Abstract

This report summarizes a series of machine learning projects that I completed as part of my learning journey. My main objective was to gain hands-on experience while building a strong understanding of the theoretical concepts that drive machine learning models. I worked on projects involving regression, classification, natural language processing (NLP), and exploratory data analysis (EDA). Each project gave me opportunities to explore different algorithms, face challenges such as handling missing data or text preprocessing, and strengthen my skills in Python and data science libraries.

1 Introduction

When I started working on these projects, my aim was not just to complete them, but to actually understand the concepts step by step. I deliberately chose projects that increased in difficulty beginning with simple regression problems and eventually working on classification, NLP, and EDA.

Through these projects, I practiced consistent workflow: loading and exploring datasets, cleaning and preprocessing the data, training models, evaluating their performance, and finally documenting the process. This approach helped me connect theory with real implementation.

2 Tools and Technologies

For every project, I mainly worked in Python because of its strong ecosystem for machine learning. I used:

pandas, numpy for data handling and preprocessing

scikit-learn for machine learning algorithms and evaluation metrics

matplotlib, seaborn for visualization

Sweetviz and WordCloud for advanced EDA and text visualization

Jupyter Notebook and VS Code as my coding environments

3 Methodology

The steps I followed in nearly all projects were:

Data Loading – Reading datasets into pandas DataFrames.

Exploratory Data Analysis (EDA) – Understanding the structure, checking missing values, visualizing distributions.

Preprocessing – Encoding categorical variables, imputing missing data, scaling features, or text preprocessing.

Model Selection – Choosing a suitable algorithm (linear regression, logistic regression, decision trees, k-NN, Naive Bayes).

Model Training – Splitting data into training and testing sets, then fitting the model.

Model Evaluation – Using metrics such as Accuracy, Confusion Matrix, Precision, Recall, R^2, and MSE.

Reflection – Writing down what worked well, where I struggled, and how I could improve.

This structured approach helped me not only complete the projects but also develop habits for real-world ML workflows.

4 Project Work

4.1 Titanic Survival Prediction

This was my first classification project. I used the Titanic dataset to predict whether a passenger survived.

- What I did: Cleaned the dataset (handled missing ages, encoded categorical variables like Sex and Embarked), applied Logistic Regression.
- Results: Achieved about 80% accuracy. Found that women and passengers in higher classes had higher survival chances.
- What I learned: I realized how crucial preprocessing is before applying models. This project gave me my first hands-on exposure to classification and feature engineering.

4.2 Diabetes Prediction System

I worked on predicting whether a person has diabetes using health metrics.

- What I did: Used Logistic Regression and Decision Trees. Compared their performance.
- Results: Accuracy was around 75%. Glucose and BMI stood out as the most important features.
- What I learned: Accuracy alone does not always tell the whole story. I started paying attention to other metrics like precision and recall because in health-related predictions, false negatives can be critical.

4.3 Advertising Sales Prediction

This project helped me understand regression better. The task was to predict sales based on TV, radio, and newspaper advertising budgets.

- What I did: Applied Multiple Linear Regression and analyzed feature importance.
- Results: TV and Radio had a strong impact on sales, while Newspaper spend had almost no effect. R^2 was close to 0.9, which showed a strong fit.
- What I learned: I got a deeper grasp of regression lines, error metrics like MSE, and why not all features contribute equally.

4.4 Student Score Predictor

This was a very simple project, but it was important for building my foundation.

- What I did: Predicted student scores from study hours using Simple Linear Regression.
- Results: The model performed very well since the relationship was nearly linear.
- What I learned: I finally understood why the "best fit line" minimizes errors using MSE. This made regression theory much clearer to me.

4.5 Spam Classifier (SMS Dataset)

This project introduced me to Natural Language Processing. I classified SMS messages as spam or ham.

- What I did: Preprocessed the text data (stopwords removal, tokenization, TF-IDF). Applied Naive Bayes and Logistic Regression. Visualized spam vs. ham messages using WordClouds.
- Results: Naive Bayes gave me ~95% accuracy, which was impressive.
- What I learned: I saw how raw text cannot be directly fed into models and needs to be transformed into numerical vectors. I also understood why probabilistic models like Naive Bayes perform well in text classification.

4.6 Exploratory Data Analysis (EDA) Project

I practiced automated and visual EDA to better understand datasets before modeling.

- What I did: Used Pandas, Matplotlib, and Sweetviz to generate automated analysis reports.
- Results: Got detailed insights into distributions, correlations, and missing values.
- What I learned: EDA is not just a formality — it's a crucial step that guides preprocessing and model choice. Without EDA, I realized I could make wrong assumptions about data.

4.7 Iris Flower Classifier

This was a classic beginner ML project but very satisfying to complete. I classified iris flowers into three species using sepal and petal features.

What I did: Applied k-Nearest Neighbors (k-NN) and Decision Trees. Experimented with different values of k.

Results: Achieved ~95% accuracy with k-NN and ~97% with Decision Tree. Very few misclassifications occurred.

What I learned: I understood how k-NN uses distance to classify and how decision trees split data using criteria like Gini Index or Entropy. This was my first project where the model's decision-making process felt interpretable.

5 Challenges I Faced

While working on these projects, I encountered several challenges:

Handling missing and inconsistent data (Titanic, Diabetes).

Understanding which features were important and which were not (Advertising dataset).

Learning how to preprocess text data for NLP (Spam Classifier).

Realizing that evaluation requires more than accuracy (Diabetes).

At first, these challenges were frustrating, but they pushed me to research more and think critically.

6 Future Improvements

Looking forward, I want to:

Apply cross-validation instead of a simple train-test split for more reliable evaluation.

Explore ensemble methods like Random Forest and Gradient Boosting.

Deploy one of my models (perhaps the Spam Classifier or Iris Classifier) using Flask or Django to build a web application.

Document my projects in a more structured way with automated notebooks and dashboards.

7 Conclusion

Completing these projects gave me a solid foundation in machine learning. I started with simple regression tasks and gradually moved to more advanced problems involving classification, NLP, and EDA. More importantly, I did not just apply algorithms blindly I made sure to understand why each step was needed, what challenges existed, and how to evaluate results properly.

These projects have not only improved my technical skills but also taught me how to think like a data scientist: always questioning the data, validating results, and looking for ways to improve models.