Comprehensive Data Science Project

---

1. Introduction

This project demonstrates a complete **end-to-end Data Science and Machine Learning lifecycle**, starting from business problem definition to model deployment with a user interface. The objective was to simulate an **industry-style ML workflow**, focusing not only on model accuracy but also on data quality, preprocessing pipelines, deployment readiness, and error handling.

The project covers:

- Data collection and validation
- Exploratory Data Analysis (EDA)
- Feature engineering and model development
- Model evaluation
- Deployment preparation using Streamlit

---

2. Business Problem Definition

Problem Statement

To build a predictive machine learning system that accepts structured user input and provides accurate predictions to support business decision-making.

Business Objective

- Automate prediction tasks
- Reduce manual analysis effort
- Enable quick, data-driven decisions

Success Metrics

- Model accuracy and ROC-AUC score
- Stable and error-free prediction pipeline
- Successful deployment with a usable frontend

---

3. Data Collection

Data Source

- Dataset was either gathered from structured sources or synthetically generated to simulate real-world business data.

Data Dictionary (Example)

Feature Name	Description	Type
age	Age of individual	Numeric
income	Annual income	Numeric
category	Customer category	Categorical
score	Historical score	Numeric
target	Prediction label	Binary

Data Quality Validation

- Checked for missing values
- Verified data types
- Identified duplicates
- Ensured target variable consistency

---

4. Exploratory Data Analysis (EDA)

EDA was conducted to understand data distribution, relationships, and potential issues.

Key Steps

- Summary statistics (mean, median, std)
- Distribution plots
- Correlation analysis
- Category-wise comparisons

Insights

- Certain numerical features showed strong correlation with the target
- Some categorical variables required encoding
- Outliers were detected and handled where necessary

EDA helped guide **feature engineering and model selection**.

---

5. Feature Engineering & Preprocessing

Techniques Used

- Numerical feature scaling (StandardScaler)
- Categorical encoding (OneHotEncoder)
- Missing value handling (SimpleImputer)

Pipeline Design

A **ColumnTransformer + Pipeline** approach was used to ensure:

- Consistent preprocessing during training and inference
- Prevention of data leakage
- Production-ready workflow

This design choice later became crucial during deployment.

---

6. Model Development

Models Trained

- Logistic Regression
- Tree-based models (e.g., Random Forest)
- Other baseline classifiers (if applicable)

Training Process

- Train-test split
- Cross-validation
- Hyperparameter tuning using GridSearchCV

Final Model Selection

The best-performing model was selected based on:

- Accuracy
- Precision & Recall
- ROC-AUC score

The final model was saved using joblib.

---

7. Model Evaluation

Evaluation Metrics

- Accuracy
- Precision
- Recall
- F1-score
- ROC-AUC

Results

- Model achieved stable performance across test data
- No overfitting observed
- Consistent predictions after pipeline integration

---

8. Deployment Preparation

Backend (Prediction API Logic)

- Loaded trained pipeline model
- Accepted user input as DataFrame
- Applied preprocessing automatically
- Generated prediction and probability

Frontend (Streamlit App)

- Input fields for each feature
- Predict button
- Display of prediction result and confidence

Execution Command

streamlit run Deployment Prep.py

---

9. Challenges Faced & Solutions

1. Streamlit ScriptRunContext Warning

Cause: Running Streamlit app using python file.py
Solution: Always run using streamlit run file.py

---

2. FileNotFoundError (model.pkl)

Cause: Model file not created before deployment
Solution: Ensured model training script saves the model correctly

---

3. Feature Mismatch Error

ValueError: X has 5 features, but ColumnTransformer is expecting 13

Cause: Inference input did not match training features
Solution: Used DataFrame with exact column names and structure

---

4. TypeError: isnan not supported

Cause: Passing incorrect data types to OneHotEncoder
Solution: Ensured categorical inputs remained strings and matched training categories

---

10. Business Impact

- Demonstrates a **production-ready ML pipeline**
- Reduces manual prediction effort
- Can be extended to real business datasets
- Provides a foundation for scalable ML systems

---

11. Future Enhancements

- Connect to real-time database
- Add authentication to frontend
- Deploy using Docker / Cloud (AWS, Azure)
- Add monitoring and logging
- Improve UI/UX design

12. Conclusion

This project successfully demonstrates:

- Complete ML lifecycle understanding
- Strong debugging and problem-solving skills
- Industry-standard preprocessing pipelines
- Deployment awareness using Streamlit

It reflects **real-world data science practices** and is suitable for:

- Internship evaluation
- College final project
- GitHub portfolio
- Interview discussion