**Title Page**

**Report Title**



**Name: Muhannad Naser**

**Report for kunskapskontroll 2**

Date:21/03/2024

# Abstract

This study explains the process of developing, assessing, and implementing models by delving into the core ideas and useful applications of machine learning. The author starts by looking at data partitioning strategies and then explores the functions of test sets, validation, and training in model training and evaluation. The sections that follow examine classification and regression issues presenting a range of models and their practical uses. The author also looks into categorical data encoding methods and provides Streamlit a flexible tool for building interactive online apps. Lastly, the use of neural networks and support vector machines to recognize handwritten digits is a way to show how machine learning may be applied practically. Through this thorough investigation, readers will obtain important knowledge about machine learning theory and application, enabling them to take advantage of this game-changing technology across a variety of industries.

Table of Contents

[Abstract 2](#_Toc161895724)

[List of Figures 5](#_Toc161895725)

[Introduction 6](#_Toc161895726)

[1. Question 1: ML Subsets 6](#_Toc161895727)

[1.2. Validation set 6](#_Toc161895728)

[1.3. Test set 7](#_Toc161895729)

[2. Question 2: Cross-Validation 7](#_Toc161895730)

[3. Question 3: Regression Problem 7](#_Toc161895731)

[4. Question 4: RMSE 8](#_Toc161895732)

[4.1. RMSE Interpretation 9](#_Toc161895733)

[4.2. Objective 9](#_Toc161895734)

[5. Question 5: Classification Problem 9](#_Toc161895735)

[5.1. Definition 9](#_Toc161895736)

[5.2. Examples 9](#_Toc161895737)

[5.3. Confusion Matrix 10](#_Toc161895738)

[6. Question 6: K-Mean Algorithm 10](#_Toc161895739)

[6.1. K-mean 10](#_Toc161895740)

[6.2. Examples 11](#_Toc161895741)

[7. Question 7: Methods for Transforming Categorical Variables 11](#_Toc161895742)

[7.1. Ordinal Encoding 11](#_Toc161895743)

[7.2. Categorical Variables 12](#_Toc161895744)

[7.3. Dummy Variables 12](#_Toc161895745)

[8. Question 8: Ordinal and Nominal 12](#_Toc161895746)

[9. Question 9: Streamlit 13](#_Toc161895747)

[10. Question 10: Handwritten Digit Recognition Using Neural Networks and SVM 14](#_Toc161895748)

[10.1. Introduction 14](#_Toc161895749)

[10.2. Methodology 14](#_Toc161895750)

[10.2.1. Data Acquisition and Preprocessing 14](#_Toc161895751)

[10.2.2. Neural Network Approach 15](#_Toc161895752)

[10.2.3. SVM Approach 16](#_Toc161895753)

[Conclusion 18](#_Toc161895754)

[References 19](#_Toc161895755)

# List of Figures

[Figure 1: Confusion Matrix 9](#_Toc161895716)

[Figure 2: Data Loading and Normalization 14](#_Toc161895717)

[Figure 3: Neural Network Architecture 14](#_Toc161895718)

[Figure 4: Model Settings 15](#_Toc161895719)

[Figure 5: Model Accuracy 15](#_Toc161895720)

[Figure 6: SVM model 16](#_Toc161895721)

[Figure 7: Classification Report 16](#_Toc161895722)

[Figure 8: Prediction 17](#_Toc161895723)

# Introduction

Machine learning is a field that is transforming industries enabling intelligent systems, and helping to solve the puzzles associated with large and complicated datasets. This report covers the fundamental ideas and real-world applications of machine learning (ML) which emphasizes clarity and conciseness, provides a thorough overview of machine learning algorithms and their practical applications for both inexperienced and seasoned practitioners.

The sections that follow explore regression and classification issues and give instances of models and how they are used in various fields. After examining encoding methods for handling categorical data, Streamlit a potent tool for building interactive web applications is introduced. The report's practical demonstration of handwritten digit recognition utilizing support vector machines and neural networks at the end illustrates the entire model development, testing, and deployment process. By following this methodical process, readers will get a thorough comprehension of machine learning principles and methods, enabling them to fully utilize this revolutionary technology.

# Question 1: ML Subsets

To facilitate machine learning Kalle separates data into three subsets: training, validation, and test sets. Each subset has a particular function in the workflow.   
1.1. Training set

The machine learning model is mostly trained using the training set which makes up the majority of the data. The model learns the correlations, patterns, and features found in the data through iterative training. The model optimizes its performance by minimizing prediction errors by adjusting its internal parameters as it is exposed to a wide variety of instances from the training set (Vabalas et al., 2019).

## Validation set

The validation set is essential for adjusting the hyper-parameters of the model and evaluating its performance. The validation set is not directly utilized to update the model's parameters in contrast to the training set. Rather, it functions as a separate dataset for assessing how well the model performs with hypothetical cases. Practitioners can make well-informed decisions about hyper-parameter tweaking, model selection, and preventing over-fitting by keeping an eye on the model's performance on the validation set (Vabalas et al., 2019).

## Test set

The test set is the last yardstick for evaluating the effectiveness of the trained model. It offers a dispassionate assessment of the model's capacity to generalize to fresh, untested data sets. The test set is not modified while the model is being developed or the hyper-parameters adjusted guaranteeing that the assessment outcomes are unaffected by the training procedure. Practitioners can assess the model's performance in the actual world and ascertain whether it satisfies the required standards for accuracy and dependability by testing it against the test set (Raschka, 2018).

# Question 2: Cross-Validation

If Julia hasn't produced a different validation dataset she can assess how well the three models ("Linear Regression," "Lasso Regression," and a "Random Forest model") perform on the training data by using a method known as cross-validation. In cross-validation the training data is divided into several subsets. The models are then trained on various combinations of these subsets and their performance is assessed on the remaining data.  
K-fold cross-validation is a popular cross-validation technique in which the training data is split into k subsets or folds. Using a separate fold as the validation set and the remaining data as the training set the model is trained k times. An estimate of the model's performance that is more reliable is then obtained by averaging the performance measures from each fold. Julia may select the model that performs best on the training set by comparing the average performance measures (such as accuracy, F1-score, and RMSE) of the three models following cross-validation. Using the test dataset the selected model can then be further assessed and adjusted as needed (Nti et al., 2021).

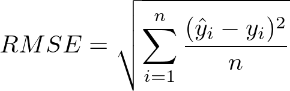
# Question 3: Regression Problem

Predicting a continuous outcome variable from one or more input factors is the goal of regression problems. To create a relationship between the continuous target variable and the input variables is the aim. The linear relationship between the input features and the target variable is assumed by the widely used linear regression model. For example, the price of a property based on its location, square footage, and number of bedrooms may be predicted using linear regression. This model sheds light on the relationship between changes in the independent and dependent variables (Maulud & Mohsin Abdulazeez, 2020).

By adding polynomial terms to the input features, polynomial regression expands on linear regression and makes it possible to capture non-linear correlations between variables. It is especially helpful in situations when there isn't a strictly linear relationship between the variables, like in quadratic or cubic relationships. Furthermore, regularized variants of linear regression such as Lasso Regression and Ridge Regression add penalty terms to the loss function to assist prevent over-fitting. While Lasso Regression can result in feature selection by decreasing the coefficients of less significant features to zero, Ridge Regression is appropriate when multicollinearity is present in the data. Regression tasks also use Decision Trees and Random Forests which divide the feature space into regions and predict values based on the average of the target variable in each region (Rong & Bao-Wen, 2018). These models are resistant to data outliers and can represent intricate interactions between variables. Regression models have a wide range of potential uses in fields like environmental research, marketing, and finance. Regression models can be used to evaluate the impact of marketing campaigns on client acquisition and retention, forecast product sales based on advertising expenditure and seasonality, and predict stock prices based on historical data and market indicators. Regression models can also be used to anticipate energy consumption based on past data and meteorological conditions, as well as to predict the price of used cars based on their age, mileage, and other characteristics.

# Question 4: RMSE

A regression model's performance can be assessed using the Root Mean Squared Error (RMSE) which calculates the average magnitude of errors between the target variable's actual values (𝑦𝑖) and predicted values (𝑦̂𝑖) for all observations (Yin et al., 2023). The RMSE formula is:

​

## 4.1. RMSE Interpretation

The square root of the average squared difference between the actual and anticipated values is what RMSE stands for. It indicates how widely distributed the residuals that is the variations between actual and expected values are. Better performance is implied by a smaller RMSE which shows that the model's predictions are closer to the real values (Yin et al., 2023).

4.2. Objective

The root mean square error (RMSE) serves as a gauge for how well a model predicts continuous events. By quantifying the typical divergence between the expected and actual values it offers information about how well the model predicts values. Because of the squaring process which penalizes greater errors more severely. RMSE helps highlight the significance of avoiding large prediction errors.

# Question 5: Classification Problem

## 5.1. Definition

Predicting discrete categories or labels for input data based on its attributes is the task of classification problems. The output variable in these questions is categorical which means that it belongs to a limited number of classes or categories. Creating a model that can precisely classify new instances into existing ones is the aim of classification models. Several models, such as logistic regression, decision trees, random forests, and support vector machines (SVM) are frequently employed for classification problems. To predict the class labels of new instances these models utilize patterns and relationships found in the data (Palanivinayagam et al., 2023).

## Examples

Classification models have potential uses in many different fields including picture recognition, sentiment analysis, email spam detection, and medical diagnosis. Classification models can determine a patient's likelihood of having a specific disease based on their medical history and symptoms when it comes to medical diagnosis. Models used in email spam detection determine if an email is spam or not by looking at its metadata and content. Analyzing text data for instance reviews or messages on social media to ascertain their sentiment (positive, negative, or neutral) is known as sentiment analysis. Classification models in image recognition recognize objects or patterns inside images an example of this would be the MNIST dataset's handwritten digit classification (An et al., 2023).

## Confusion Matrix

A table that makes it possible to see how well a classification method is performing is called a confusion matrix. It displays a summary of a classifier's predictions about the test set's actual labels. The matrix is arranged in rows and columns where the actual class is represented by each row and the anticipated class by each column. True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) are the primary elements of a confusion matrix. These elements offer insightful information about the classifier's performance and enable the computation of several metrics, including accuracy, precision, recall, and F1-score, which aid in evaluating how well the model distinguishes between classes (An et al., 2023).



Figure 1: Confusion Matrix

# Question 6: K-Mean Algorithm

## 6.1. K-mean

A well-liked unsupervised machine-learning technique for grouping data into discrete groups or clusters according to feature similarities is the K-means model. The input data is divided into K clusters using the K-means clustering technique. The mean of all the data points allocated to each cluster serves as the cluster's centroid representing the cluster. To minimize the within-cluster variation, the algorithm iteratively assigns data points to the closest centroid and updates the centroids until convergence (Saputra et al., 2023).

## Examples

Marketing segmentation of customers is one example of how the K-means model is applied. Assume a retail corporation wants to better target its marketing efforts and promotions by segmenting its consumer base. Through the use of K-means clustering to client data including demographics, purchase history, and browsing behavior the business can discern different consumer categories based on shared purchasing habits or preferences. These market categories could consist of budget-conscious clients, infrequent shoppers, and regular buyers among others. The business may enhance customer happiness and retention, personalize marketing strategies, and optimize product offers by comprehending the traits and behaviors of each segment (Ali & Sheng-Chang, 2020).

# Question 7: Methods for Transforming Categorical Variables

Frequently employed methods for transforming categorical variables into a machine learning algorithm-processable format are ordinal encoding, one-hot encoding, and dummy variable encoding.

## 7.1. Ordinal Encoding

Ordinal encoding entails allocating a distinct integer value to every distinct category within a categorical variable. The given integers indicate the categories' rank or order but they may not always suggest a particular numerical relationship between them. Taking the category "education level" as an example has three categories: "high school," "college," and "graduate." These categories could be encoded as 1, 2, and 3, respectively using ordinal encoding. Even though ordinal encoding maintains the ordinal link between categories not all algorithms will benefit from it particularly those that use the assumption that features have numerical correlations (Dahouda & Joe, 2021).

## Categorical Variables

Categorical variables are converted into binary vectors using one-hot encoding commonly referred to as one-of-K encoding. Each category is represented by a binary flag (0 or 1). Each distinct category in this encoding technique generates a new binary feature and only one of these binary features is "hot" (set to 1) for each observation signifying the category's existence. For instance, the categories "high school," "college," and "graduate" would be represented as [1, 0, 0], [0, 1, 0], and [0, 0, 1], respectively, using one-hot encoding for the "education level" variable. Although one-hot encoding guarantees that every category is handled independently and uniquely, it can result in high-dimensional feature spaces particularly when working with variables that have several categories (Dahouda & Joe, 2021).

## Dummy Variables

Similar to one-hot encoding, dummy variable encoding is usually applied to categorical variables (binary variables) that have just two distinct categories. To indicate the presence or absence of one category entails establishing a single binary feature or dummy variable with the absence of the first category implicitly representing the other category. Dummy variable encoding would produce a single binary feature indicating the presence of one variable such as "male" = 1, "female" = 0 vice versa if had a binary variable "gender" with categories "male" and "female." Compared to one-hot encoding dummy variable encoding is helpful for binary variables and can aid in reducing multicollinearity in regression models (Dahouda & Joe, 2021).

# Question 8: Ordinal and Nominal

Göran's claim that data is either "ordinal" or "nominal" is simplistic in the context of data analysis. Julia's viewpoint which necessitates interpretation is more true and indicative of the intricacy of data found in the real world (Liu et al., 2020).

Categorical data with a natural order or ranking within its categories is referred to as ordinal data. Rating scales (such as "low," "medium," and "high") and survey results (such as "strongly disagree," "disagree," "neutral," "agree," and "strongly agree") are a few examples. Although colors like red, green, and blue are normally thought of as nominal categories with no intrinsic order in Julia's example of clothing the context of wearing a red shirt to be the most attractive at a party introduces an ordinal element. But this interpretation depends on the circumstances and isn't always relevant to all color-related scenarios. Contrarily, nominal data describes categorical information that lacks a natural ranking or order within its categories. Examples include ethnicity (e.g., "Asian," "Black or African American," "Hispanic or Latino," "White"), gender (e.g., "male," "female," "non-binary"), and fruit varieties (e.g., "apple," "banana," "orange").

In summary, by restricting data types to being strictly ordinal or nominal Göran's statement oversimplifies the complexity of data types. As demonstrated by the example of colors Julia's viewpoint appreciates the complexities involved in reading different data kinds and the possibility that the distinction may not always be obvious. As a result Julia's interpretation better captures the context-dependency and unpredictability that are frequently present in real-world data processing.

# Question 9: Streamlit

A Python package called “Streamlit” is used to create interactive online apps for data science and machine learning tasks. By enabling developers to create dynamic web apps with simple Python scripts it streamlines the process of developing user-friendly interfaces. With Streamlit developers don't need to be experts in web development tools like HTML, CSS, or JavaScript to create visually beautiful and user-friendly dashboards, visualizations, and applications. For data scientists and machine learning engineers who wish to share their models, analyses, and visualizations with others in an approachable and user-friendly manner Streamlit is very helpful. Because it makes rapid prototyping and iteration easier developers may create and launch web applications more quickly and showcase their work or provide stakeholders with information.

Streamlit's typical use cases include:

* Developing web-based interfaces for machine learning models that enable users to input data and view predictions in real-time.
* Building interactive dashboards to visualize data and insights.
* Developing data exploration tools to interactively explore datasets and perform analysis.
* Disseminating data science projects and findings in an easily comprehensible format to coworkers, clients or the general public.

# Question 10: Handwritten Digit Recognition Using Neural Networks and SVM

## 10.1. Introduction

A fundamental job in computer vision handwritten digit identification has several applications, including self-driving automobiles, check processing, and postal automation. Handwritten numerals must be sorted into the appropriate digital form (0–9). This section investigates neural networks and support vector machines (SVM) two machine learning methods for handwritten digit recognition. To train and assess models use the MNIST dataset which is a well-liked standard for image classification tasks and to evaluate each method's performance in handwritten digit recognition through this experiment (Lejeune, 2020).

## Methodology

The goal of this study is to determine how well Support Vector Machines (SVM) and Neural Networks perform in handwritten digit recognition. Both models are trained and assessed using the MNIST dataset as the basis. Below is a thorough explanation of the methodology:

### Data Acquisition and Preprocessing

* MNIST Dataset: This collection of 70,000 grayscale images shows handwritten numbers (28 × 28 pixels) with labels ranging from 0 to 9. To load the data, use the tf.keras.datasets.mnist function from the TensorFlow package. MNIST data is split by default into training (60,000 images) and testing (10,000 images) sets
* Normalization: The photos' pixel intensities fall between 0 and 255. The method uses tf.keras.utils.normalize to normalize the pixel values between 0 and 1 in order to guarantee greater convergence during training.

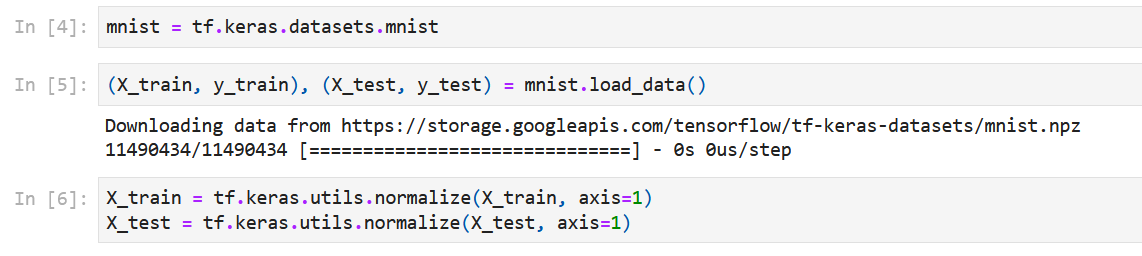


Figure 2: Data Loading and Normalization

### **Neural Network Approach**

Architecture Model

Keras's high-level neural network building API is used to create a sequential model. The 28x28 image is first flattened into a vector of 784 elements each of which represents an intensity of pixel. Two hidden dense layers each containing 128 neurons have been added. These layers use the Rectified Linear Unit (ReLU) activation function to add nonlinearity to the model. ReLU enables the model to discover intricate patterns in the data. Ten neurons make up the output layer one for each digit class (0–9). This layer generates probability distributions across the digit classes using the softmax activation function. The output shows how likely it is that each digit class will include the supplied image (Bowers et al., 2023).

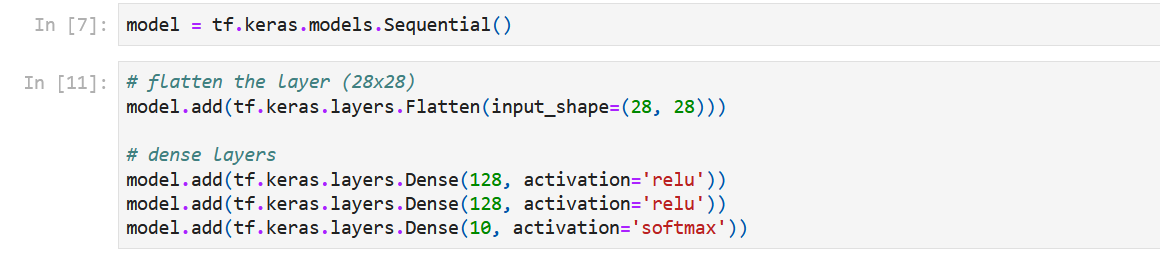


Figure 3: Neural Network Architecture

|  |  |
| --- | --- |
| Model Attributes | Values |
| Optimizer | Adam |
| Loss Function | Crossentropy loss |
| Evaluation Metrics | Accuracy |

Due to its effectiveness in updating model weights during training the Adam optimizer was selected. The loss function employed is the sparse categorical crossentropy. For every image this function calculates the difference between the true label distribution and the predicted probability distribution. The main statistic used to assess how well the model performs in accurately categorizing digits is accuracy.

For a predetermined number of epochs the model is trained on the training set. The model processes batches of images at each epoch, determines the loss, and modifies its internal parameters to reduce the loss. For the predetermined number of epochs, this process is repeated (Kwabena Patrick et al., 2022).

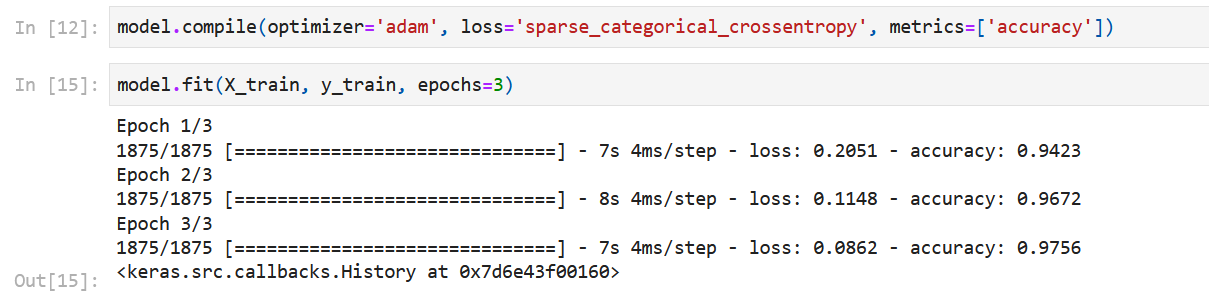


Figure 4: Model Settings

The model's performance is assessed on the test set that hasn't been viewed yet after training. For every test image the model predicts the digit class the accuracy of correctly classified digits is then computed. The accuracy of the model is 97.18%.

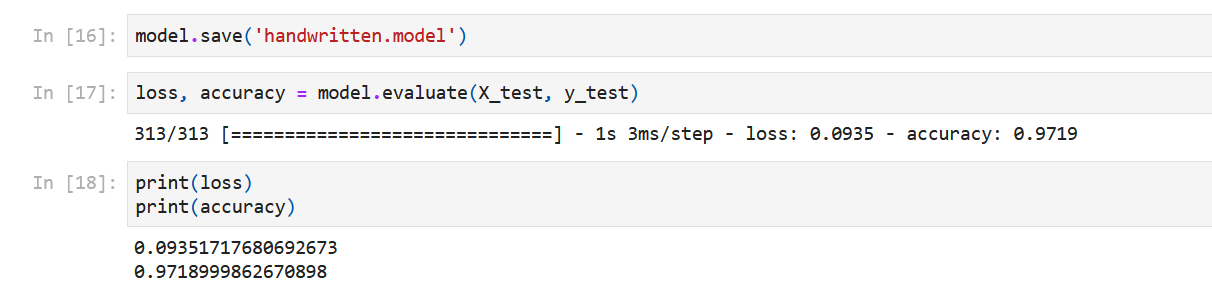


Figure 5: Model Accuracy

### SVM Approach

#### Model Selection and Hyper-parameter Tuning

The alternate method of choice is a Support Vector machine (SVM) classifier. SVMs search the feature space for a hyper-plane that maximizes the margin between classes. An important hyper-parameter that regulates the impact of training data points is the SVM model's gamma parameter. To prevent over-fitting the code sets gamma to a low value (0.001). It is frequently necessary to conduct experiments using methods like grid search or randomized search in order to determine the ideal hyper-parameter value. Using the scikit-learn library the SVM model is trained on the training set of data. In the high-dimensional feature space the model gains the ability to distinguish between data points that belong to various digit classes (Kaddoura et al., 2022).

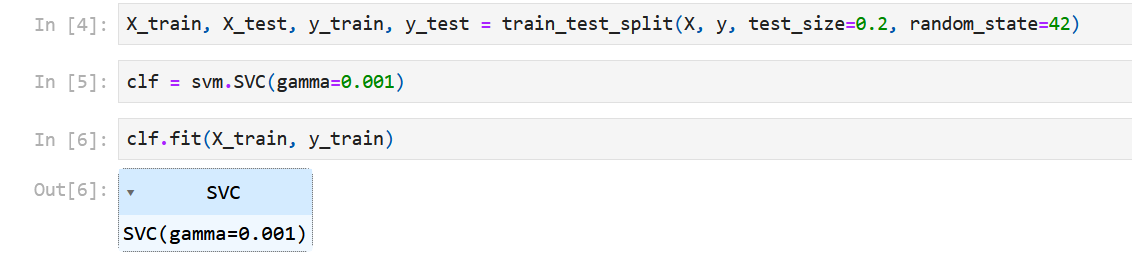


Figure 6: SVM model

On the test set the trained SVM model is assessed same like in the Neural Network method. For every test image the model predicts the digit class and classification metrics such as precision, recall, and F1-score are computed these metrics give a more comprehensive insight of the model's performance.

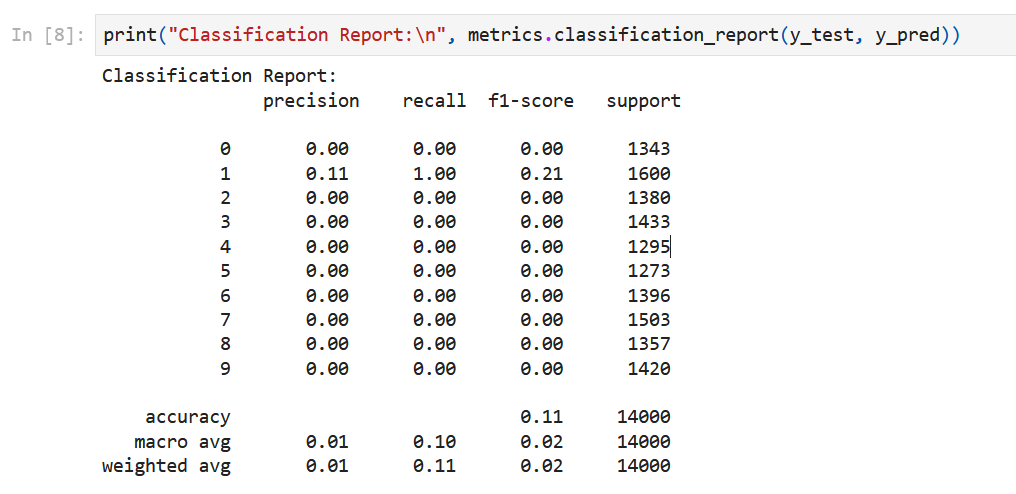


Figure 7: Classification Report

#### Model Prediction Functionality

The project additionally showcases the trained Neural Network model's image prediction capabilities. Users can submit image files with handwritten numbers on them. Assuming the uploaded image is properly named and located the algorithm successfully analyzes it. It then preprocesses it (reading the grayscale channel, inverting it), and feeds the processed data to the model for prediction. The most likely digit class for example, "This digit is probably 6" is predicted by the model.

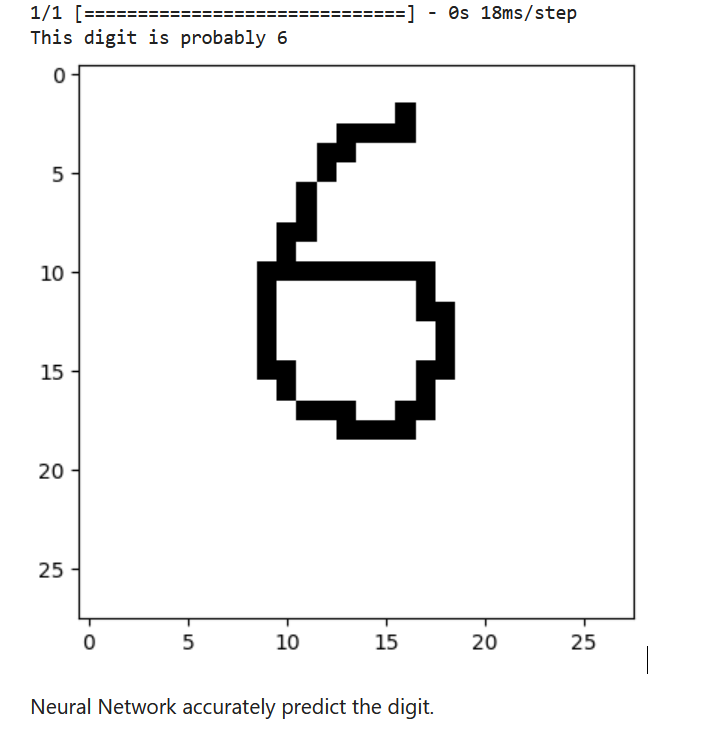


Figure 8: Prediction

# Conclusion

To sum up, this report explores the basic ideas and real-world uses of machine learning while providing an understanding of data partitioning, model assessment, and encoding methods. The research highlights the significance of appropriate data management in guaranteeing the robustness and generalization of models by distinguishing between training, validation, and test sets. In addition, talks about regression and classification issues clarify the wide range of machine learning models and how they are used in industries such as marketing, finance, and healthcare. The role of encoding techniques including ordinal, one-hot, and dummy variable encoding in converting categorical data into a format suitable with machine learning a necessary step in preprocessing is investigated and also presents Streamlit as an approachable platform for building interactive web apps which facilitates the easy sharing of data science initiatives and discoveries. The report provides a step-by-step explanation of the model building, evaluation, and deployment process using a real-world scenario involving the recognition of handwritten digits using neural networks and support vector machines. This provides readers with a comprehensive knowledge of machine learning ideas and techniques by combining theoretical insights with practical demonstrations. This enables readers to effectively utilize these tools for problem-solving and creativity in real-world scenarios.

# References

Ali, A., & Sheng-Chang, C. (2020). Characterization of well logs using K-mean cluster analysis. *Journal of Petroleum Exploration and Production Technology*, *10*(6), 2245–2256. https://doi.org/10.1007/S13202-020-00895-4/FIGURES/13

An, Q., Rahman, S., Zhou, J., & Kang, J. J. (2023). A Comprehensive Review on Machine Learning in Healthcare Industry: Classification, Restrictions, Opportunities and Challenges. *Sensors 2023, Vol. 23, Page 4178*, *23*(9), 4178. https://doi.org/10.3390/S23094178

Bowers, J. S., Malhotra, G., Dujmović, M., Llera Montero, M., Tsvetkov, C., Biscione, V., Puebla, G., Adolfi, F., Hummel, J. E., Heaton, R. F., Evans, B. D., Mitchell, J., & Blything, R. (2023). Deep problems with neural network models of human vision. *Behavioral and Brain Sciences*, *46*, e385. https://doi.org/10.1017/S0140525X22002813

Dahouda, M. K., & Joe, I. (2021). A Deep-Learned Embedding Technique for Categorical Features Encoding. *IEEE Access*, *9*, 114381–114391. https://doi.org/10.1109/ACCESS.2021.3104357

Kaddoura, S., Popescu, D. E., & Hemanth, J. D. (2022). A systematic review on machine learning models for online learning and examination systems. *PeerJ Computer Science*, *8*(Ml), 1–32. https://doi.org/10.7717/PEERJ-CS.986

Kwabena Patrick, M., Felix Adekoya, A., Abra Mighty, A., & Edward, B. Y. (2022). Capsule Networks – A survey. *Journal of King Saud University - Computer and Information Sciences*, *34*(1), 1295–1310. https://doi.org/10.1016/J.JKSUCI.2019.09.014

Lejeune, E. (2020). Mechanical MNIST: A benchmark dataset for mechanical metamodels. *Extreme Mechanics Letters*, *36*, 100659. https://doi.org/10.1016/J.EML.2020.100659

Liu, P. J., McFerran, B., & Haws, K. L. (2020). Mindful Matching: Ordinal Versus Nominal Attributes. *Journal of Marketing Research*, *57*(1), 134–155. https://doi.org/10.1177/0022243719853221/ASSET/IMAGES/LARGE/10.1177\_0022243719853221-FIG7.JPEG

Maulud, D. H., & Mohsin Abdulazeez, A. (2020). A Review on Linear Regression Comprehensive in Machine Learning. *Journal of Applied Science and Technology Trends*, *1*(2), 140–147. https://doi.org/10.38094/jastt1457

Nti, I. K., Nyarko-Boateng, O., Aning, J., & Justice, A. (2021). Performance of Machine Learning Algorithms with Different K Values in K-fold Cross-Validation. *Article in International Journal of Information Technology and Computer Science*, *6*, 61–71. https://doi.org/10.5815/ijitcs.2021.06.05

Palanivinayagam, A., El-Bayeh, C. Z., & Damaševičius, R. (2023). Twenty Years of Machine-Learning-Based Text Classification: A Systematic Review. *Algorithms 2023, Vol. 16, Page 236*, *16*(5), 236. https://doi.org/10.3390/A16050236

Raschka, S. (2018). *Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning*. https://arxiv.org/abs/1811.12808v3

Rong, S., & Bao-Wen, Z. (2018). The research of regression model in machine learning field. *MATEC Web of Conferences*, *176*, 01033. https://doi.org/10.1051/MATECCONF/201817601033

Saputra, D., Haryani, Junaidi, A., Baidawi, T., & Surniandari, A. (2023). Application of K-mean clustering algorithm in grouping data prospective new students. *AIP Conference Proceedings*, *2714*(1). https://doi.org/10.1063/5.0128402/2889724

Vabalas, A., Gowen, E., Poliakoff, E., & Casson, A. J. (2019). Machine learning algorithm validation with a limited sample size. *PLOS ONE*, *14*(11), e0224365. https://doi.org/10.1371/JOURNAL.PONE.0224365

Yin, Y., Shi, D., & Fairchild, A. J. (2023). The Effect of Model Size on the Root Mean Square Error of Approximation (RMSEA): The Nonnormal Case. *Structural Equation Modeling: A Multidisciplinary Journal*, *30*(3), 378–392. https://doi.org/10.1080/10705511.2022.2127729