OSTIM TECHNICAL UNIVERSITY
ENGINEERING FACULTY

APPLYING MACHINE LEARNING ON STREAMLIT

SEMESTER PROJECT

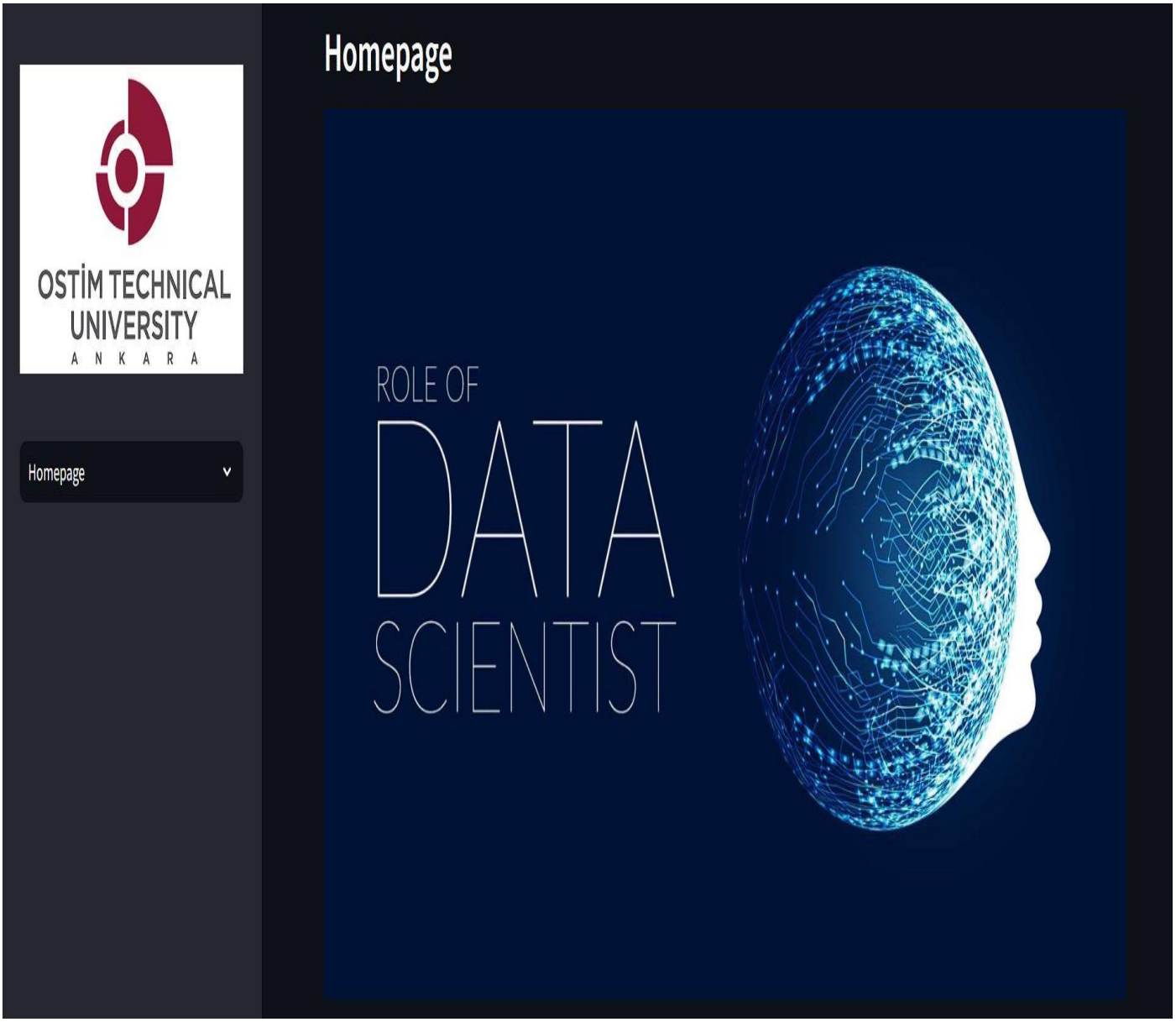
Emil YOLCHİYEV
200201834

Aslıhan YILDIRIM
190201026

COMPUTER ENGINEERING

ANKARA, 2024

GRADUATION PROJECT REPORT



TERM PROJECT ACCEPTANCE AND APPROVAL

Student Name and Surname: Emil Yolchiyev, Aslıhan Yıldırım

Student No.: 200201834,190201026

Department: Computer Engineering

The student, whose explicit information is given above, realized in the Spring Semester of the 2023/2024 Academic Year “Applying Machine Learning on Streamlit” his titled work has been accepted as a bachelor's semester project.

Advisor Approval	**Dean Manager Approval**
(Name, Surname and Signature)	(Name, Surname and Signature)
Date of Approval:	**Date of Approval:**

ÖZET

Bu proje, çeşitli veri kümeleri üzerinde keşifsel veri analizi ve makine öğrenimi modellemesi yapmak için Streamlit kullanarak web tabanlı bir uygulama geliştirmeyi amaçlamaktadır. Uygulama, veri kümelerini yüklemek, verileri görselleştirmek, verileri önceden işlemek ve makine öğrenimi modellerini eğitmek için kullanıcı dostu bir arayüz sağlar. Birincil amaç, kullanıcıların veri odaklı kararlar almalarına yardımcı olmak için model seçimi ve değerlendirme sürecini otomatikleştirmektir. Uygulama birden fazla veri kümesini destekler ve veri modellerine ve model performansına ilişkin görsel içgörüler sunar. Kullanıcılar, uygulama aracılığıyla veri kümelerini kolayca karşılaştırabilir ve analiz sonuçlarını etkileşimli grafikler ve tablolarla görüntüleyebilir. Uygulama, veri temizleme, öznitelik mühendisliği ve veri normalizasyonu gibi ön işleme adımlarını kullanıcı dostu bir şekilde sunar. Ayrıca, farklı makine öğrenimi algoritmalarını kullanarak modelleri eğitme ve değerlendirme imkanı sağlar. Kullanıcılar, model performansını çeşitli metriklerle değerlendirebilir ve en iyi performans gösteren modeli seçebilir. Uygulama, eğitimli modelleri kaydetme ve ileride kullanmak üzere yükleme işlevselliği sunarak, veri analitiği süreçlerini daha verimli hale getirir. Bu sayede, veri bilimi ve makine öğrenimi alanında her seviyeden kullanıcının daha etkili ve hızlı bir şekilde çalışmasına olanak tanır.

Anahtar Kelimeler: Makine Öğrenmesi, Veri Analizi, Veri Ön İşleme,Veri Görselleştirme, Streamlit

ABSTRACT

This project aims to develop a web-based application using Streamlit for performing exploratory data analysis and machine learning modeling on various datasets. The application provides a user-friendly interface to upload datasets, visualize data, preprocess data, and train machine learning models. The primary objective is to automate the process of model selection and evaluation to assist users in making data-driven decisions. The application supports multiple datasets and offers visual insights into data patterns and model performance. Users can easily compare and analyze datasets through interactive graphs and tables provided by the application. The application includes features for data cleaning, feature engineering, and data normalization, making preprocessing steps straightforward and accessible. It supports training and evaluating models using various machine learning algorithms, allowing users to assess model performance using multiple metrics. Additionally, users can save and load trained models for future use, enhancing the efficiency of data analytics workflows. This project aims to cater to users of all levels, enabling more effective and rapid work in the field of data science and machine learning.

Key Words: Machine Learning, Data Analysis, Data Preprocessing, Data Visualization, Streamlit

ACKNOWLEDGEMENT

We would like to express our deepest gratitude to our project advisor, Assoc. Prof. Dr. Murat Şimşek, for their invaluable guidance and support throughout this project. Their expertise, insightful feedback, and encouragement have been instrumental in shaping the direction and success of our work. We are profoundly grateful for their dedication and for always being available to assist us with any challenges we encountered. Their mentorship has not only enhanced our academic growth but also inspired us to strive for excellence

Table of Contents

1.Introduction

In the era of big data, the ability to analyze and visualize data effectively has become crucial in various fields. This graduation project aims to harness the power of Streamlit, an open-source Python library, to create an interactive web application for data analysis and visualization. The project focuses on five key datasets: water potability, loan prediction, Income, Diabetes, Credit Card defaults . By analyzing these datasets, the application aims to provide insights and predictions that can be useful for decision-making processes.

The scope of this project includes data preprocessing, feature engineering, model training, and the development of an interactive user interface. The application allows users to explore the data visually and make predictions based on the trained models. This report details each step of the project, providing a comprehensive overview of the methods and tools used, as well as the results obtained.

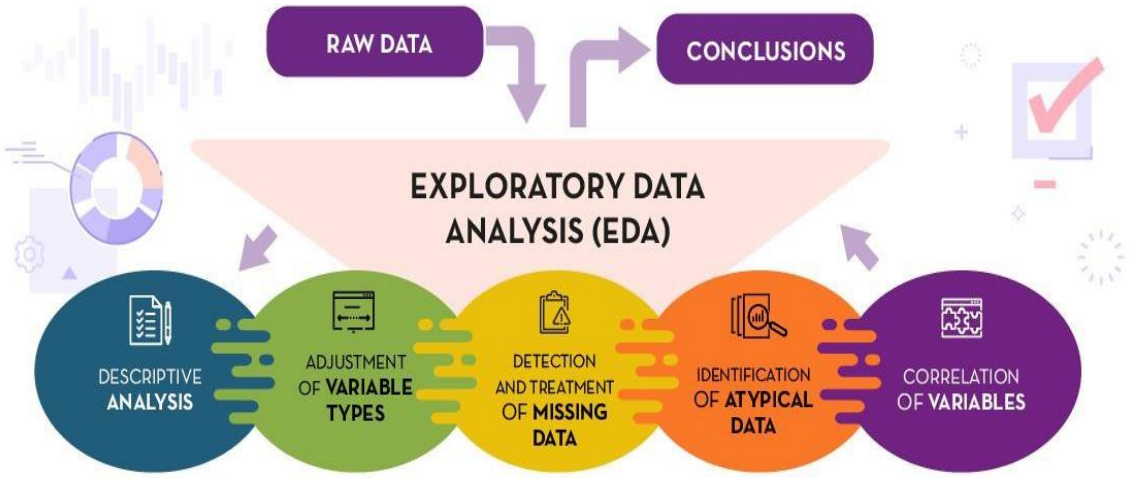
Streamlit was chosen for this project due to its simplicity and effectiveness in creating interactive web applications directly from Python scripts. The project involved several stages, starting with data collection and preprocessing, followed by exploratory data analysis (EDA), model selection, and implementation of the web application.

The project highlights the importance of data-driven decision-making and demonstrates the practical application of data science techniques in real-world scenarios. By providing an interactive platform for data analysis, the application aims to empower users with valuable insights and facilitate better decision-making processes.

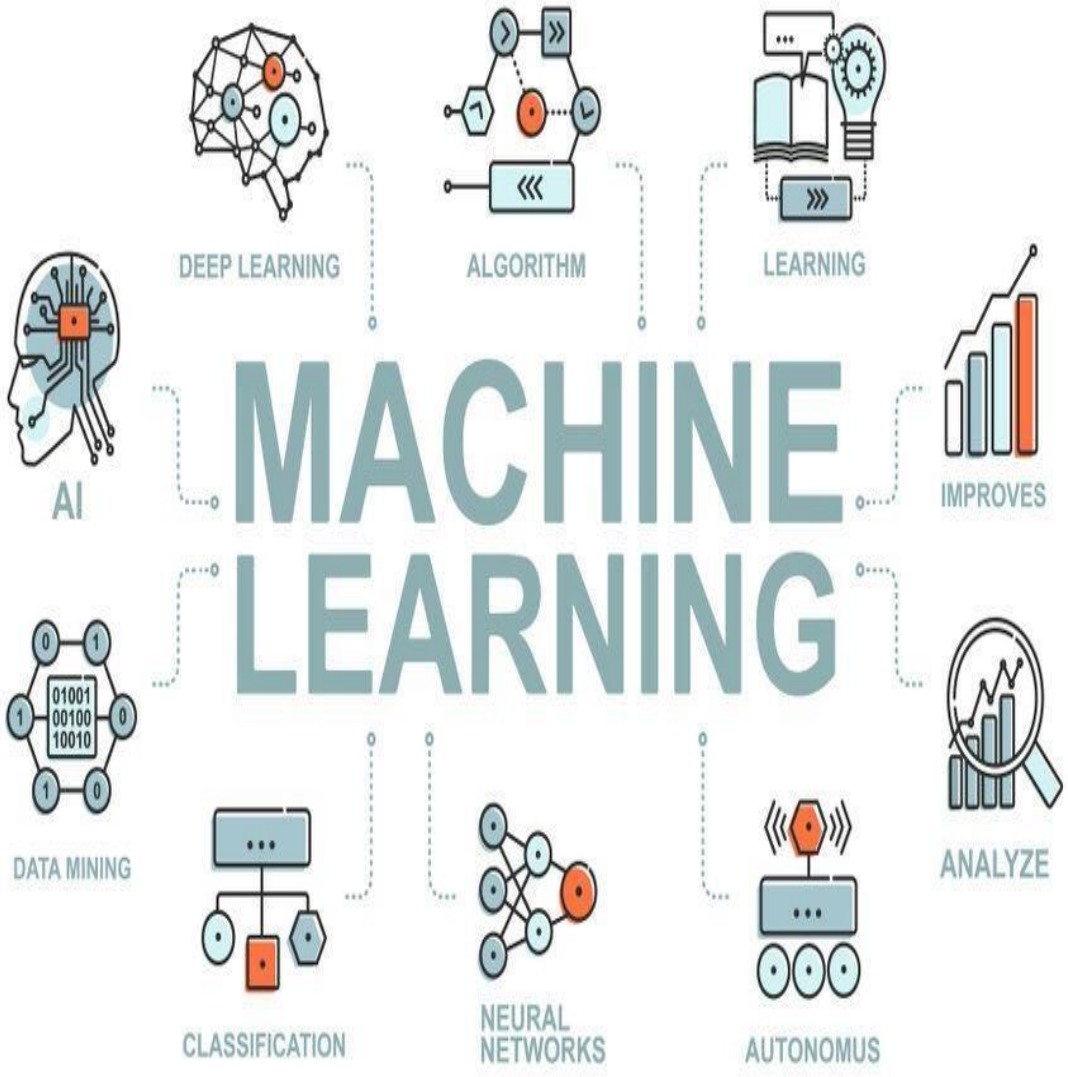
2. List of Tables

List of Symbols and Abbreviations

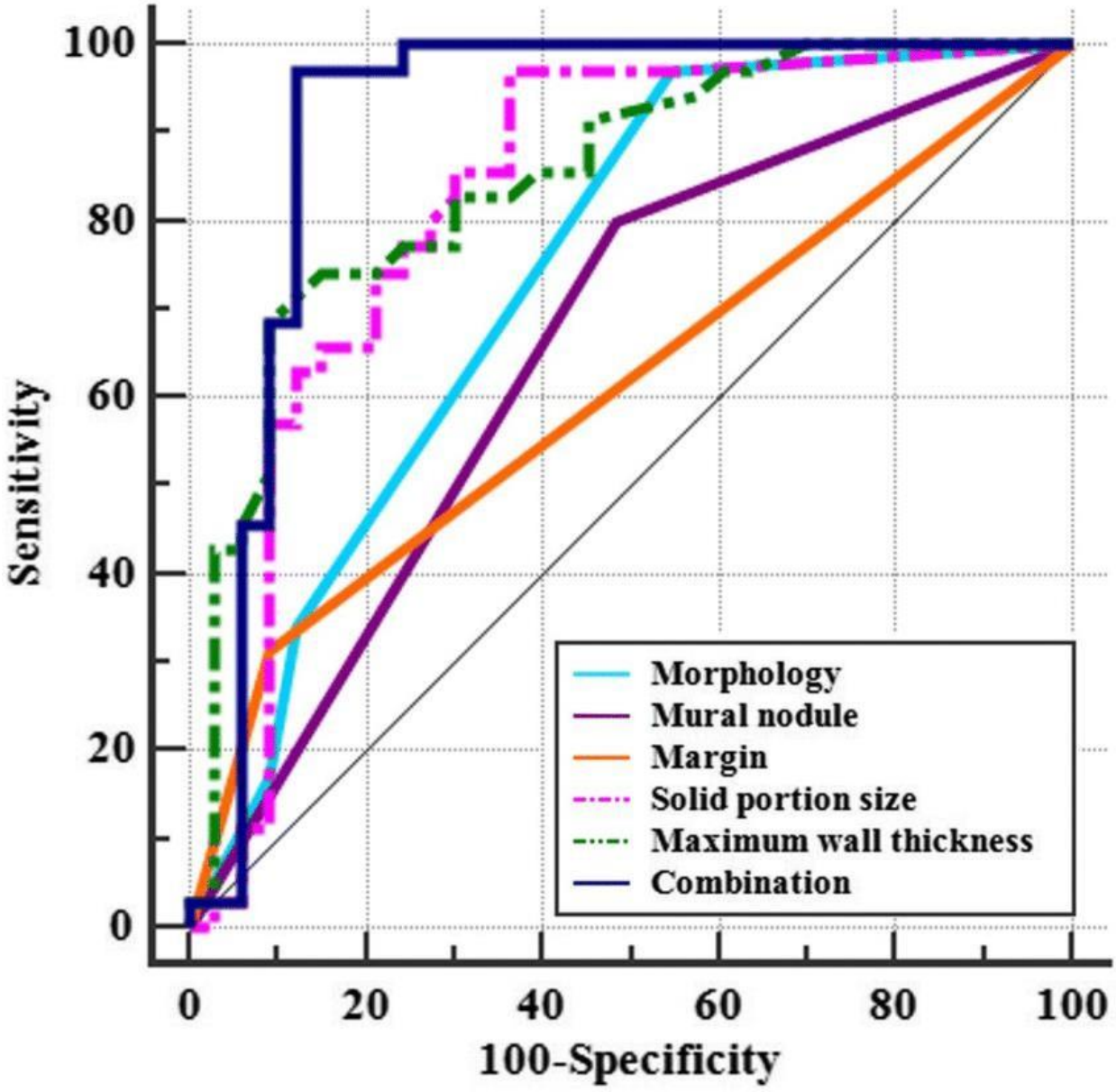
- EDA: Exploratory Data Analysis



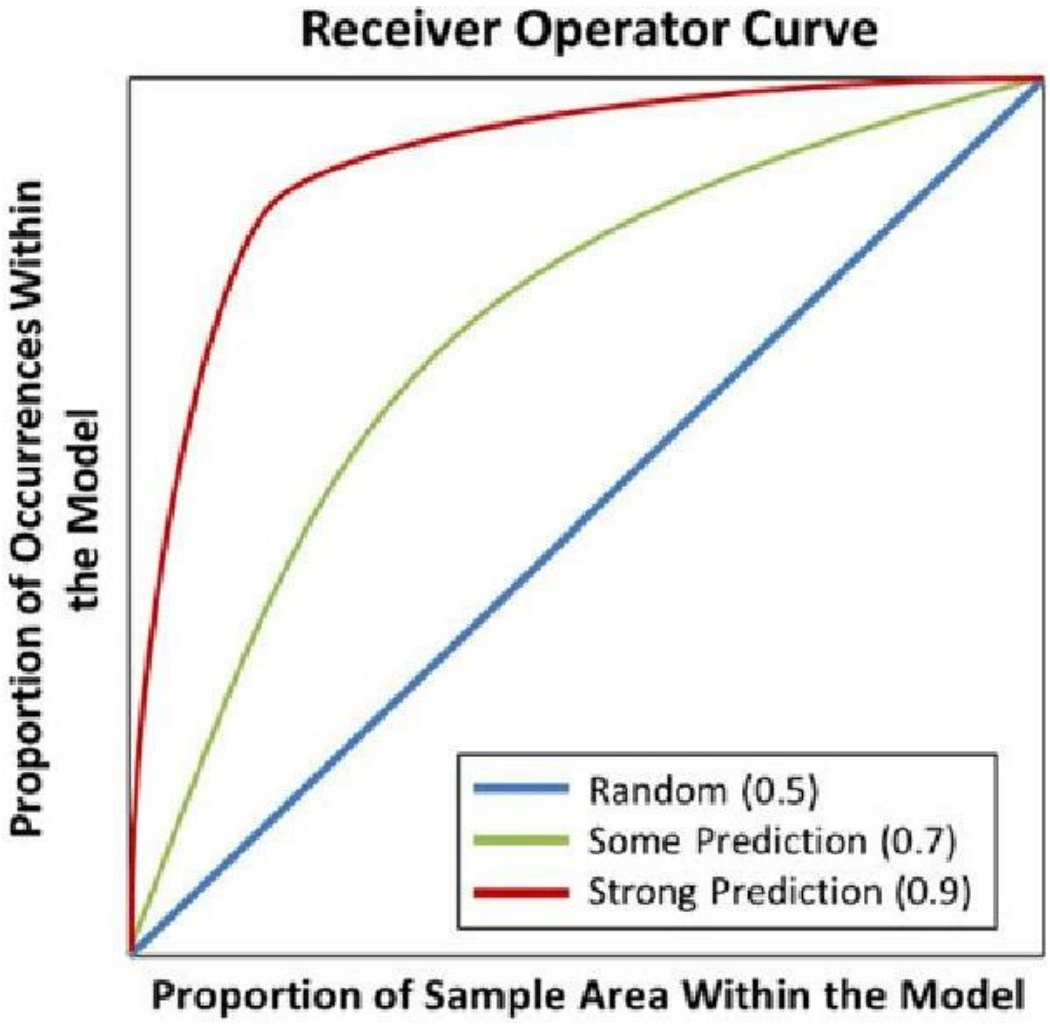
- ML: Machine Learning



ROC: Receiver Operating Characteristic



- AUC: Area Under the Curve



3.Project Overview

The primary objective of this project is to develop a Streamlit-based web application that can perform data analysis and visualization on five datasets: water potability and loan prediction, Income, Diabetes, Credit Card defaults. Streamlit is chosen for its simplicity and efficiency in creating interactive web applications directly from Python scripts.

Water Potability Dataset

The Water Potability dataset contains information about various water quality parameters and indicates whether the water is potable (safe to drink). This dataset includes measurements such as pH level, hardness, solids, chloramines, sulfates, conductivity, organic carbon, trihalomethanes, and turbidity. The primary goal is to predict water potability based on these physical and chemical parameters, making it useful for water quality assessment and ensuring public health safety.

Loan Prediction Dataset

The Loan Prediction dataset includes features related to applicants' demographic information, financial details, and loan application status. This dataset contains information such as the applicant's gender, marital status, education, employment status, income, loan amount, loan term, credit history, and property area. The primary objective is to predict whether a loan application will be approved based on these factors, assisting financial institutions in automating the loan approval process and assessing credit risk.

Income Dataset

The Income dataset contains demographic information about individuals, such as age, work class, education, marital status, occupation, relationship, race, sex, capital gain, capital loss, hours per week, and native country. The primary goal is to predict whether an individual's income exceeds a certain threshold (e.g., $50,000 per year). This dataset is commonly used for classification tasks, where the target variable is binary (e.g., income > $50K or income <=
$50K).

Diabetes Dataset

The Diabetes dataset comprises diagnostic measurements and medical details of patients. It includes features such as pregnancies, glucose level, blood pressure, skin thickness, insulin level, BMI, diabetes pedigree function, age, and the target variable indicating whether the patient has diabetes. This dataset is used to predict the onset of diabetes based on these medical attributes, making it a valuable resource for healthcare applications and disease prediction models.

Credit Card Default Dataset

The Credit Card Default dataset contains information about credit card holders, including their demographic details, credit history, and financial behavior. Features in this dataset include age, gender, education level, marital status, payment history, bill statement amounts, previous payments, and the target variable indicating whether the individual defaulted on their credit card payment. This dataset is used for classification tasks to predict the likelihood of a credit card default, aiding financial institutions in assessing credit risk and making lending decisions.

The application provides users with functionalities to explore the datasets through visualizations, such as histograms, scatter plots, and box plots. Additionally, it includes predictive models to determine water potability and loan approval status based on user inputs.

The project involved the following key steps:

1. Data Collection: Gathering the water potability and loan prediction datasets.
2. Data Preprocessing: Cleaning the data, handling missing values, and encoding categorical variables.
3. Exploratory Data Analysis (EDA): Analyzing the datasets to understand their structure and key characteristics.
4. Feature Selection: Identifying the most relevant features for the predictive models.
5. Model Training: Training various machine learning models to predict water potability and loan approval status.
6. Application Development: Building the Streamlit application to provide an interactive interface for data exploration and prediction.
7. Evaluation: Assessing the performance of the models using various metrics and refining the application based on user feedback.

This project demonstrates the practical application of data science techniques and highlights the potential of Streamlit in creating user-friendly, data-driven web applications.

4.Data Description

This project utilizes 5 datasets:

1. Water Potability Dataset:

This dataset contains information about water quality parameters such as pH, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic_carbon, Trihalomethanes, and Turbidity. The target variable is 'Potability', which indicates whether the water is safe to drink. The dataset includes several key features:

- **pH**: The measure of how acidic/basic water is.
- **Hardness**: The measure of the concentration of calcium and magnesium ions in water.
- **Solids**: Total dissolved solids (TDS) in water.
- **Chloramines**: The amount of chlorine compounds present in water.
- **Sulfate**: The concentration of sulfate ions in water.
- **Conductivity**: The ability of water to conduct an electric current.
- **Organic_carbon**: The amount of organic carbon in water.
- **Trihalomethanes**: Chemical compounds that can be formed during water chlorination.
- **Turbidity**: The clarity of water.

2. Loan Prediction Dataset:

This dataset includes features related to applicants' demographic information, financial details, and loan application status. The target variable is 'Loan_Status', indicating whether the loan was approved or not. The dataset includes several key features:

- **Gender**: Gender of the applicant.
- **Marital Status**: Whether the applicant is married.
- **Education**: Education level of the applicant.
- **Self_Employed**: Whether the applicant is self-employed.
- **ApplicantIncome**: Income of the applicant.
- **CoapplicantIncome**: Income of the co-applicant, if any.
- **LoanAmount**: Amount of the loan applied for.
- **Loan_Amount_Term**: Term of the loan in months.
- **Credit_History**: Credit history of the applicant.
- **Property_Area**: Area type of the property (Urban, Semiurban, Rural).

3. Income Dataset:

This dataset contains demographic information about individuals, aiming to predict whether their income exceeds a certain threshold (e.g., $50,000 per year). The

target variable is 'Income', indicating high or low income. The dataset includes several key features:

- **Age**: Age of the individual.
- **Work Class**: Category of employment (e.g., Private, Self-Employed).
- **Education**: Level of education attained.
- **Marital Status**: Marital status of the individual.
- **Occupation**: Type of job the individual has.
- **Relationship**: Relationship status within a family.
- **Race**: Race of the individual.
- **Sex**: Gender of the individual.
- **Capital Gain**: Income from capital gains.
- **Capital Loss**: Losses from capital.
- **Hours per Week**: Number of hours worked per week.
- **Native Country**: Country of origin.

4. Diabetes Dataset:

The Diabetes dataset comprises medical details and diagnostic measurements to predict the onset of diabetes. The target variable is 'Outcome', indicating whether the patient has diabetes or not. The dataset includes several key features:

- **Pregnancies**: Number of times the patient has been pregnant.
- **Glucose**: Plasma glucose concentration over 2 hours in an oral glucose tolerance test.
- **Blood Pressure**: Diastolic blood pressure (mm Hg).
- **Skin Thickness**: Triceps skinfold thickness (mm).
- **Insulin**: 2-hour serum insulin (mu U/ml).
- **BMI**: Body mass index (weight in kg/(height in m)^2).
- **Diabetes Pedigree Function**: A function which scores likelihood of diabetes based on family history.
- **Age**: Age of the patient.

5. Credit Card Default Dataset:

This dataset contains information about credit card holders, including their demographic details, credit history, and financial behavior to predict default risk. The target variable is 'Default', indicating whether the individual defaulted on their credit card payment. The dataset includes several key features:

- **Age**: Age of the credit card holder.
- **Gender**: Gender of the credit card holder.
- **Education Level**: Education level of the credit card holder.
- **Marital Status**: Marital status of the credit card holder.
- **Payment History**: History of payments made by the credit card holder.
- **Bill Statement Amounts**: The amount on the bill statements for previous months.

Previous Payments: Amounts of previous payments made by the credit card holder

5.Methodology

Data Preprocessing

Data preprocessing is a crucial step in preparing the data for analysis and modeling. The application includes several preprocessing techniques:

Handling Missing Values: Missing values are imputed using `SimpleImputer` from `scikit-learn`. Users can choose different strategies, such as mean, median, or most frequent, to fill in missing values.

Encoding Categorical Features: Categorical features are encoded using `LabelEncoder` to convert them into numerical values suitable for machine learning models.

Scaling Numerical Features: Numerical features are scaled using different scalers (`StandardScaler`, `RobustScaler`, `MinMaxScaler`) to ensure that all features contribute equally to the model.

Exploratory Data Analysis (EDA)

EDA involves summarizing the main characteristics of the dataset, often with visual methods. The application provides the following EDA features:

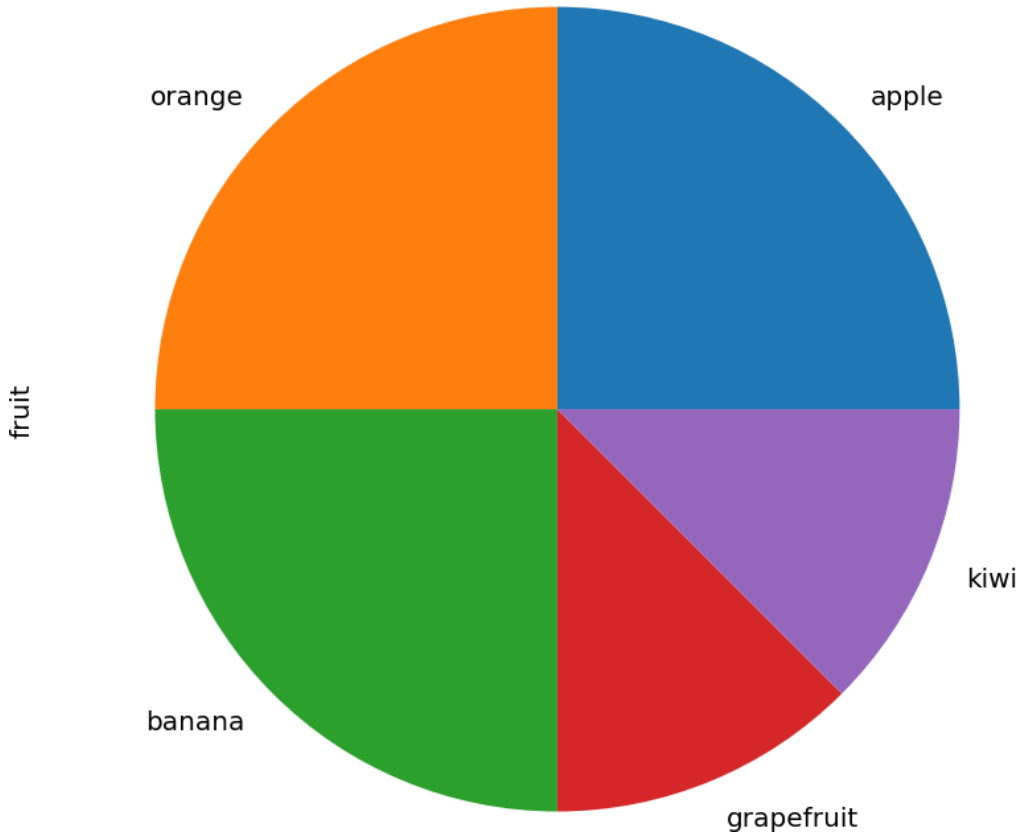
Data Preview: Displays the first few rows of the dataset to give an initial overview.

Basic Statistics: Computes and displays basic statistics, such as mean, median, standard deviation, and quartiles.

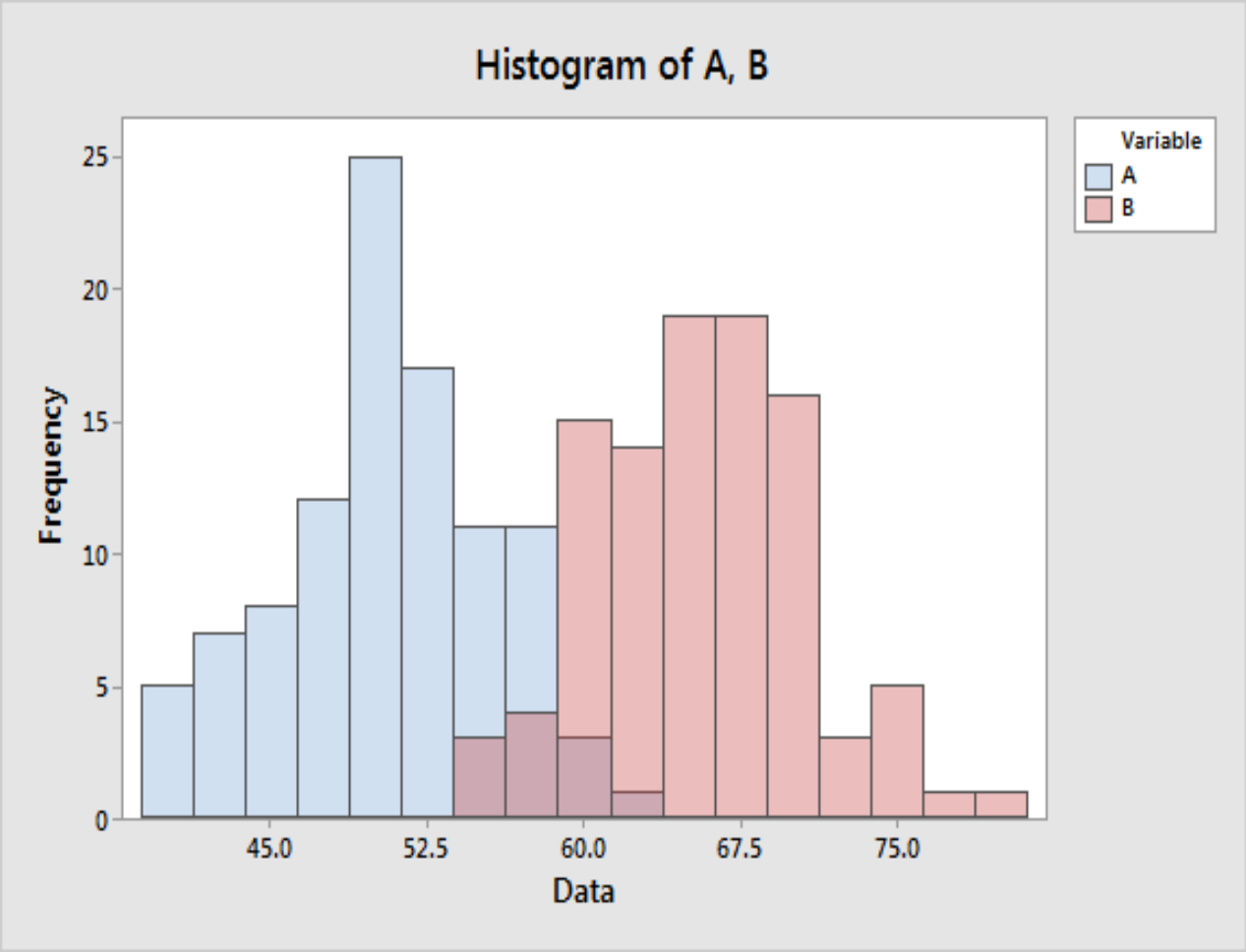
Data Types and Null Values: Identifies the data types of each column and counts the number of missing values.

Visualizations:

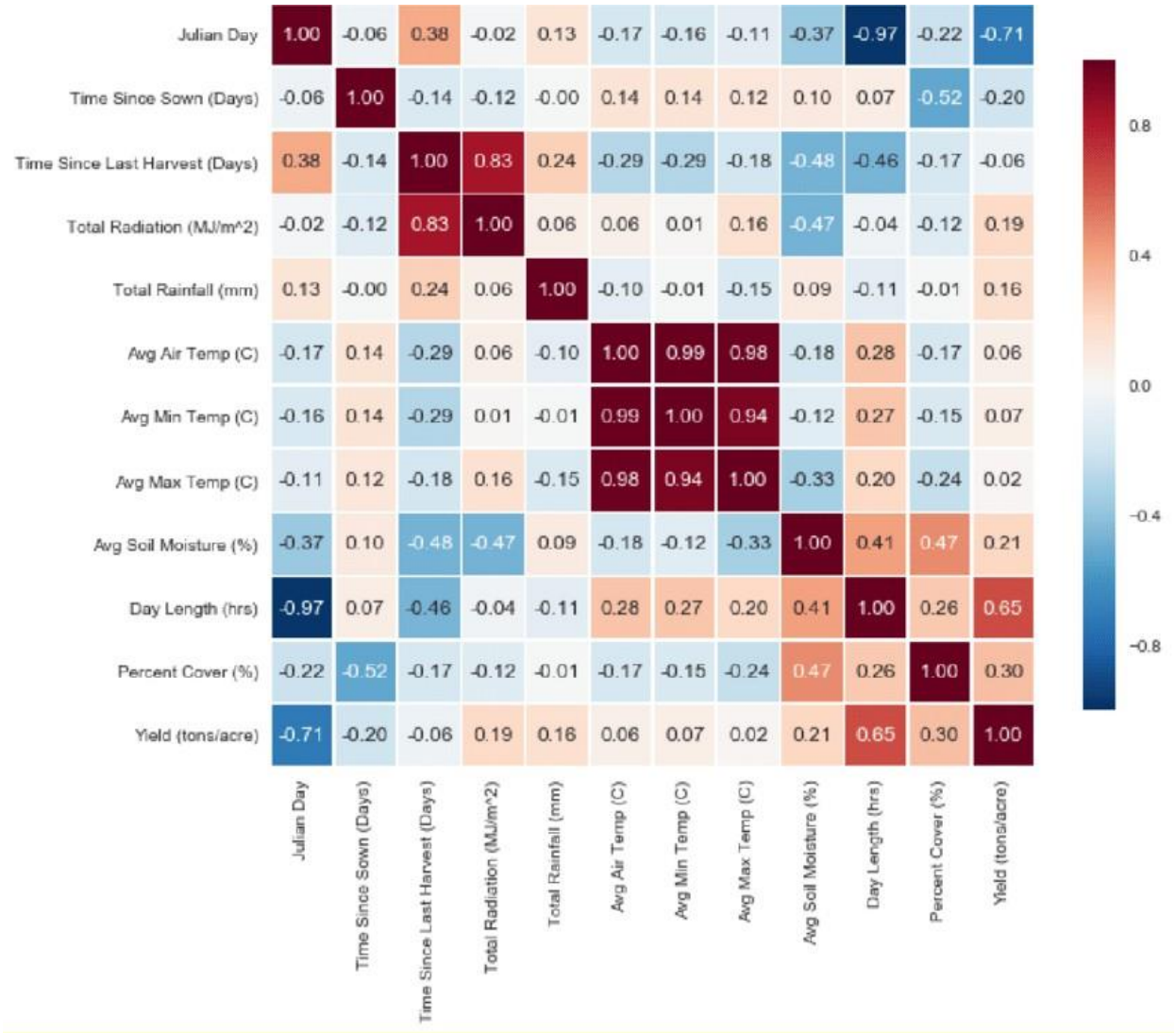
Pie Chart: Shows the distribution of categorical features.



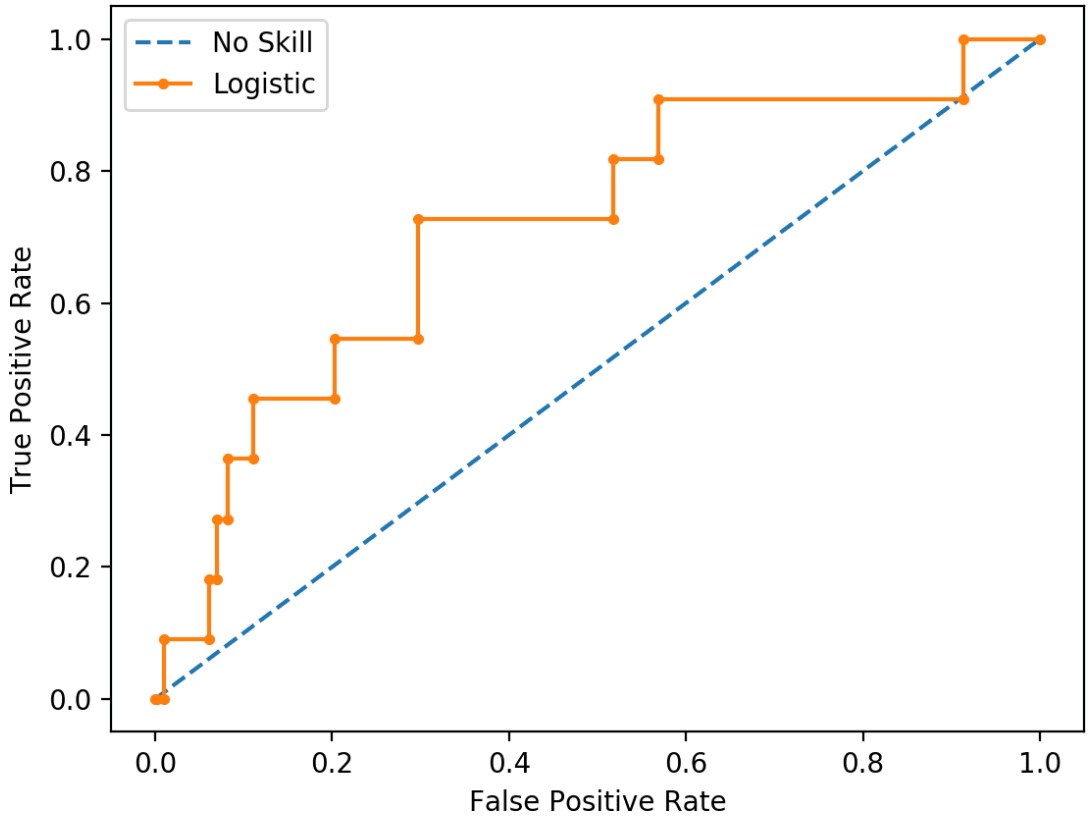
Histogram: Displays the distribution of numerical features.



Heatmap: Visualizes the correlation between numerical features.



ROC Curve for Classification Model



Modeling

The application supports two popular machine learning models:

- **XGBoost**: XGBoost, or Extreme Gradient Boosting, is an optimized gradient boosting library designed to deliver high performance and accuracy. It excels in both classification and regression tasks by implementing a more regularized model formalization, which helps in reducing overfitting. XGBoost is known for its speed and efficiency, thanks to advanced features such as parallel processing, tree pruning, and built-in cross-validation, making it a popular choice for many competitive machine learning tasks.
- **CatBoost**: CatBoost is a powerful machine learning algorithm that excels at handling categorical data automatically, eliminating the need for extensive preprocessing. Developed by Yandex, CatBoost is particularly effective in providing excellent performance with high accuracy in various applications, including classification and regression tasks. It achieves this by employing a unique method of handling categorical features and using ordered boosting, which reduces overfitting and improves the stability of the model. CatBoost's ability to work efficiently with large datasets and its ease of use make it a valuable tool in the machine learning toolkit.

Model Evaluation

The performance of the models is evaluated using several metrics:

Accuracy Score: The ratio of correctly predicted instances to the total instances.

AUC Score: The area under the ROC curve, which measures the ability of the model to distinguish between classes.

Confusion Matrix: A table that summarizes the performance of the model by showing the true positives, false positives, true negatives, and false negatives.

Accuracy Score: The ratio of correctly predicted instances to the total instances.

AUC Score: The area under the ROC curve, which measures the ability of the model to distinguish between classes.

Confusion Matrix: A table that summarizes the performance of the model by showing the true positives, false positives, true negatives, and false negatives.

Classification Report: Provides precision, recall, and F1-score for each class.

ROC Curve: A graphical representation of the true positive rate versus the false positive rate at various threshold settings.

The application provides visual representations of these metrics to help users understand model performance.

6.Implementation

Tools and Technologies

The project leverages several tools and technologies to develop the Streamlit application:

a. Streamlit: For building the interactive web application.
b. pandas: For data manipulation and analysis.
c. numpy: For numerical computations.
d. matplotlib and plotly: For data visualization.
e. scikit-learn: For machine learning and data preprocessing.
f. XGBoost and CatBoost: For implementing machine learning models.
g. imblearn: For handling imbalanced datasets.

Application Development

The application is structured into three main sections:

Homepage: Provides an introduction to the application and allows users to select a dataset. It displays the first few rows of the selected dataset and provides a brief problem description.

EDA (Exploratory Data Analysis): Allows users to upload custom datasets, view data previews, compute basic statistics, identify data types and null values, and generate visualizations such as pie charts, histograms, and heatmaps.

Modeling: Handles data preprocessing, model selection, and evaluation. Users can select between XGBoost and CatBoost models, and the application displays performance metrics including accuracy, AUC score, confusion matrix, classification report, and ROC curve.

User Interface

The application features a sidebar for navigation and a main area for displaying content. Users can upload datasets, select models, and view results interactively. The interface is designed to be intuitive and user-friendly, allowing users to perform complex data analysis tasks with minimal effort.

7.Results

EDA Findings

The EDA section provides insights into the data through visualizations and statistical summaries. Key findings include:

Data Distributions: Histograms show the distribution of numerical features, highlighting any skewness or outliers.

Correlations: The heatmap visualizes correlations between numerical features, identifying potential relationships.

Categorical Feature Distribution: Pie charts display the distribution of categorical features, helping to understand the composition of the dataset.

Missing Values: The summary of missing values helps identify potential data quality issues.

Model Performance

Model performance metrics for each dataset are displayed, including:

- **Accuracy**: This metric measures the proportion of correctly classified instances out of the total instances in the dataset. It is a fundamental measure of a model's performance, indicating how often the model makes correct predictions compared to all predictions made.

- **AUC Score**: The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) score quantifies the overall ability of the model to distinguish between positive and negative classes. A higher AUC score indicates better model performance in distinguishing between the classes.

- **Confusion Matrix**: This table summarizes the performance of a classification model by displaying the counts of true positives, false positives, true negatives, and false negatives. It provides a comprehensive view of how well the model is performing across all categories.

- **Classification Report**: This detailed report includes metrics such as precision, recall, and F1-score for each class. Precision measures the accuracy of positive predictions, recall (or sensitivity) measures the ability to identify all positive instances, and F1-score provides a harmonic mean of precision and recall.

- **ROC Curve**: The Receiver Operating Characteristic (ROC) curve visualizes the trade-off between the true positive rate (sensitivity) and the false positive rate (1-specificity) across different threshold values. This curve helps in understanding the performance of a classification model at various thresholds, aiding in selecting the optimal threshold for classification.

The results demonstrate the effectiveness of the application in automating data preprocessing, model training, and evaluation. The performance metrics provide valuable insights into the strengths and weaknesses of each model, helping users make informed decisions.

8.Discussion

Interpretation of Results

The results from the EDA and modeling sections provide valuable insights into the data and the performance of different machine learning models. Key observations include:

Data Quality:

Identifying missing values and data distributions helps assess the quality of the data and informs necessary preprocessing steps.

Feature Relationships:

Correlation heatmaps reveal relationships between features, guiding feature selection and engineering.

Model Performance:

The evaluation metrics highlight the strengths and weaknesses of each model, helping users choose the most suitable model for their dataset.

Comparison with Existing Solutions

The application is compared with existing data analysis and machine learning tools, highlighting its ease of use, flexibility, and interactive features. Key advantages include:

User-Friendly Interface:

The Streamlit application provides an intuitive interface that simplifies complex data analysis tasks.

Automation:

Automated data preprocessing and model evaluation reduce the need for manual coding and minimize errors.

Interactivity:

Interactive visualizations and real-time updates enhance the user experience and facilitate data exploration.

Overall, the application offers a streamlined and efficient solution for data analysis and machine learning, making it accessible to a broader audience.

9.Conclusion and Future Work

This project successfully developed a Streamlit web application for exploratory data analysis and machine learning modeling. The application simplifies the process of data analysis and model building, making it accessible to users with minimal coding experience. Key achievements include:

A user-friendly interface for uploading and analyzing datasets.

Automated data preprocessing pipelines for handling missing values, encoding categorical features, and scaling numerical features.

Support for multiple machine learning models with detailed performance metrics.

Interactive visualizations that provide meaningful insights into data patterns and model performance.

Future Work

Future enhancements could include:

Additional Models: Adding support for more machine learning models, including regression and clustering algorithms.

Advanced Preprocessing: Implementing advanced data preprocessing techniques, such as feature selection and dimensionality reduction.

Visualization Options: Integrating additional visualization options to provide deeper insights into data patterns and model performance.

Enhanced User Interface: Improving the user interface for better usability and a more seamless experience.

These enhancements would further increase the functionality and usability of the application, making it an even more powerful tool for data analysis and machine learning.

10.References

Streamlit Documentation:

The official Streamlit documentation serves as a comprehensive resource for learning how to utilize Streamlit in the development of web applications. It includes tutorials, detailed API references, and practical examples to facilitate a thorough understanding of the platform.

URL: streamlit.io

Streamlit for Machine Learning and Data Science" by Tyler Richards:

This book serves as an in-depth guide to leveraging Streamlit for the creation of interactive machine learning and data science applications. It covers a wide array of features and best practices essential for developing efficient and user-friendly applications.

ISBN: 978-180056

"Streamlit: The Fastest Way to Build Data Apps" by Marc Skov Madsen:

This book provides a practical introduction to Streamlit, focusing on expedited development of data applications. It encompasses a variety of topics ranging from basic usage to advanced functionalities, making it an invaluable resource for both beginners and experienced developers alike.

ISBN: 978-1839211876

Towards Data Science - Building a Machine Learning Web App with Streamlit:

- Description: This article on Towards Data Science walks through the process of creating a machine learning web application using Streamlit, covering data loading, visualization, and model training.
-
- Link: https://towardsdatascience.com/building-a-machine-learning-web-application-with-streamlit-61d0c3765e80

Analytics Vidhya - How to Build a Machine Learning Application Using Streamlit:

- Description: This tutorial from Analytics Vidhya provides step-by-step instructions on building a machine learning application using Streamlit, including code examples and explanations.
-
- Link: https://www.analyticsvidhya.com/blog/2020/12/streamlit-create-machine-learning-web-app/

Medium - Deploying Machine Learning Models with Streamlit:

- Description: This Medium article discusses deploying machine learning models using Streamlit, including practical tips and code snippets for effective deployment.
- Link: https://medium.com/swlh/deploying-machine-learning-models-with-streamlit-101d911c0b24

DataCamp - Interactive Data Apps with Streamlit:

- Description: DataCamp offers a tutorial that explores how to create interactive data applications with Streamlit, focusing on data visualization and user interface elements.
- Link: https://www.datacamp.com/tutorial/streamlit