👑 325483444_322695883 👑

Project Proposal - Machine Learning- Gidi Rabi & Roi Bruchim

The dataset we are interested in working on is the MNIST dataset. This dataset is particularly interesting to us because of its simplicity and its significance as a benchmark in the field of machine learning. We believe it provides an excellent opportunity to explore and compare different machine learning models while allowing us to understand the strengths and limitations of each.

Through this project, we aim to gain deeper insights into how various machine learning algorithms perform on a well-known dataset, and we hope to learn valuable skills that will help us in more complex projects in the future..

The Models and General Explanations for Their Selection (5)

1. **SVM (Support Vector Machines)**

- SVMs are known for their solid theoretical foundation and high accuracy on datasets like MNIST. By using kernels like RBF, they can handle non-linear data effectively and create optimal decision boundaries. They're great for projects where accuracy and theoretical understanding are key.

2. **CNN (Convolutional Neural Networks)**

- CNNs are the go-to model for image classification tasks. Their ability to capture spatial features in images makes them ideal for MNIST. While they're more complex, their performance justifies the added effort, making them a top choice for this dataset.

3. **Random Forest**

- Random Forests are easy to implement and very interpretable. They're good for handling non-linear relationships and offer insights through feature importance. While they may not match CNNs in accuracy, they're reliable and useful for explaining results.

4. **Logistic Regression**

- Logistic Regression is simple and provides a good baseline for classification. It's not the best for a complex dataset like MNIST, but it's helpful to understand how linear models perform as a starting point.

5. **K-Nearest Neighbors (KNN)**

- KNN is easy to understand and works well for MNIST because it classifies based on similarity in feature space. It's especially useful for exploring how data distributions influence predictions, even though it can be slow for larger datasets.

Questions we would like to answer in this project:

1. How does computation time (training/testing) compare between the models?
2. Does the added complexity of CNN justify its improved accuracy?
3. Which digits are frequently misclassified?
4. Why are certain digits harder to classify? Is it due to their similarity or variability in writing styles?
5. Can models like Random Forest or SVM identify which features (pixel regions) are most important for classification?
6. Which model brings the best overall results for this dataset?
7. How do models perform when visualized through graphs and statistical analysis?
8. How does artificial class imbalance affect model performance?
9. Which models are more interpretable (e.g., decision paths in Random Forests vs. CNN feature maps)?
10. What are the trade-offs between model simplicity, accuracy, and computational cost?