INNOMATICS®
RESEARCH LABS

INNOVATION. AUTOMATION. ANALYTICS

PROJECT ON

RECOGNISING HANDWRITTEN ALPHABETS

About us

- Vijay sada
- B-tech in Electronics and Communications Engineering.
- Fresher
- LinkedIn - https://www.linkedin.com/in/vijay-sada/
- GitHub - https://github.com/vijaysada

- Dawa Phuti Lepcha
- MSc. Mathematics
- Fresher
- LinkedIn - https://www.linkedin.com/in/dawa-phuti-lepcha-646768240/

INNOMATICS RESEARCH LABS

Introduction

- 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 analyse 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  - **Archives**: Digitize historical documents for better accessibility.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.

INNOMATICS RESEARCH LABS

Data Overview

The dataset was provided to us by the client.

The dataset includes the following features:

- **Col_0 to Col_783:** contains the flattened pixel values of an images of size 28*28
- **label:** .contains the information of the alphabets the image belong to.

The target column is '**label**'

S No	Feature Name	Data Type
1	Col_0 to Col_783	Int64
2	label	Object

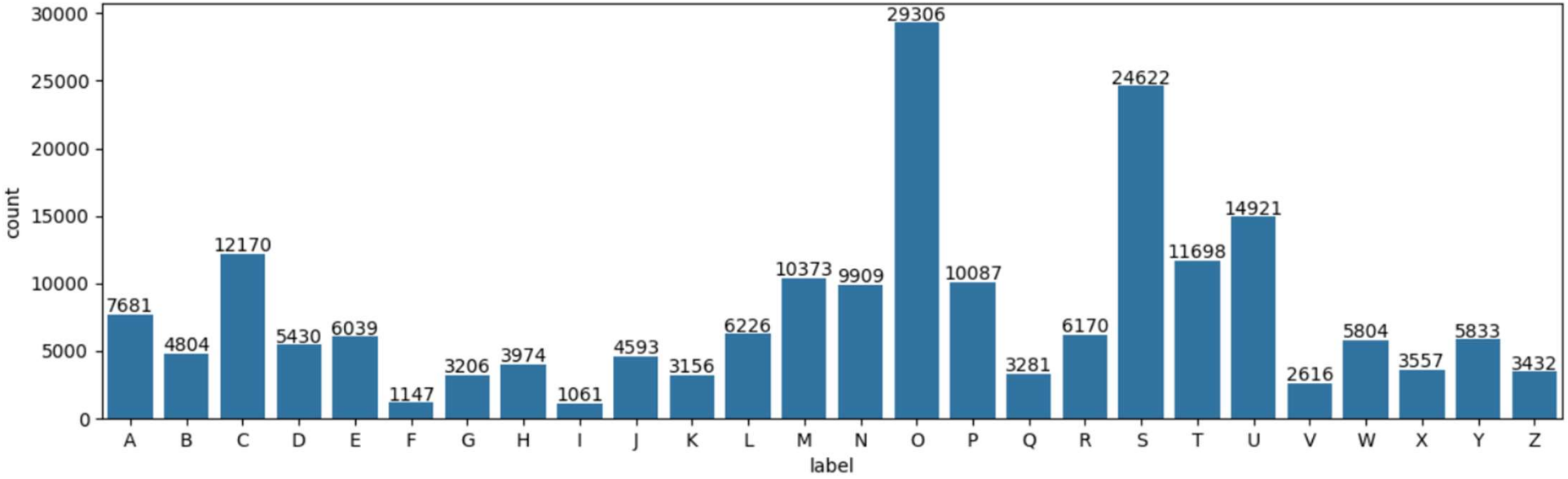
INNOMATICS RESEARCH LABS

Pre-Processing

For the project, the following preprocessing steps were performed:

- **Handling the duplicates:** Dropped the duplicates because they are unnecessary. There were 171355 duplicate rows.
- **Reshaping:** Reduced the size of the dataframe for ease of computation. Resized from 28*28 to 20*20.
- **Scaling:** Used standardization to scale down all the input columns into a similar scale ranging between 0 to 1 based on standard deviation.

Exploratory Data Analysis (EDA)



- **The above plot shows the count of label in the dataset.**
- **We can observe that there are more number of images which has the alphabets 0 and S. And F and I are in least number.**

INNOMATICS RESEARCH LABS

Model Building

For the classification task of score prediction, the below machine learning models were used:

- **K-Nearest Neighbors (KNN) Classifier:** A non-parametric method that classifies data points based on the majority class among their k-nearest neighbors, which can be effective for capturing local patterns in the data.
- **Logistic Regression:** A classic binary classification algorithm that models the probability of a customer churning based on input features. It's interpretable and efficient for linearly separable data.
- **Decision Trees:** These models partition the feature space into regions and make predictions based on majority voting within each region. They're intuitive and can capture non-linear relationships between features and the target variable.
- **Random Forest:** An ensemble method that builds multiple decision trees and combines their predictions to improve accuracy and robustness. It's effective for handling complex datasets and mitigating overfitting.
- **Support Vector Machines (SVM):** SVMs aim to find the hyperplane that best separates the data points into different classes. They're effective in high-dimensional spaces and suitable for datasets with clear margins of separat.
- **Gradient Boosting Classifier**: It 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.
- **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.

Model Training

The model training process involved:

- Splitting the dataset into training and testing sets (80:20 Ratio) to train the models on the data and evaluate their performance on unseen data.
- Training each model using the training set, where the model learns patterns and relationships between features and the target variable (label).

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

Model Evaluation

Evaluation metrics employed to assess model effectiveness was accuracy:

- **Accuracy:** The accuracy metric measures the proportion of correctly classified instances out of total instances, providing a straightforward evaluation of overall model performance. It is commonly used in classification tasks to assess the model's ability to correctly predict the class labels of the dataset.

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.

INNOMATICS RESEARCH LABS

THANK YOU

INNOMATICS
RESEARCH LABS