Machine Learning Model Report

Team id:GSTN_1011

Machine Learning Model Training Report

Introduction

This project aims to build a machine learning model to solve a classification problem. The objective is to develop a robust model that can accurately predict the target class. Various machine learning algorithms, such as Random Forest, Logistic Regression, and others, were evaluated. After multiple experiments and model performance evaluations, the CatBoost classifier was chosen as the final model due to its superior performance.

Data Overview

The dataset used for this project consists of the following features:

- Input Features: A collection of numerical and categorical features.
- Target Variable: A numerical value (0,1) representing the class labels for classification.
-

The data was split into training and testing sets for model training and evaluation. Cross-validation was also used to assess model stability and reliability.

Data Exploration and Preprocessing

The next critical step is exploring and preprocessing the data. Here are the key activities involved:

- **Exploratory Data Analysis (EDA):** This involves understanding the dataset, identifying trends, and detecting patterns. EDA includes statistical analysis, data visualization, and identifying potential correlations among features.
- **Handling Missing Data:** Missing values were imputed to prevent loss of information. The **IterativeImputer** method was used for this purpose. This approach models each feature with missing values as a function of other features and uses those models to predict and fill in missing values iteratively. This method is particularly effective as it considers relationships between variables, often leading to more accurate imputations. For example, numerical data might be imputed with the mean or median, while

categorical data might use mode or forward-filling.

- Feature Scaling and Normalization: This step ensures that features with different units or magnitudes do not disproportionately influence the model. Techniques like Min-Max scaling or Standardization are often used. Scaling helps improve the convergence speed of optimization algorithms and ensures that all features contribute equally to the model.

Feature Engineering

This step involves creating or transforming features to improve model performance. Common techniques include:

- Deriving New Features: Creating new features from existing data to better represent relationships. This can involve polynomial features, interaction terms, or aggregating features.
- Removing Irrelevant or Redundant Features: Eliminating features that add noise or don't contribute much to the prediction helps simplify the model and improve performance. The following features were eliminated:
    - Column9, Column5, Column10, Column11, Column13, Column14, Column15, Column16, Column20, Column21.
- Scaling and Normalization: Features were scaled to ensure uniformity across different magnitudes, which enhances the model's performance and stability.

Model Building and Experimentation

Multiple models were evaluated for this classification problem:

1. Logistic Regression: Simple and interpretable but struggled with complex patterns.
2. Random Forest: Provided better results but was computationally expensive and prone tooverfitting.
3. Gradient Boosting: An improvement over Random Forest, but training time was longer.
4. CatBoost Classifier: The most effective model due to its efficient handling of numerical features.

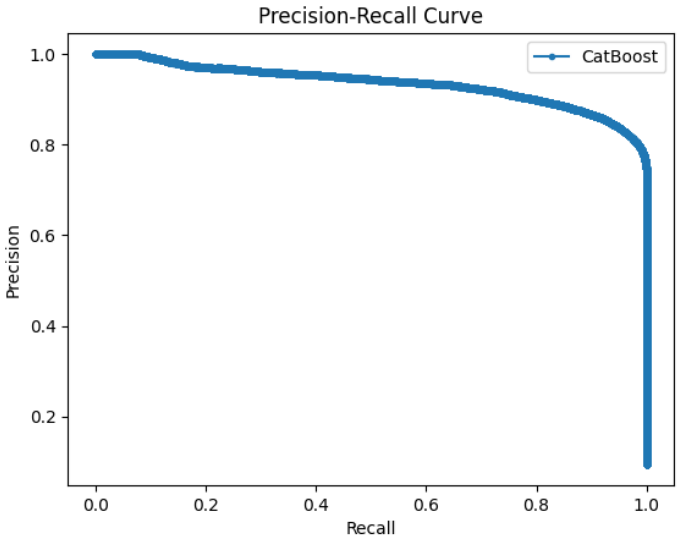
After comparing the models, **CatBoost** was selected as the final model.

Machine Learning Model Report

Model Evaluation

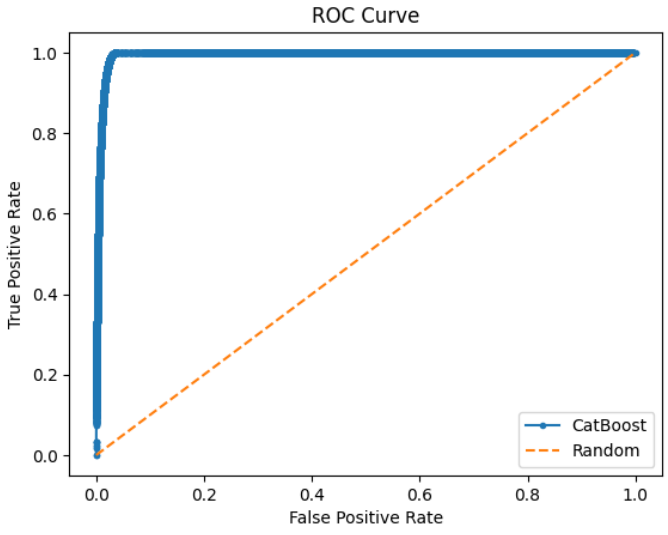
The CatBoost model was evaluated using the following metrics:

- **Accuracy**: 0.97.8, indicating that 98% of predictions were correct.
- **Precision**: 0.9794, meaning that 97% of positive predictions were correct.
- **Recall**: 0.9780, meaning the model correctly identified 98% of all actual positives.



-

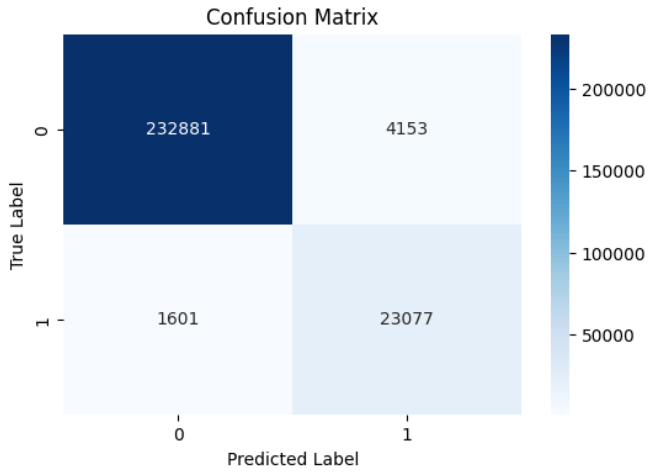
- **F1-Score**: 0.978, representing a balanced measure of precision and recall.
- **ROC-AUC Score**: 0.9946, demonstrating excellent ability to distinguish between classes.



-

Confusion matrix:

ɔdel Report



Conclusion and Insights

After experimenting with several models, the CatBoost Classifier was chosen due to its ability to handle complex patterns and categorical variables efficiently. The model performed exceptionally well across all evaluation metrics, making it a reliable choice for deployment.

Key Findings

1. CatBoost Outperformed Other Models

- The **CatBoost model** outperformed alternative algorithms like Logistic Regression, Random Forest, and Gradient Boosting, particularly in handling numerical features efficiently with minimal preprocessing.
- With a near-perfect **ROC-AUC score of 0.9946**, the model demonstrated superior ability to differentiate between classes, making it the top performer compared to other models tried during experimentation.

2. Exceptional Performance Across Evaluation Metrics

- **Accuracy: 97.8%** – The model achieved an impressive accuracy, correctly classifying 98% of the samples. This suggests strong generalization to unseen data.
- **Precision: 97.94%** – The model minimized false positives, with 97.94% of positive predictions being correct.
- **Recall: 97.80%** – The model successfully identified 98% of actual positive cases, demonstrating high sensitivity.
- **F1-Score: 97.8%** – The F1-score, balancing precision and recall, was extremely high,

- confirming the model's robustness in managing both false positives and false negatives.

3. Excellent Discrimination Between Classes

- The **ROC-AUC score of 0.9946** indicates the model's excellent ability to discriminate between positive and negative classes, meaning it can correctly rank predictions with near-perfect accuracy.

4. Balanced Trade-Off Between Precision and Recall

- With a nearly equal **Precision (97.94%)** and **Recall (97.80%)**, the model shows an optimal trade-off between identifying true positives and avoiding false positives. This balance makes the model highly reliable in situations where both sensitivity and precision are important.

5. High Interpretability and Low Preprocessing Overhead

- The **CatBoost model** required minimal preprocessing, making it highly efficient in both training time and deployment. Its native handling of numerical features reduced the need for complex feature engineering steps.

Recommendations

- CatBoost is recommended for its superior performance.
- Regular monitoring and retraining of the model with new data will help maintain its accuracy.
- Future work can explore hyperparameter tuning for further performance gains.