5.2 – Example of how a Support Vector Machine works*

## 2.6 Hyper parameters and GridSearchCV

ML models often requires different hyperparameters to be optimized. Hyperparameters are values that determine how a model learns.” Hyperparameters are parameters whose values control the learning process and determine the values of model parameters that a learning algorithm ends up learning” (Nyuytiymbiy, 2020, para. 2). GridSearchCV is used to search over a grid of Hyperparameters and find the optimal ones using cross validation.

# 3 Method

This chapter will show what methods and tools were used in the project.

## 3.1 Tools

This project was developed using python with the scikit-learn library for the ML models and splitting the data. Numpy was used for handling data and matplotlib for visualizing the data. Creating an application was done by using the package streamlit. When predicting on handwritten digits, Image and ImageOps was used to format the pictures.

## 3.2 Data collection

The data was collected from the famous MNIST dataset which contains a total of 70000 images of handwritten images. This is a well known dataset often use to benchmark different models.

## 3.3 Exploratory data analysis (EDA)

Using the MNIST.DESCR function I learn that the digits are in a 20x20 pixel box centered on a 28x28 black background. By using matploblib I am able to see how the MNIST images look and also plot a colorbox which gives me valuable information about the colors of the images.

## 3.4 Data preprocessing

The first step of processing the data is to split it into y and X, in this case X is the image and y is the corresponding digit. Then as per usual in ML, the data is split into training, validation, and test sets. Because the dataset is quite large a smaller dataset containing of 10000 images and values was created to use when finding the optimal hyperparameters using scikit-learns GridSearchCV. To standardize the data and reduce dimensions a pipeline containing StandardScaler and PCA was applied. This was done while keeping a 95% variance to ensure the variance is kept after the PCA transformation.

## 3.5 Model training and testing

To find the best performing model 4 different models were tested. To find the optimal hyper parameters for the models GridSearchCV was used over a created grid of different hyperparameters. First a simple logistic regression, then an SVC model, a RandomForestClassifier and lastly a KNeighborsClassifier. Using accuracy as the metric to compare the models, since the goal is for the model to perform well overall, the SVC model performed the best out of the 4 models and therefore it was trained on the training and validation data and evaluated on the test data. The SVC model with gamma=’poly’, C=0.5 and gamma=1 which was the optimal hyperparameters was then saved using joblib, as well as the pipeline which will be used later for image processing.

## 3.6 Streamlit

Using the python library “Streamlit” an application was created with a single page where the user can draw a digit on a canvas and get a prediction from the model. The user can also choose to view the code that is used to process the images.

## 3.7 Image preprocessing

Using the python library “Image” and “ImageOps” the handwritten digits are formatted to look the same as the images the model is trained on. Firstly, the canvas is converted to 28x28 pixels image and by using “LANCZOS” good quality is ensured. Then the Image is converted to grayscale and finally a numpy array. The array is then flattened and reshaped to a 1x1 array to fit the pipeline. Using joblib the pipeline is loaded and applied to the image. Lastly the model that is also loaded using joblib predicts on the image and gives the user its prediction.

*3.6- Streamlit Application*

En bild som visar text, skärmbild, design

Automatiskt genererad beskrivning

|  |  |
| --- | --- |
| **Accuracy for different models** | |
| SVC | 98% |
| KNeighborsClassifier | 95% |
| RandomForestClassifier | 94% |
| LogisticRegression | 92% |

*4.1-Accuracy for models*

# 4 Results and Discussion

This chapter will cover the results and discuss what worked well during the project and what did not work equally well.

## 4.1 Results

As we can see the SVC model performed better than the rest, this is not surprising since support vector machines are known to work well on complex data such as images. This is why I chose to use the SVC even though it requires a lot of training time. In the end we can see it was a good decision since it achieved 3% higher accuracy then the KNeighborsClassifier. Using the MNIST.DESCR function I learned that the digits are in a 20x20 pixel box centered on a 28x28 pixel black background. Using matplotlib to plot a digit and a colorbox I learned that each pixel is between 0-255 where 255 is the whitest and 0 the blackest. This proved to be very useful knowledge when processing handwritten digits since I needed to format the images to match the ones the model was trained on. Applying all the processing steps to the canvas after a digit is drawn the model can accurately predict the digit.

## 4.2 Discussion

It proved very valuable to create a smaller set of data to find optimal hyperparameters, the time to search for optimal hyperparameters was reduced significantly and the performance was still satisfactory. Applying a PCA to reduce the dimensions also sped up the training time and the results were not affected. The model performed very well on the MNIST data reaching a 98% accuracy on unseen images. When predicting on handwritten digits it also performed good, but it is very sensitive to rotation and centering. The results still satisfy our goals, and it is a well working classifier. As we can see below when the digit is centered and upright the model correctly predicts a 4, when the digit is slightly rotated it instead predicts a 7.

En bild som visar symbol, text, skärmbild, design

Automatiskt genererad beskrivningEn bild som visar skärmbild, symbol, vit, linje

Automatiskt genererad beskrivning *application*

*4.2 Images from streamlit*

# 

# 5. Conclusion

In this chapter we will make a conclusion of the project and discuss what further steps can be taken. As well as challenges that appeared during the project.

## 5.1 Conclusion

In conclusion we were able to create an image classifier that successfully is able to classify unseen images. However, it is crucial that the image is in the same format as the training data, and also centered. This leaves room for improvement. To conclude we will answer our research questions, **can we predict MNIST data with close to 100% accuracy?** This was achieved since we reached a 98% accuracy which is close to 100%.

**Can the model predict handwritten digits with relatively high accuracy?** The model quite confidently predicts most handwritten digits, but as mentioned it is very sensitive to rotation and centering, so it is important the user follows the instructions in the streamlit application for the model to work as intended.

## 5.2 Further steps

With reaching a 98% accuracy on the MNIST data there is not much room for improvement. Although when predicting on handwritten digits the model performed well but as mentioned before it is sensitive to rotation and needs to be centered to work well. To improve this the handwritten digits would need to be centered and this could be done by calculating the center of mass and center the digit in the middle of the image. To make the model work better on rotated digits a possible solution would be to rotate a portion of the MNIST data used for training and this would potentially make the model more robust and able to predict better on rotated images. There are more steps that can be taken to improve the model and any reader is more than welcome to try this out.

## 5.3 Challenges

During this project there occurred some challenges. The biggest one was to format the handwritten digits for the model to predict on. For this the EDA proved valuable because to be able to format the images we firstly need to understand how the data used for training the model looks. The size of the dataset also proved to be a challenge since it meant long training times and hyperparameter tuning. The smaller dataset of 10000 instances proved crucial since it significantly sped up training and hyperparameter tuning. Using PCA to reduce dimensions also reduced the training time while keeping a good performance.

# Teoretiska frågor

1. **Kalle delar upp sin data i ”Träning”, ”Validering” och ”Test”, vad används respektive del för?**

Träningsdata används för att träna olika modeller samt hitta optimala hyperparametrar.

Valdieringsdata används för att utvärdera förmågan av olika modeller och hitta den bäst presterande.

Testdata används för att testa den valda modellen på osedd data och få en känsla av hur den presterar på ny data.

1. **Julia delar upp sin data i träning och test. På träningsdatan så tränar hon tre modeller; ”Linjär Regression”, ”Lasso regression” och en ”Random Forest modell”. Hur skall hon välja vilken av de tre modellerna hon skall fortsätta använda när hon inte skapat ett explicit ”validerings-dataset”?**

Det kan hon göra genom att använda en cost function och välja den modell som presterar bäst. t.ex. om RMSE väljs är den modellen med minst RMSE den modell som vars prediktioner är närmast de sanna data punkterna.

1. **Vad är ”regressionsproblem? Kan du ge några exempel på modeller som används och potentiella tillämpningsområden?**

Regressionsproblem är maskininlärning problem där y värdet kan anta kontinuerliga värden. Prediktera någons lön baserat på ålder och erfarenhet är ett exempel på ett regressionsproblem då lön kan anta kontinuerliga värden.

1. **Hur kan du tolka RMSE och vad används det till:**

**=√Σ(𝑦𝑖−𝑦̂𝑖)2𝑖**

RMSE står för Root Mean Squared Error och kan tolkas som medelavståndet från en models prediktioner till de sanna data punkterna. RMSE används ofta som en cost function i reggresionsproblem för att kunna jämföra modellers prestanda.

1. **Vad är ”klassificieringsproblem? Kan du ge några exempel på modeller som används och potentiella tillämpningsområden? Vad är en ”Confusion Matrix”?**

Klassifcieringsproblem är maskininlärningsproblem där y värdet kan anta diskreta värden. Vanligt förekommande modeller är RandomForestClassifier och SupportVectorClassifier. De kan användas om man t.ex. baserat på features ska klassa något som man eller kvinna. En confusion Matrix är en matris där man kan avläsa antalet korrekta prediktioner och felaktiga prediktioner.

1. **Vad är K-means modellen för något? Ge ett exempel på vad det kan tillämpas på.**

K-means modellen används för att klustra olika grupper av data. Den är vanligt förekommande vid unsupervised learning problem där datan saknar ett y-värde. Om man t.ex. har olika kunder man vill gruppera i olika grupper kan K-menas vara en passande modell.

1. **Förklara (gärna med ett exempel): Ordina encoding, one-hot encoding, dummy variabler encoding. Se mappen ”l8” på GitHub om du behöver repetition.**

**Ordinal Encoding** ger en siffra till varje kategoriskt värde, så t.ex. blir hund 1, katt blir 2, fågel blir 3 osv.

**One-hot-encoding** ger varje kategoriskt värde antingen 0 eller 1, t.ex. blir hund [0, 0, 1], katt blir [0, 1, 0] och fågel blir [1, 0, 0]

**Dummy variable encoding** gör samma sak som one-hot-encoding men gör en kategori till ”baseline”, det innebär att om katt är [0 ,1] och hund är [1, 0] blir fågel [0, 0] (baseline) i stället för att skapa en ny etta och då få 3 siffror, detta sparar tid och utrymme.

1. **Göran påstår att datan antingen är ”ordinal” eller ”nominal”. Julia säger att detta måste tolkas. Hon ger ett exempel med att färger såsom {grön, röd, grön} generellt sett inte har någon inbördes ordning (nominal) men om du har en röd skjorta så är du vackrast på festen (ordinal) – vem har rätt?**

Julia har rätt. Eftersom för sig själv saknar färgen en naturlig rangordning men beroende på vad färgen används till kan den anta en naturlig rangordning.

1. **Kolla följande video om Streamlit: https://www.youtube.com/watch?v=ggDa-RzPP7A&list=PLgzaMbMPEHEx9Als3F3sKKXexWnyEKH45&index=12 Och besvara följande fråga: - Vad är Streamlit för något och vad kan det användas till?**

Streamlit är ett open source python library som gör det möjligt att skapa applikationer där man t. ex som i vårt kan ladda upp en ML model och låta användaren skriva in värden för modellen att prediktera på.

# Självutvärdering

1. Utmaningar du haft under arbetet samt hur du hanterat dem.

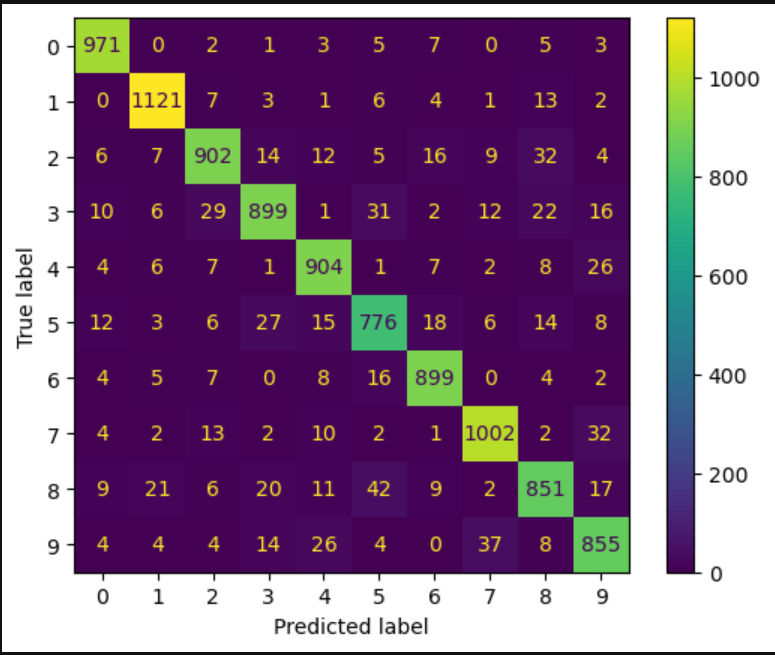
Den största utmaning under arbetet har varit att formatera bilder så att modellen kan prediktera på de. Genom att verkligen förstå hur MNIST datan ser ut kan man lättare lösa problemet. Även genom att löpande under processen av att formatera bilder printa ut så man kan se hur bilden ser ut jämfört med MNIST datan. Att skriva rapporten har också varit utmanande eftersom jag inte är van vid sättet en rapport skrivs på, men efter diskussioner med klasskamrater och med hjälp av guiden som fanns har det gått bra. Det har uppstått fler utmaningar men med hjälp av boken, genomgångar under lektionerna och diskussioner har det kunnat lösas.

1. Vilket betyg du anser att du skall ha och varför.   
   Jag anser att jag bör få VG då jag först har lyckats med att skapa en välfungerande modell samt ladda upp den till en streamlit applikation som var del av kraven för VG. Jag har även löst problem genom att använda olika metoder inom maskininlärning som t.ex. PCA för att dimensionsreducera och snabba upp träningen. Slutligen tycker jag att jag har redovisat för mitt modellval på ett tydligt och logiskt sätt.
2. Något du vill lyfta fram till Antonio?

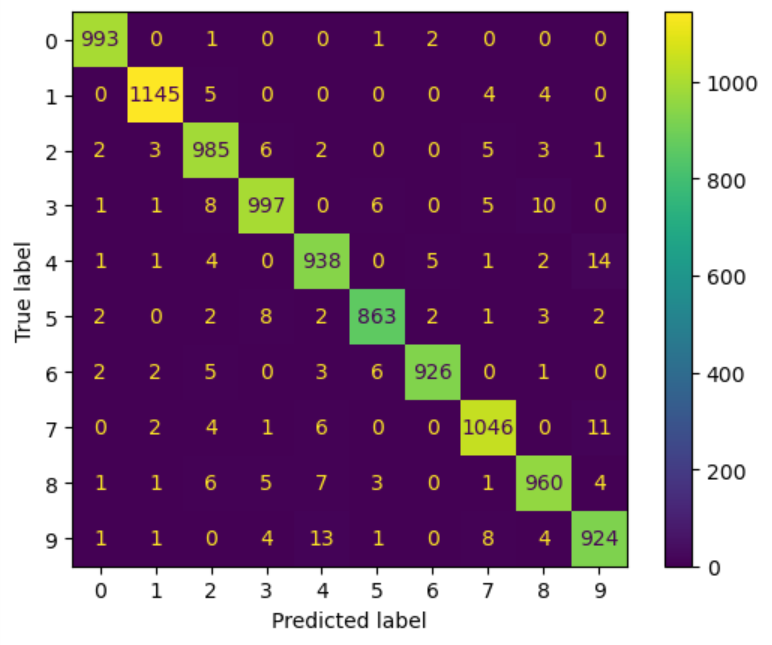
Nej, endast att det var en rolig och väldigt utmanande uppgift!

# Appendix A

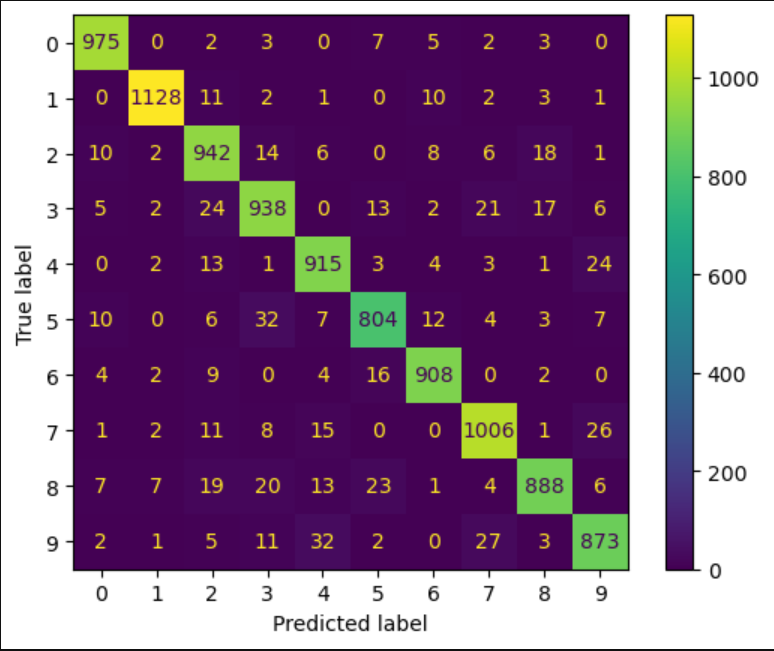
Below Confusion Matrixes for the tried models will be included to further strengthen the choice of model.



Figur 1, Confusion Matrix for LogisticRegression



Figur 2, Confusion Matrix for SVC model



Figur 3, Confusion Matrix for RandomForest

En bild som visar skärmbild, text, kvadrat, diagram

Automatiskt genererad beskrivning

Figur 4, Confusion Matrix for KNeighborsClassifier

# Källförteckning

Géron, A. 2019. In: Roumeliotis, R & Tache, N. (Eds.) *Hands-On Machine Learning with Scikit-learn, Keras, and TensorFlow.* Page 2&8.Canada: O’Reilly Media, inc.

Bonnet, A. (2019). *Accuracy vs Precision vs Recall in Machine Learning: What is the difference?* Para. 4–6. <https://encord.com/blog/classification-metrics-accuracy-precision-recall/>

Brownlee, J. (2021). *One-vs-Rest and One-vs-One for Multi-Class Classification*. Para 4. <https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/>

IBM. (2019). *What are support vector machines (SVMs)?* Para. 1. <https://www.ibm.com/topics/support-vector-machine>

Miguel, L. (2021). *KNN (K Nearest Neighbors) and KNeighborsClassifier — What it is, how it works, and a practical example!* Para. 1. <https://luis-miguel-code.medium.com/knn-k-nearest-neighbors-and-kneighborsclassifier-what-it-is-how-it-works-and-a-practical-914ec089e467>

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, PP 2825-2830, 2011. <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>

[Nyuytiymbiy](https://kizitonyu.medium.com/?source=post_page-----aa609601a9ac--------------------------------), K. (2020). *Parameters and Hyperparameters in Machine Learning and Deep Learning.* Para. 2. https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac