YESHWANTRAO CHAVAN COLLEGE OF ENGINEERING

Project Quality Assurance Initiative-2(PQAI-2) Seminar

ON

MACHINE LEARNING PIPELINE DEVELOPMENT

PRESENTED BY -
VALHARI MESHRAM - 19
AMAN RAUT - 29
ANIKET KALOO - 30
ATHARVA NERKAR - 35
VIRANCHI DAKHARE - 67

SECTION: AIDS - A
YEAR: 2024-25

COLLEGE GUIDE
Dr. Prarthana Deshkar

INDUSTRIAL GUIDE:
Mr. Kaustubh Laghate
Incredo Technologies Pvt. Ltd

CONTENTS

INTRODUCTION

Abstract

Developing a machine learning pipeline to automate the lifecycle of machine learning models, encompassing tasks from data pre-processing and feature extraction to model training and deployment.

Aim

To create an open-source ML pipeline tool that automates and optimizes the machine learning workflow, maintaining accuracy, rapid model iterations, and consistency.

Objectives

- Implement customizable modules for all steps in the machine-learning process
- Design a flexible evaluation framework with algorithm-specific metrics
- Integrate user input mechanisms for pipeline customization and adaptability
- Implement a user-friendly interface to support users of varying expertise levels
- Build a dashboard to upload and test the trained model and the visualize the predictions for better understanding.

LITERATURE SURVEY

Reference No.	Title and Publication Details	Methodology Used	Key Understanding	Limitations
[1]	**[1] STREAMLINE: A Simple, Transparent, End-To-End Automated Machine Learning Pipeline Facilitating Data Analysis and Algorithm Comparison**	[1] STREAMLINE provides a comprehensive AutoML pipeline integrating exploratory analysis, data cleaning, ML modeling with hyperparameter optimization, evaluation, and automatic result export.	An AutoML pipeline focusing on binary classification. Transparent in ML analysis, including exploratory analysis, hyperparameter optimization, and result export. It requires domain expertise for interpretation and lacks versatility beyond binary classification.	STREAMLINE's binary classification focus restricts its versatility. Despite its comprehensive pipeline, users may need domain expertise for result interpretation and model decisions.
[2]	**[2] GAMA: A General Automated Machine Learning Assistant**	[2]GAMA utilizes AutoML for optimized ML pipelines, including data preprocessing, fine-tuned hyperparameters, various search procedures, ensemble methods, and detailed logs.	An AutoML system enables users to track/control ML pipeline optimization. It supports various AutoML techniques. Designed for both end-users and researchers, GAMA specializes in tabular data classification and regression.	GAMA's modular design may need advanced skills for customization and adding components, potentially leading to complexity and compatibility challenges, limiting its applicability for specific ML tasks.

PATENT SEARCH

Sr. No	Title	Publication Details(Patent no, author etc)	Methodology used	Summary of invention	Limitations
[3]	Annotation pipeline for machine learning algorithm training and optimization	US11475358B2 Inventor - Marc T. Edgar Travis R. Frosch Gopal B. Avinash Garry M. Whitley	The method collects data, prioritizes annotations, and selects techniques, improving annotation quality and streamlining the supervised machine learning process for efficiency and accuracy.	The advanced annotation pipeline improves efficiency, reduces manual effort, enhances data quality for ML models, leading to better predictions, decision-making, and technological advancements.	The advanced annotation pipeline may face challenges with complex data, requiring domain expertise. Predefined criteria might miss nuanced data traits, and priority-based technique selection could impact model performance.

WORK DONE

1)Data Ingestion

Data Ingestion
Exploratory Data Analysis
Data Cleaning
Feature Importance
Feature Scaling
Model Selection
Model Tuning
Ensembling
Report

Deploy

Upload Data

Upload your CSV file

Drag and drop file here
Limit 200MB per file • CSV

Browse files

diabetes.csv 23.3KB

Data preview:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

2. Exploratory Data Analysis

- Data Ingestion
- **Exploratory Data Analysis**
- Data Cleaning
- Feature Importance
- Feature Scaling
- Model Selection
- Model Tuning
- Ensembling
- Report

Exploratory Data Analysis

☑ Show Descriptive Statistics

Data Description

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768	768	768	768	768	768	768	768	768
mean	3.8451	120.8945	69.1055	20.5365	79.7995	31.9926	0.4719	33.2409	0.349
std	3.3696	31.9726	19.3558	15.9522	115.244	7.8842	0.3313	11.7602	0.477
min	0	0	0	0	0	0	0.078	21	0
25%	1	99	62	0	0	27.3	0.2438	24	0
50%	3	117	72	23	30.5	32	0.3725	29	0
75%	6	140.25	80	32	127.25	36.6	0.6263	41	1
max	17	199	122	99	846	67.1	2.42	81	1

Data Types and Null Values

	Data Type	Null Values	Non-null Count
Pregnancies	int64	No null values	768
Glucose	int64	No null values	768
BloodPressure	int64	No null values	768
SkinThickness	int64	No null values	768
Insulin	int64	No null values	768
BMI	float64	No null values	768
DiabetesPedigre	float64	No null values	768
Age	int64	No null values	768
Outcome	int64	No null values	768

2. Exploratory Data Analysis

Deploy

- Data Ingestion
- Exploratory Data Analysis
- Data Cleaning
- Feature Importance
- Feature Scaling
- Model Selection
- Model Tuning
- Ensembling
- Report

Highly Correlated Features

	Feature 1	Feature 2	Correlation
0	Pregnancies	Age	0.5443
1	Glucose	Outcome	0.4666
2	Insulin	SkinThickness	0.4368
3	BMI	SkinThickness	0.3926
4	Glucose	Insulin	0.3314
5	BMI	Outcome	0.2927
6	BloodPressure	BMI	0.2818
7	Glucose	Age	0.2635
8	BloodPressure	Age	0.2395
9	Age	Outcome	0.2384

Interpretation of Correlations

Understanding Correlation:

Correlation values range from **-1** to **1**:

- **Positive Correlation** (closer to 1): As one feature increases, the other feature tends to increase. Example: Higher study hours leading to better grades.
- **Negative Correlation** (closer to -1): As one feature increases, the other feature tends to decrease. Example: More exercise might result in lower body fat percentage.
- **No Correlation** (closer to 0): Minimal or no linear relationship between features. Example: Shoe size vs. exam scores.

Importance for Predictive Modeling:

- Strong correlations (values near ±1) indicate a significant relationship and are often key features for prediction.
- High correlation between independent features can lead to multicollinearity, which may require addressing by removing or combining features to avoid redundancy and overfitting.

2. Exploratory Data Analysis

- Data Ingestion
- **Exploratory Data Analysis**
- Data Cleaning
- Feature Importance
- Feature Scaling
- Model Selection
- Model Tuning
- Ensembling
- Report

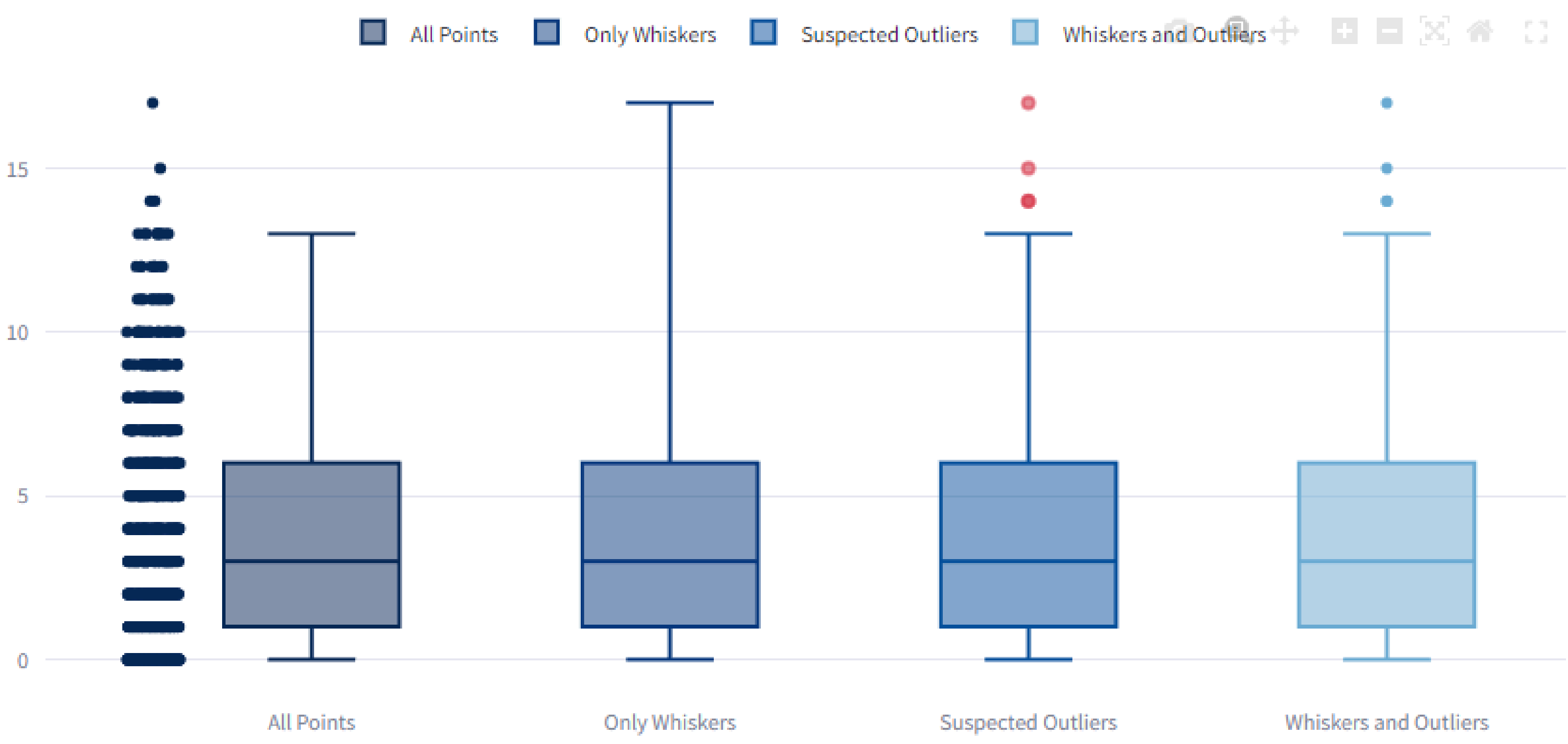
☑ Outlier Investigation

Single Feature Outliers

Select feature for outlier detection

Pregnancies

Boxplot for : Pregnancies



Understanding Box Plots and Outliers

- **Box**: Represents the interquartile range (IQR), which contains the middle 50% of the data.Lower Edge: 1st Quartile (Q1). Upper Edge: 3rd Quartile (Q3). Horizontal Line inside the Box: Median
- **Whiskers**: The lines that extend from the box to the smallest and largest values within 1.5 * IQR.
- **Outliers**: Data points that lie outside the whisker range are considered outliers and are displayed as individual dots.
- **All Points**: Every data point, including outliers.
- **Only Whiskers**: Displays only the key data range within the whiskers, hiding outliers.
- **Suspected Outliers**: Data points that fall outside 1.5 * IQR but aren't extreme enough to be definite outliers.
- **Whiskers and Outliers**: Displays both the whiskers and any definite outliers.

3. Data Cleaning

Deploy

Data Ingestion
Exploratory Data Analysis
Data Cleaning
Feature Importance
Feature Scaling
Model Selection
Model Tuning
Ensembling
Report

Data Cleaning

Outlier Detection Methods

Outlier detection helps identify extreme values in the dataset. Below are two common methods:

1. **Tukey's Method**: Uses the interquartile range (IQR) to detect outliers. It is robust to extreme values and identifies outliers outside the range [Q1 - 1.5*IQR, Q3 + 1.5*IQR].
2. **Z-score Method**: Identifies outliers based on how many standard deviations a data point is from the mean. A common threshold is 3, meaning points more than 3 standard deviations away from the mean are flagged as outliers.

☑ Remove Outliers

Select features to remove outliers from

Pregnancies ×

Select outlier removal method

Tukey's Method

Total number of outliers removed: 4

New dataset has 764 samples and 9 features.

Feature 'Pregnancies': 4 outliers found using Tukey's Method.

☑ Handle Missing Values

No missing values found in the dataset.

☑ Data Type Conversion

Select features for data type conversion

Choose an option

4. Feature Engineering

Data Ingestion
Exploratory Data Analysis
Data Cleaning
Feature Importance
Feature Scaling
Model Selection
Model Tuning
Ensembling
Report

Deploy

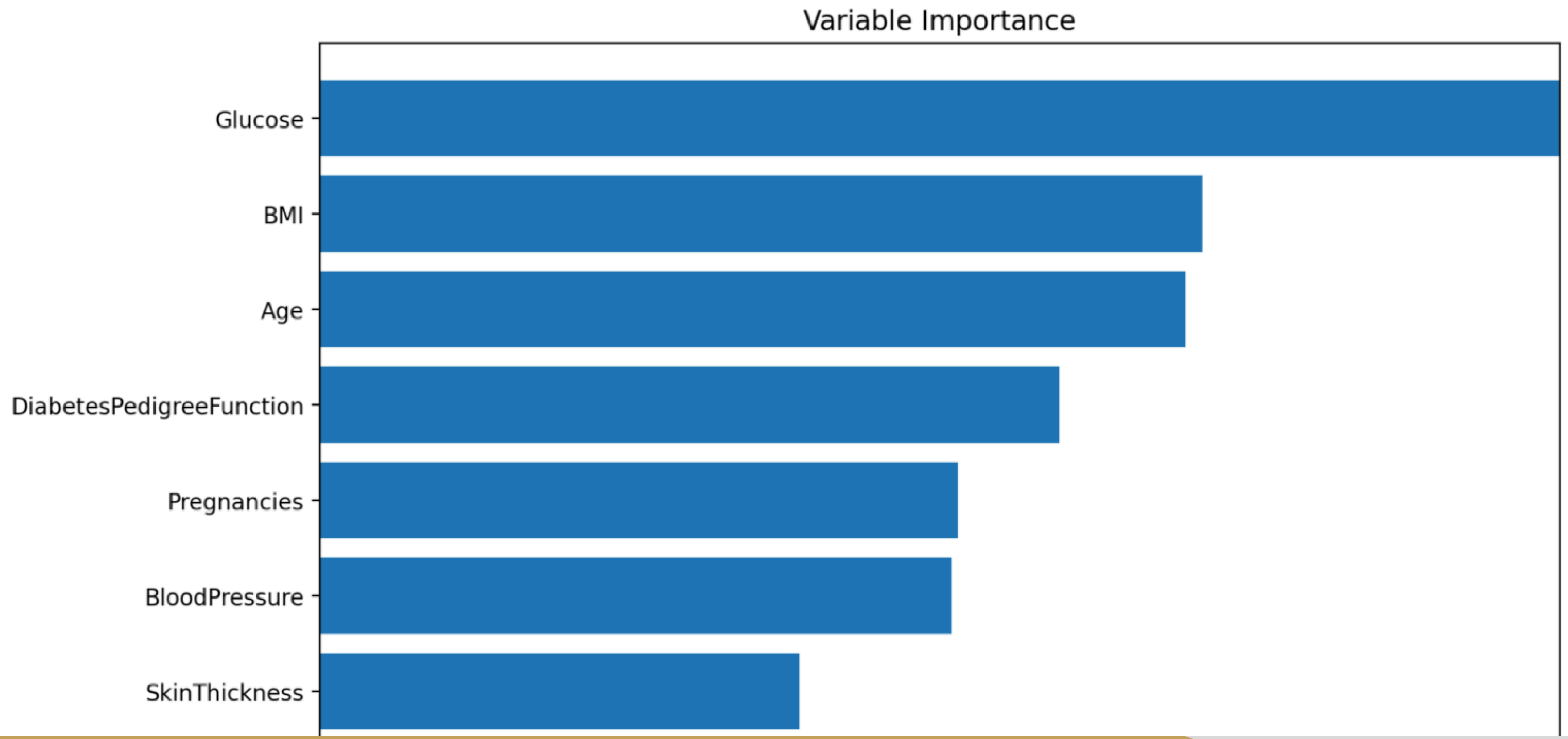
Feature Engineering

Select Target Variable

Select the target variable:

Outcome

Independent features: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age



5. Feature Scaling

Data Ingestion
Exploratory Data Analysis
Data Cleaning
Feature Importance
Feature Scaling
Model Selection
Model Tuning
Ensembling
Report

Deploy

Feature Scaling / Normalization

Scaling will be applied to the following features: Age, BMI, Glucose

The target variable is: Outcome (which will not be scaled).

Choose a scaling method:

- (•) MinMaxScaler
- ( ) StandardScaler

MinMaxScaler scales the features to a fixed range, usually [0,1].

Scaled Data Preview (including only selected features and target variable):

	Age	BMI	Glucose	Outcome
0	0.4833	0.5007	0.7437	1
1	0.1667	0.3964	0.4271	0
2	0.1833	0.3472	0.9196	1
3	0	0.4188	0.4472	0
4	0.2	0.6423	0.6884	1
5	0.15	0.3815	0.5829	0
6	0.0833	0.462	0.392	1
7	0.1333	0.5261	0.5779	0
8	0.5333	0.4545	0.9899	1
9	0.55	0	0.6281	1

Download Scaled Data CSV

6. Model Selection

Data Ingestion
Exploratory Data Analysis
Data Cleaning
Feature Importance
Feature Scaling
Model Selection
Model Tuning
Ensembling
Report

Deploy

Model Selection & Baseline Algorithm Evaluation

Model will be trained on features: Age, BMI, Glucose

The target variable is: Outcome

Binary Classification

The target variable `Outcome` has exactly two unique values, indicating a binary outcome. Therefore, the problem can be modeled as a binary classification task, where the goal is to predict whether an individual falls into one of two categories.

Suggested Models for Binary Classification

The following models are suggested for binary classification tasks:

- Logistic Regression
- K-Nearest Neighbors
- Support Vector Machine
- Decision Tree
- AdaBoost
- Gradient Boosting
- Random Forest
- Extra Trees

Metric Comparison Across Models

Model	Accuracy	Precision	Recall	F1-score	ROC AUC
Logistic Regression	0.8182	0.8421	0.5926	0.6957	0.8452
K-Nearest Neighbors	0.7792	0.7273	0.5926	0.6531	0.807
Support Vector Machine	0.7922	0.7895	0.5556	0.6522	0.877
Decision Tree	0.6753	0.5333	0.5926	0.5614	0.6563
AdaBoost	0.8312	0.8889	0.5926	0.7111	0.8893
Gradient Boosting	0.7922	0.7619	0.5926	0.6667	0.8681
Random Forest	0.7532	0.7	0.5185	0.5957	0.8352
Extra Trees	0.7792	0.75	0.5556	0.6383	0.8189

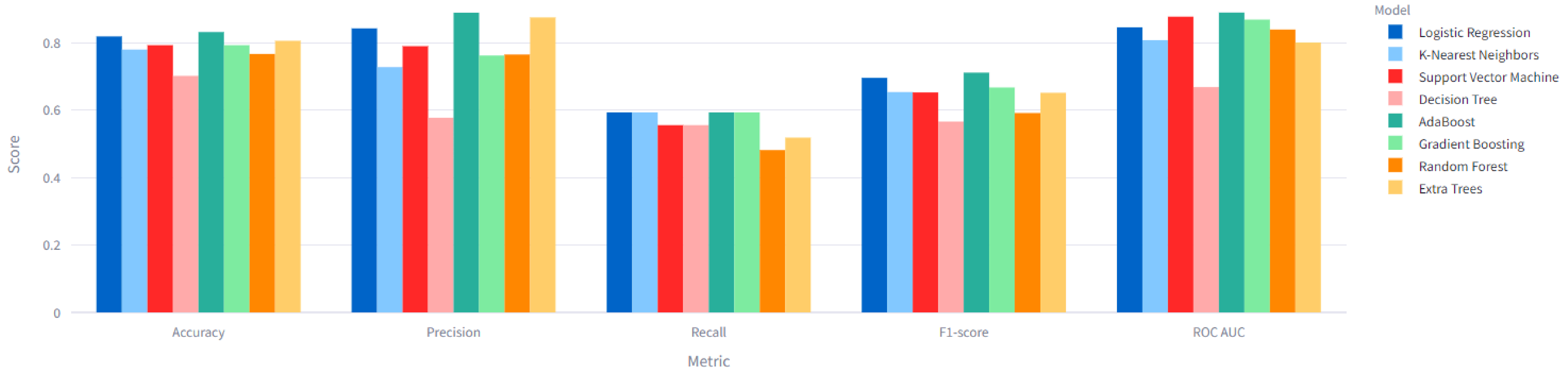
6. Model Selection

Deploy

- Data Ingestion
- Exploratory Data Analysis
- Data Cleaning
- Feature Importance
- Feature Scaling
- **Model Selection**
- Model Tuning
- Ensembling
- Report

Comparison Plot of Model Performance

Model Performance Comparison



Select Models for Ensembling

Choose one or more models for further evaluation:

Logistic Regression × K-Nearest Neigh... × Support Vector ... ×

Selected models saved: Logistic Regression, K-Nearest Neighbors, Support Vector Machine

You have selected the following models for further evaluation:

Logistic Regression, K-Nearest Neighbors, Support Vector Machine

7. Model Tuning

Deploy

Data Ingestion
Exploratory Data Analysis
Data Cleaning
Feature Importance
Feature Scaling
Model Selection
Model Tuning
Ensembling
Report

Selected Models for Hyperparameter Tuning


```
[
  0 : "Logistic Regression"
  1 : "K-Nearest Neighbors"
  2 : "Support Vector Machine"
]
```


Hyperparameter Tuning Methods

- [x] Auto Tuning Hyperparameters
- [ ] Manual Tuning Hyperparameters

Auto-Tuning Hyperparameters

Tuning: Logistic Regression

Tuning: K-Nearest Neighbors

Tuning: Support Vector Machine

Best Hyperparameters for Each Model:

Logistic Regression: {'penalty': 'l2', 'C': 0.1}

K-Nearest Neighbors: {'n_neighbors': 7, 'weights': 'uniform'}

Support Vector Machine: {'C': 10, 'kernel': 'poly'}

Data Ingestion

Exploratory Data Analysis

Data Cleaning

Feature Importance

Feature Scaling

Model Selection

Model Tuning

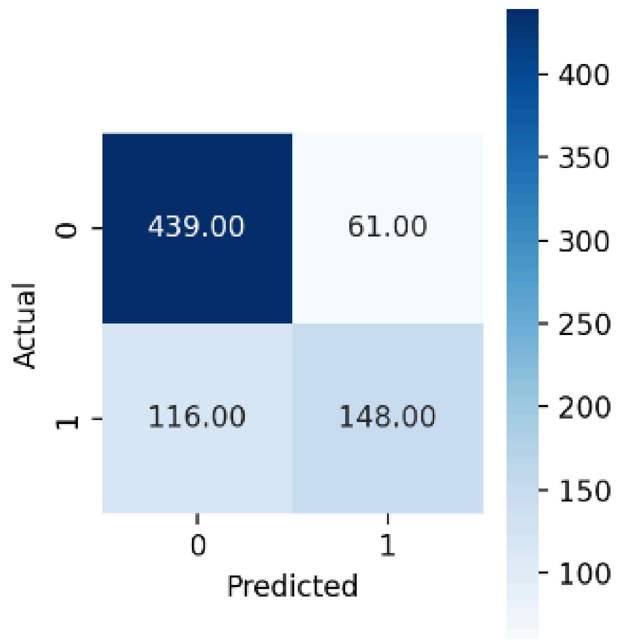
Ensembling

Report

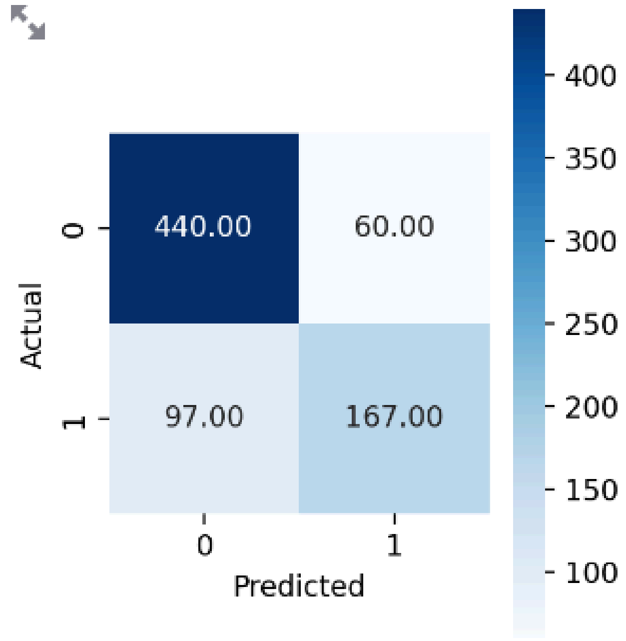
Deploy

Evaluation Metrics for Tuned Models:

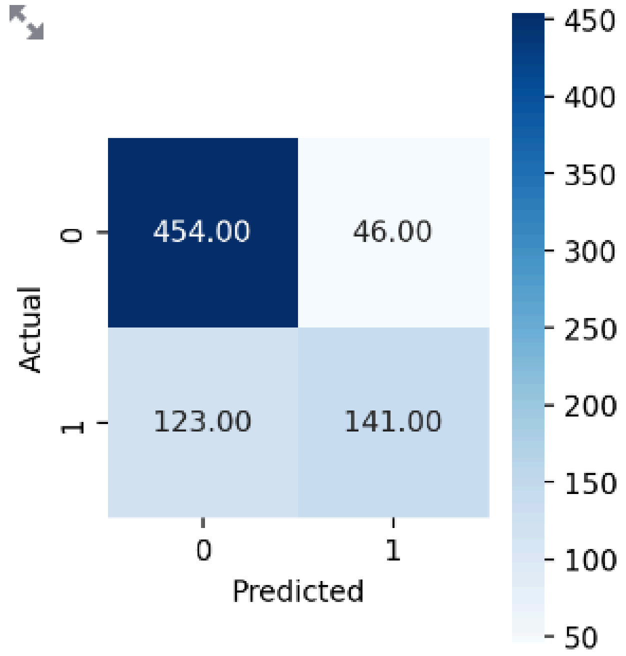
Logistic Regression Confusion Matrix



K-Nearest Neighbors Confusion Matrix



Support Vector Machine Confusion Matrix



Classification Reports:

Classification Report for Logistic Regression:

	precision	recall	f1-score	support
0	0.7910	0.8780	0.8322	500.0000
1	0.7081	0.5606	0.6258	264.0000
accuracy	0.7683	0.7683	0.7683	0.7683
macro avg	0.7496	0.7193	0.7290	764.0000
weighted avg	0.7624	0.7683	0.7609	764.0000

8.Ensembling

Data Ingestion
Exploratory Data Analysis
Data Cleaning
Feature Importance
Feature Scaling
Model Selection
Model Tuning
Ensembling
Report

Deploy

Super Learner Training

- [x] Select All Models

Selected models:

[
0 : "Logistic Regression"
1 : "K-Nearest Neighbors"
2 : "Support Vector Machine"
]

Train Super Learner

Super Learner trained successfully!

Super Learner Accuracy: 0.8438

Super Learner Precision: 0.8571

Super Learner Recall: 0.6000

Super Learner F1-score: 0.7059

Super Learner ROC AUC: 0.7773

9. Report

Report for model evaluation

How Models Perform as per Model,Accuracy,Precision,Recall,F1-score,ROC AUC

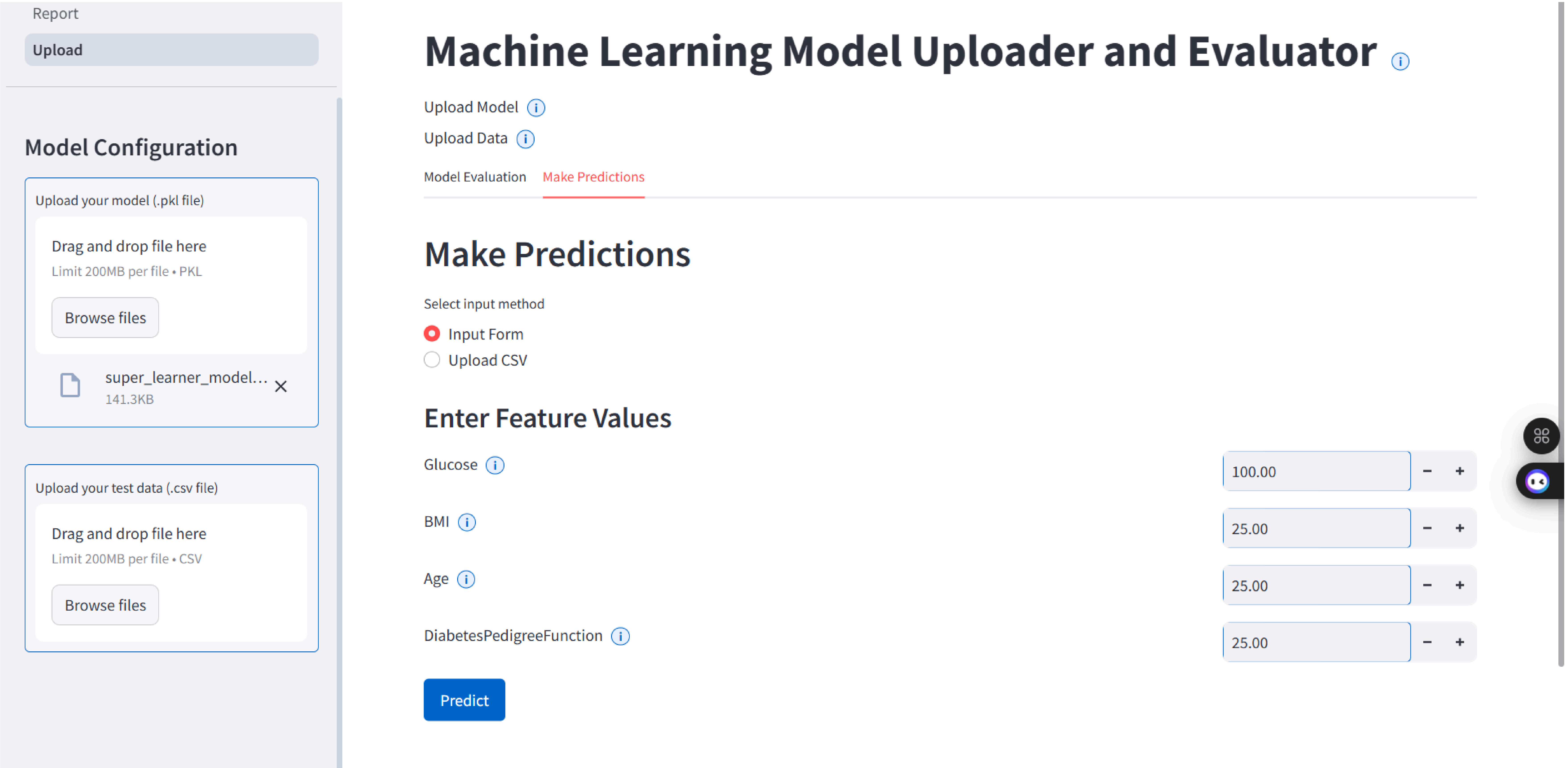
What a treasure trove of insights!

Here are some conclusions and observations that can be drawn from this data:

1. **AdaBoost** and **Random Forest** are strong performers, with high accuracy (around 79%), precision (73%), and F1-score (62-52%). These ensemble methods tend to perform well on complex datasets.
2. **Logistic Regression** surprisingly underperforms on precision (80%), but still maintains high accuracy (82.8%) and F1-score (69%). This might be due to the dataset's complexity or sheer size.
3. **K-Nearest Neighbors** is consistently struggling, with relatively low accuracy (64%), precision (42%), and F1-score (41%). This could be due to the dataset's non-linear relationships or high dimensionality.
4. **Support Vector Machine, Decision Tree, and Gradient Boosting** lie in the middle, with accuracy ranging from 75% (SVM) to 72.5% (Gradient Boosting). These models might be suitable for smaller datasets or when interpretability is crucial.
5. **ROC AUC** (Area Under the Receiver Operating Characteristic Curve) is above 82.5% for all models except K-Nearest Neighbors (70.6%). This suggests that most models are relatively good at distinguishing between positive and negative classes.
6. **Precision-Recall trade-off**: AdaBoost and Random Forest have relatively high precision (73-73%) and mid-range recall (55-40%). This suggests that they prioritize accuracy over recall, whereas Logistic Regression and Decision Tree have higher recall (60-35%) at the cost of precision.
7. **Variability in performance**: While some models (AdaBoost, Random Forest) consistently perform well across all metrics, others (K-Nearest Neighbors, Support Vector Machine) are more unpredictable.
8. **Dataset characteristics**: Without knowing the specifics of the dataset, we can't directly attribute the performance of these models. However, it's possible that the dataset is relatively simple (for AdaBoost and Random Forest) or complex (for K-Nearest Neighbors).

These observations provide a solid foundation for refining your machine learning pipeline, exploring different models and hyperparameters, and learning from the strengths and weaknesses of each approach.

10. Testing Dashboard



CONCLUSION

- **Data Processing** - Implemented CSV data ingestion module
- **EDA** - Implemented Descriptive statistics, data visualization capabilities, and Outlier Investigation
- **Data Preprocessing** - Removing Outlier, Handling Missing Values & Data Type Conversion
- **Feature Importance** - Extra Tree classifier-based feature importance to list top important features.
- **Feature Scaling** - MinMax Scaling and Standard Scaling
- **Model Selection** - Binary Classification and Ensembling Classifiers.
- **Hyperparameter Tuning** - Autotuning with GridSearchCV and Manual Tuning
- **Training & Evaluation** - Classification report, Confusion Metrics, and other metrics.
- **Report** - Summary of Model evaluation and model selection.
- **Model Testing and Uploading Module** - Dashboard to upload pretrained model and visualize predictions

PATENTABILITY OF PROJECT/COPYRIGHT

Potential Patentable Aspects & Copyright Considerations

- The unique arrangement and selection of ML algorithms in our pipeline
- Integrating user expertise and inputs with automated decisions
- User-friendly interface and layout accessible to both novices and experts
- Source code implementing the whole process flow.
- LLM-based Report Generation for summaries of end-to-end pipeline functions performed
- Option for users to download the trained model (.tflite or .pkl files)
- Ability to Uplad pretrained models and visualise the prediction output in realtime.

SOCIAL UTILITY

- Time and Resource Savings
  - Automates ML lifecycle and repetitive tasks, saving hours of manual work
  - Enables quicker iterations and experimentation with different models
  - Less coding more focus on high-value tasks and interpretation

- Efficiency and Quality
  - Facilitates reproducibility of results across different projects
  - Enables easy comparison of multiple models for optimal selection
  - Improves overall model performance through systematic optimization

References

[1] Urbanowicz RJ, Zhang R, Cui Y, Suri P. STREAMLINE: a simple, transpar- ent, end-to-end automated machine learning pipeline facilitating data analysis and algorithm comparison. ArXiv220612002 2022.

[2] Gijsbers, P., Vanschoren, J. (2021). GAMA: A General Automated Machine Learning Assistant. In: Dong, Y., Ifrim, G., Mladenić, D., Saunders, C., Van *Hoecke, S. (eds) Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track. ECML PKDD 2020. Lecture Notes in Computer Science(), vol 12461*. Springer, Cham. https://doi.org/10.1007/978-3-030-67670-4_39

[3] B. Derakhshan, A. R. Mahdiraji, T. Rabl, and V. Markl, "Continuous deployment of machine learning pipelines," *Advances in Database Technology - EDBT, vol. 2019-March, pp. 397–408, 2019*, doi: 10.5441/002/edbt.2019.35.

FLOW DIGRAM OF ML PIPELINE

Streamlit

Data Input

XLSX + CSV

Data /Feature Store

Exploratory Data Analysis EDA

- **Descriptive Data Statistics**
  - Data Description
  - Data Type & Null values
- **Data Visualization**
  - Correlation Heatmap
  - Scatter Plot
  - Interpretation of Correlations
- **Outlier Investigation**
  - Single Feature Outliers
  - Boxplot
  - Descriptive Analysis

Data Cleaning

- **Outlier Removing**
  - TukeyOutliers
  - Z-Score Method
- **Handle Missing Values**
  - Random Sample Imputation
  - Mean Imputation
  - Median Inputation
  - Drop
  - Replace
- **Data Type Conversion**
  - int
  - float
  - string
  - datetime

Feature Engineering

- **Target Variable Selection**
  - Variable Importance Visualisation
- **Best Feature Selection for model training**

Feature Scaling / Normalization

- Feature Scaling
  - MinMaxScaler
  - StandardScaler

Model Selection & Baseline Algorithm Evaluation

- **Classification Type Suggestion**
  - Binary
  - Multiclass
- **Comparison Plot of Model Performance**
  - Accuracy
  - Precision
  - Recall
  - F1-Score
  - ROC AUC
- **Models Selection for Ensembling**

Model Tuning

- **Hyperparameter Tuning**
  - Auto Tuning
  - Manual Tuning
- **Evaluation Metrics for Tuned Models**
- **Classification Reports for models**

Ensembling

- **Ensembled Base Model Evaluation**
  - Classification Report
  - Evaluation Plots
- **Training Super Learner**

Report & Model Generation

Model Testing & Visualisation module

- Make Predictions
- Enter Feature Values manually


THANK YOU!