INNOMATİCS
RESEARCH LABS

Machine Learning - Project Report Document

Student Name	Sada Vijay, Dawa Phuti Lepcha
Batch	AI Elite 18
Project Name	Recognising Handwritten Alphabets
Project Domain	Image Classification
Type of Machine Learning	Supervised ML
Type of Problem	Classification
Project Methodology	CRISP-DM
Stages Involved	• Data Collection and Understanding • Data Preparation • Model Building • Model Training

	• Model Evaluation

Business Understanding:

Recognizing handwritten alphabets is a crucial task in the field of computer vision and has wide-ranging applications across various industries. The MNIST dataset, which comprises images of handwritten digits, is a standard benchmark for evaluating classification algorithms. By extending this to handwritten alphabets, we can significantly impact several business domains:

- **Education**: Automate grading and provide personalized handwriting feedback.
- **Finance**: Speed up cheque processing and form digitization.
- **Healthcare**: Digitize handwritten medical notes and prescriptions.
- **Customer Service**: Automate mail sorting and analyze handwritten feedback.
- **Archives**: Digitize historical documents for better accessibility.

By successfully recognizing handwritten alphabets, this project can pave the way for numerous practical applications, improving efficiency and accuracy in various sectors. The insights gained from this project will enable businesses to leverage advanced machine learning techniques to enhance their operations and deliver better services to their customers.

Problem Statement: The objective is to develop a machine learning model using the MNIST dataset by creating a structured pandas DataFrame to accurately recognize and classify handwritten alphabets.

Stage 1: Data Collection and Understanding

a) **Data Collection:** The data was provided to us by the client.
b) **Data Understanding:**

The dataset 'mnist_data.zip' contains handwritten alphabet images, comprising 26 classes representing each letter of the English alphabet. Each image is grayscale and has dimensions of 28x28 pixels.

The dataset has a shape of (372451, 785), indicating that it consists of 372,451 samples (images) with 785 columns. In typical MNIST datasets, each sample is represented by a flattened array of pixel values, where 784 columns represent the pixel values of a 28x28 image, and one additional column represents the label of the alphabets.

Here are the features and their descriptions:

1. column 0 to 783: contains the flattened pixel values of an image of size 28*28
2. label: Contains the information of the alphabet the image belongs to.

S No	Feature Name	Data Type
1	Col 0 to 783	Int8
2	label	Object

Stage 2: Data Preparation

a) Exploratory Data Analysis:

S No	Type	Feature Names	Observation
1	Missing Values	NA	NA
2	Duplicates	In rows	There were a lot of duplicates – 171355 rows.
3	Outliers	NA	NA

b) Data Cleaning/wrangling:

S no	Type of Cleaning	Technique	Feature Name	Reason
1	Missing value	NA	NA	NA
2	Duplicates	Drop	In rows	They are unnecessary.
3	Reshaping	Resized columns to size 0 to 399	Col 0 to 783	To reduce the size of the dataframe for ease in computation.
4	Scaling	Standard Scaler	Col 0 to 399	Used standardization to scale down all the columns into a similar scale ranging between 0 to 1 based on standard deviation.

Stage 3: Model Building:

S No	Type of Problem	Algorithm Name
1	Classification	KNNeighbors Classifier
2	Classification	Logistic Regression
3	Classification	SVC
4	Classification	Random Forest Classifier
5	Classification	Decision Tree Classifier
6	Classification	Gradient Boost Classifier
7	Classification	Naïve Bayes Classifier

1. **Logistic Regression:** Logistic regression is a statistical method used to predict the probability of an event happening, such as whether an email is spam or not. Unlike linear regression, it works well for situations where the outcome is binary (yes/no) instead of continuous.

2. **SVC:** A support vector classifier (SVM) excels at finding the best separation line between categories in your data. It prioritizes a wide margin between the classes, making it effective even for complex datasets.

3. **KNeigbors Classifier:** The K-Nearest Neighbors (KNN) classifier predicts a data point's class by analyzing the labels of its closest neighbors in the training data, making it simple to understand and effective for various classification tasks.

4. **Decision Tree Classifier**: A decision tree classifier is a machine learning method that uses a tree-like structure to classify data. It asks a series of questions about the data's features, branching out based on the answers, until it reaches a final leaf node that predicts the class.

5. **Random Forest Classifier:** Random Forest Classifier is a machine learning algorithm that combines multiple decision trees for stronger predictions. By training a "forest" of trees on random subsets of data, it reduces the risk of overfitting and improves overall accuracy.

6. **Gradient Boosting Classifier**: Gradient Boosting Classifier is a machine learning algorithm that builds a series of decision trees sequentially for stronger predictions. By

correcting errors of previous trees in each step, it improves overall accuracy and handles complex datasets effectively.

7. **Naive Bayes Classifier**: The Naïve Bayes classifier is a probabilistic machine learning algorithm based on Bayes' Theorem with an assumption of independence between features. Despite its simplicity and "naïve" assumption, it's surprisingly effective in many-real world applications, especially in text classification and sentiment analysis.

Stage 4: Model Training:

S No	Algorithm Name	Metric used for Evaluation
1	KNN Classifier	Accuracy
2	Logistic Regression	Accuracy
3	Support Vector Classifier	Accuracy
4	Random Forest Classifier	Accuracy
5	Decision Tree Classifier	Accuracy
6	Gradient Boost Classifier	Accuracy
7	Naïve Bayes Classifier	Accuracy

Stage 5: Model Evaluation:

S No	Algorithm Name	Metric Score
1	KNN Classifier	0.913600
2	Logistic Regression	0.878543
3	Support Vector Classifier	0.962805
4	Random Forest Classifier	0.951069
5	Decision Tree Classifier	0.846370
6	Gradient Boost Classifier	0.913178
7	Naïve Bayes Classifier	0.521457

Challenges Faced:

While running the dataset we've experienced ram crashes since the dataset was of high dimensionality, which also lead to higher computation time as the dataset was large size. So, to overcome this we've resized the image data from size of 28x28 to 20x20.

Conclusion:

From the above Accuracy results we can observe that **Support Vector Classifier** and **Random Forest Classifier** gives the highest accuracy when compared to all the other models i.e. they have more than **0.95 accuracy score.**

While the **Support Vector Classifier** took around 48 mins to learn and predict, the **random forest classifier** took only 3 mins to learn and predict.

The model which took the least time was gaussian naïve bayes and the model which took the most time was gradient boosting classifier.