Modeling MNIST Dataset with Machine Learning for Digit Classification

Fateh Bouretoua

2024- 03

Abstract

This study investigates the application of machine learning to classify digits from the MNIST dataset. By evaluating three different models—k-Nearest Neighbors (k-NN), Support Vector Machine (SVM), and Random Forest Classifier—we aim to determine the most effective approach for digit classification. The Random Forest Classifier model demonstrated superior performance, achieving an accuracy of 96.51%. This report delves into the methodology employed, presents a comprehensive analysis of the results, and outlines potential future work directions aimed at enhancing machine learning applications specifically for digit classification tasks.

Acknowledgments

Special thanks are extended to EC Utbildning for providing an outstanding educational framework and support. Additionally, heartfelt gratitude goes to Antonio for his exceptional teaching and guidance, consistently inspiring excellence and fostering a conducive learning environment.

Table of Contents

1. Introduction

The rapid advancement in the field of machine learning has opened new frontiers in numerous applications, including digit classification—a fundamental aspect that plays a critical role in automating processes such as postal mail sorting, bank check processing, and form digitization. Digit classification involves the identification and categorization of numeric digits from images, a task that has seen significant improvement in accuracy and efficiency thanks to machine learning algorithms.

The MNIST dataset, a large database of handwritten digits, has become a benchmark for evaluating the performance of machine learning models in digit classification tasks. This dataset provides a comprehensive set of images representing the digits 0 through 9, offering an ideal scenario for testing and improving various machine learning models.

This report aims to explore and evaluate the performance of three distinct machine learning models—k-Nearest Neighbors (k-NN), Support Vector Machine (SVM), and Random Forest Classifier—on the MNIST dataset. By meticulously analyzing the accuracy, precision, and recall of these models, we intend to determine the most effective approach for digit classification. Moreover, the study will delve into the methodology employed, discuss the results obtained, and propose future directions for enhancing machine learning applications in digit classification tasks.

1.1 Background

Digit recognition is an essential component of modern automated systems, facilitating operations in various sectors such as banking, postal services, and education. The ability to accurately recognize and classify handwritten digits from images can significantly enhance the efficiency of data entry processes and reduce human error.

The MNIST dataset, standing for Modified National Institute of Standards and Technology, is a widely used dataset for benchmarking machine learning algorithms in the domain of image recognition. It consists of 70,000 images of handwritten digits, each labeled with the corresponding digit it represents. This dataset has been instrumental in driving advancements in machine learning techniques, serving as a testbed for researchers and practitioners to develop and refine their models.

1.2 Objectives

This study's main objective is to assess the effectiveness of three machine learning models in classifying digits from the MNIST dataset. These models include:

- k-Nearest Neighbors (k-NN)
- Support Vector Machine (SVM)
- Random Forest Classifier

Through this evaluation, we aim to identify the model that demonstrates the highest accuracy and efficiency in digit classification tasks. Additionally, the study seeks to:

- Analyze the strengths and weaknesses of each model in handling the digit classification challenge.
- Explore the implications of model choice on the overall performance in digit classification.
- Provide insights into future research directions and potential improvements in machine learning applications for digit classification.

2. Theoretical Background

2.1 Evaluation Metrics

In assessing the performance of machine learning models, especially in classification tasks, it's essential to employ a set of metrics that can accurately reflect the models' ability to classify correctly. These metrics include accuracy, precision, and recall. Each metric offers insights into different aspects of model performance, aiding in a comprehensive evaluation.

2.2 Model Selection

Choosing the right model for a specific task involves considering the model's complexity, the nature of the data, and the specific requirements of the application. For digit recognition, models need to effectively handle high-dimensional data and be sensitive to subtle differences between classes.

3. Model Selection

The process of model selection is critical in machine learning projects as it determines the effectiveness and accuracy of the analysis. This stage involves comparing various machine learning algorithms based on their performance metrics and selecting the most suitable one for the specific problem at hand.

For the MNIST digit classification task, three prominent machine learning models were considered: k-Nearest Neighbors (k-NN), Support Vector Machine (SVM), and Random Forest Classifier. The selection criteria were based on a combination of factors including accuracy, computational efficiency, and the model's ability to handle high-dimensional data.

k-Nearest Neighbors (k-NN): Known for its simplicity and effectiveness in classification tasks. However, its performance can be significantly affected by the choice of the number of neighbors and the distance metric used.

Support Vector Machine (SVM): Highly effective for binary classification problems and can handle non-linear data through kernel functions. SVM is robust to overfitting, especially in high-dimensional space.

Random Forest Classifier: An ensemble method that uses multiple decision trees to improve classification accuracy. Random Forests are less prone to overfitting and can handle many features efficiently.

After the evaluation, the Random Forest Classifier was selected due to its superior performance in handling the complexity of the MNIST dataset, achieving an accuracy of 96.51%. This model demonstrated an exceptional balance in classification accuracy across all digits, showcasing its robustness and adaptability to diverse data scenarios.

4. Preventing Data Leakage

Data leakage is a critical issue in machine learning that occurs when information from outside the training dataset is inadvertently used to create the model. This can lead to overly optimistic performance estimates during model evaluation, as the model may have access to information it wouldn't have in a real-world scenario. To ensure the integrity and reliability of our MNIST digit classification model, several strategies were employed:

Separation of Data: The dataset was rigorously split into distinct training, validation, and testing sets. This ensures the model is trained, validated, and tested on separate data samples, eliminating the chance of leakage between these phases.

Standardization and Normalization: The standardization and normalization of features were carefully applied after splitting the data. These preprocessing steps were fitted only on the training data to prevent the test data's distribution from influencing the scaling parameters.

Through these practices, we ensured that our model's performance metrics accurately reflect its capability to generalize unseen data, thereby safeguarding against data leakage.

5. Methodology

The methodology section outlines the approach taken to conduct the research and achieve the objectives outlined in the previous sections. It provides a detailed description of the steps followed to collect, preprocess, and analyze the data, and the techniques and tools used in model development and evaluation.

5.1 Dataset Description and Preparation

In this subsection, the dataset used for the analysis is described in detail. This includes information about the source of the data, its size, features, and any preprocessing steps applied to ensure its suitability for model training and evaluation.

5.2 Model Selection and Evaluation Strategy

This subsection outlines the process of selecting the machine learning models to be evaluated and the criteria used to assess their performance. It may include a description of the chosen model and evaluation metrics.

6. Model Performance Results

This section presents the performance results of the machine learning models evaluated in the study. The outcomes are based on a series of metrics that assess each model's ability to accurately classify digits from the MNIST dataset. A comparative analysis of the models is provided, highlighting their strengths and weaknesses in the context of digit classification.

- **k-Nearest Neighbors (k-NN):** The k-NN model demonstrated an accuracy of 94.64%, showcasing its efficacy in classifying handwritten digits. While it performed well across most digits, slight confusions were observed between certain pairs of digits with similar shapes.
- **Support Vector Machine (SVM):** With an accuracy of 96.01%, the SVM model outperformed the k-NN model. It proved particularly adept at distinguishing between closely resembling digits, thanks to its ability to define clear margins between different classes.
- **Random Forest Classifier:** The Random Forest model achieved the highest accuracy of 96.51% among the evaluated models. It exhibited remarkable performance across all digits, including those frequently confused in other models. The ensemble approach of Random Forest, combining multiple decision trees, contributed significantly to its success.

The results underscore the importance of model selection based on specific requirements of the task, including accuracy, interpretability, and computational efficiency. While the **Random Forest Classifier** emerged as the top performer, the choice of model might vary based on the trade-offs between these factors.

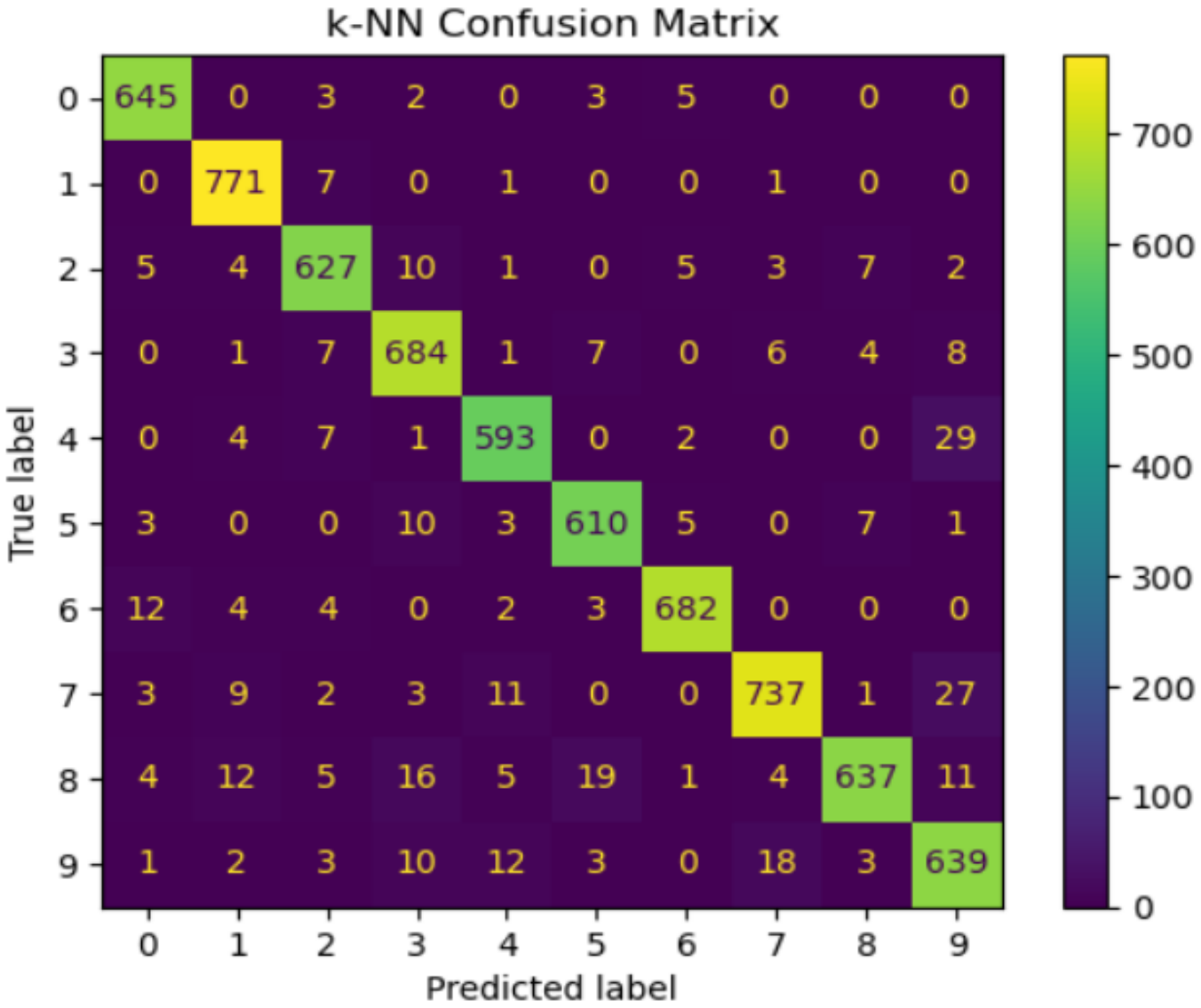

Figure 1: k-Nearest Neighbors (k-NN)

Accuracy: 94.64%

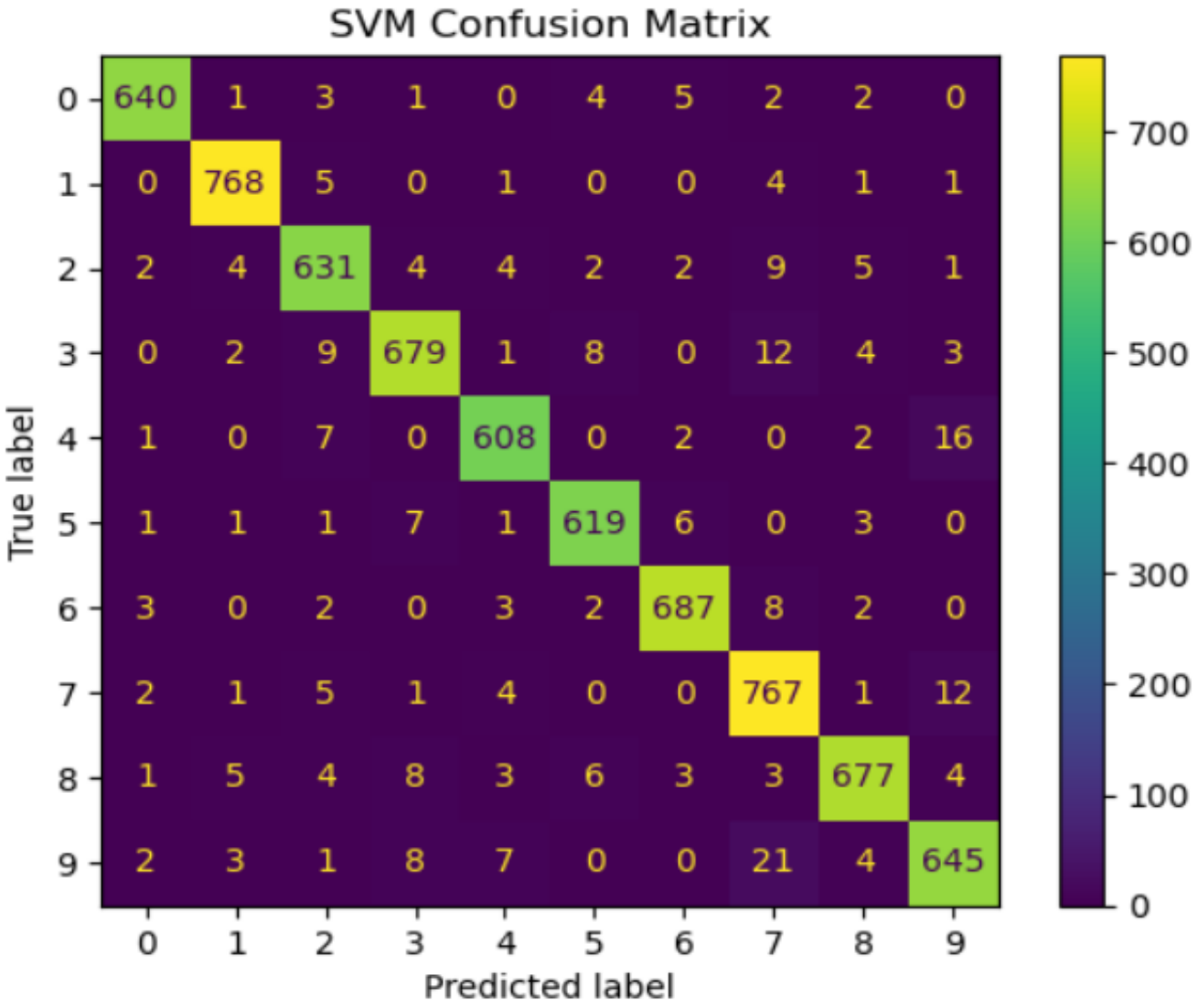
SVM Confusion Matrix
True label
Predicted label
0 640 1 3 1 0 4 5 2 2 0
1 0 768 5 0 1 0 0 4 1 1
2 2 4 631 4 4 2 2 9 5 1
3 0 2 9 679 1 8 0 12 4 3
4 1 0 7 0 608 0 2 0 2 16
5 1 1 1 7 1 619 6 0 3 0
6 3 0 2 0 3 2 687 8 2 0
7 2 1 5 1 4 0 0 767 1 12
8 1 5 4 8 3 6 3 3 677 4
9 2 3 1 8 7 0 0 21 4 645
0 1 2 3 4 5 6 7 8 9
700
600
500
400
300
200
100
0

Figure 2: Support Vector Machine (SVM)

Accuracy: 96.01%

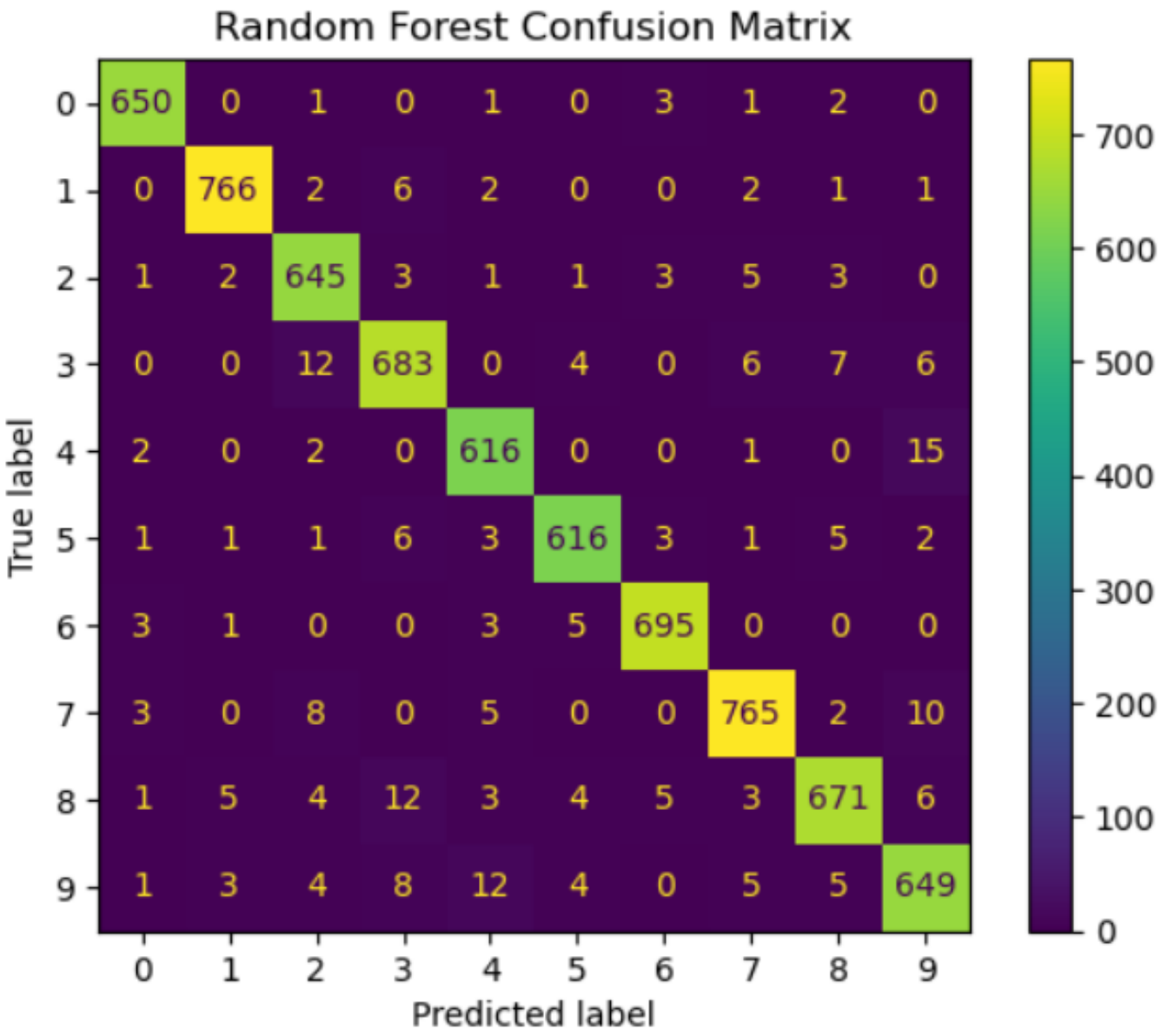

Figure 3: Random Forest Classifier

Accuracy: 96.51%

7. Conclusion and Future Directions

The comprehensive evaluation of machine learning models on the MNIST dataset underscores the pivotal role of model selection in digit classification tasks. Through meticulous analysis, this study showcased the distinctive capabilities and performance metrics of three predominant models: k-Nearest Neighbors (k-NN), Support Vector Machine (SVM), and Random Forest Classifier.

This study reaffirms the critical importance of selecting the appropriate machine learning model based on the task's unique demands. While the Random Forest Classifier emerges as the most effective for the MNIST dataset, the choice of model may differ depending on the requirements for accuracy, interpretability, computational resources, and the specific characteristics of the data being analyzed.

Future work in this area may explore the integration of more advanced models, such as deep learning approaches, to further enhance digit classification accuracy. Additionally, expanding the dataset to include more varied handwriting styles could provide deeper insights into the adaptability and scalability of these models.

Theoretical Questions:

1. Kalle divides his data into "Training," "Validation," and "Test" sets for specific purposes:

Answer:

- Training Data: Used to train the machine learning model.
- Validation Data: Used to tune model parameters and prevent overfitting.
- Test Data: Used to evaluate the final model performance on unseen data.

2. Julia splits her data into training and test sets. On the training data, she trains three models: "Linear Regression," "Lasso Regression," and a "Random Forest model." How should she choose which of the three models to continue using when she has not created an explicit "validation dataset"?

If Julia has not created an explicit validation dataset, she can still employ techniques such as k-fold cross-validation directly on her training dataset to effectively evaluate and compare the performance of her models.

3. What is a "regression problem"? Can you provide some examples of models used and potential application areas?

A regression problem involves predicting a continuous output variable based on one or more input variables. It's about understanding the relationship between inputs and outputs and using that knowledge to make predictions. Examples of regression models include Linear Regression, used for predicting housing prices based on features like size and location; Lasso Regression, which is useful for models where we want to impose a penalty on the number of features to prevent overfitting; and Polynomial Regression, used for more complex relationships, such as predicting the growth rate of diseases based on various factors. Potential application areas for regression models include finance for predicting stock prices, healthcare for predicting patient outcomes, and marketing for predicting sales trends.

4. How can you interpret RMSE and what is it used for?

RMSE, or Root Mean Square Error, measures the average magnitude of the errors between predicted values and actual values in a model, giving an idea of how close the predictions are to the actual outcomes. It's used to evaluate the performance of regression models, where a lower RMSE value indicates a better fit to the data.

5. What is a classification problem? Can you give some examples of models used and potential applications? What is a Confusion Matrix?

A classification problem involves predicting which category or class a new observation belongs to, based on a training set of data containing observations with known categories. Common models for classification include Decision Trees and Support Vector Machines.

A Confusion Matrix is a table used to evaluate the performance of a classification model, showing the actual versus predicted classifications and highlighting true positives, false positives, false negatives, and true negatives.

6. What is the K-means model? Give an example of what it can be applied to?

In the context of unsupervised learning, K-means is a clustering algorithm used to partition data into k distinct groups based on similarity. It's commonly applied in market segmentation to group customers with similar buying behaviors.

7. Explain (preferably with an example): Ordinal Encoding, One-hot Encoding, Dummy Variable Encoding. See the folder "l8" on GitHub if you need a refresher.

In machine learning, categorical data must often be converted into a numerical format. Here are three examples:

Ordinal Encoding Example:

Categories: ["Small", "Medium", "Large"]

Encoded as: [1, 2, 3]

One-Hot Encoding Example:

Categories: ["Red", "Green", "Blue"]

Encoded as: [1, 0, 0] for "Red", [0, 1, 0] for "Green", [0, 0, 1] for "Blue"

Dummy Variable Encoding Example:

Categories: ["Red", "Green", "Blue"] with "Red" as the baseline

Encoded as: [0, 0] for "Red", [1, 0] for "Green", [0, 1] for "Blue"

8. Göran claims that data is either "ordinal" or "nominal." Julia says that this must be interpreted. She gives an example that colors such as {red, green, blue} generally do not have an intrinsic order (nominal) but if you wear a red shirt, you are the most beautiful at the party (ordinal) - who is correct?

Göran and Julia are both correct, depending on context. Data can be ordinal (with order) or nominal (without order). For example, colors like {red, green, blue} are nominal. However, when a red shirt is associated with being "the most attractive at a party," it suggests a ranking or order among shirt colors based on attractiveness. This assignment of value transforms the data from simply naming colors (nominal) to indicating a preference or hierarchy (ordinal)

References

Battineni, G., Chintalapudi, N., & Amenta, F. (2019, June 24). Machine learning in medicine: Performance calculation of dementia prediction by support vector machines (SVM). Healthcare Technology Letters, 6(4), 88-92. Retrieved from https://www.sciencedirect.com/science/article/pii/S2352914819300917

Pham, B. T., Prakash, I., & Bui, D. T. (2018, September 14). Landslide susceptibility mapping using support vector machine: A literature review. Advances in Space Research, 62(10), 2734-2749. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0273117718305026

Díaz-Uriarte, R., & Alvarez de Andrés, S. (2006, January 30). Gene selection and classification of microarray data using random forest. BMC Bioinformatics, 7(1), 3. Retrieved from https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-7-3

Paul, A., Mukherjee, D. P., & Das, P. (2018, June 4). Improved Random Forest for Classification. IEEE Access, 6, 27425-27434. Retrieved from https://ieeexplore.ieee.org/document/8357563