Healthcare Predictive Analytics Project

Team Members:

1- Omar Khaled Ashour
2- Jana Khaled Beshir
3- Sara Aly Ahmed
4- Sara Samy El Wakeel
5- Mariam Taher
6- Mariam Mostafa

Project Overview:

Project Overview: The Healthcare Predictive Analytics project focuses on developing a predictive model to improve healthcare outcomes by providing data-driven insights. The model will be designed to help healthcare professionals with tasks such as patient risk prediction, trend identification in health metrics, and making informed decisions based on predictive analytics. The project will utilize machine learning models to forecast healthcare-related outcomes, focusing on improving patient care and resource management.

Technologies Used in the project:

1. Programming Languages
   - Python (preferred for machine learning & data science)
2. Data Processing & Manipulation
   - Pandas (data manipulation)
   - NumPy (numerical computing)
3. Machine Learning & AI Frameworks

- Scikit-learn (traditional ML models)
- TensorFlow / PyTorch (deep learning models)
- XGBoost / LightGBM (boosted tree models for predictive analytics)

4. Data Visualization

- Matplotlib & Seaborn (data visualization in Python)
- Power BI (dashboard & business intelligence tools)

7. Feature Engineering & Data Preprocessing

- Scikit-learn (for preprocessing)

Roadmap for Your Healthcare Predictive Analytics Project

Milestone 1: Data Collection, Exploration, and Preprocessing (Weeks 1-2)

Tasks:

- Collect healthcare datasets (e.g., patient records, clinical data, test results).
- Perform Exploratory Data Analysis (EDA) to identify trends, missing values, and outliers.
- Preprocess data (handling missing values, encoding categorical variables, normalization).

Milestone 2: Data Analysis & Feature Engineering (Weeks 3-4)

Tasks:

- Perform statistical analysis (correlation, hypothesis testing).

- Engineer new features (e.g., risk factors, trends in medical history).
- Visualize insights using heatmaps, scatter plots, and dashboards

Milestone 3: Model Development & Optimization (Weeks 5-7)

Tasks:

- Select machine learning models (Logistic Regression, Random Forest, Neural Networks).
- Train models and evaluate using metrics like Precision
- Optimize hyperparameters using Grid Search/Random Search.

Milestone 4: Deployment & Monitoring (Weeks 8-9)

Tasks:

- Deploy the model.
- Monitor model performance over time (detect data drift)..

Milestone 5: Final Documentation & Presentation

(Weeks 10-12)

Tasks:

- Summarize findings, methodologies, and challenges.
- Create an engaging PowerPoint presentation for stakeholders.
- Discuss future improvements (e.g., adding more patient data, refining model accuracy).

Team Roles & Responsibilities (6 Members)

1- **Project Manager –** Oversees progress, ensures deadlines are met, and manages communication.
2- **Data Engineer –** Handles data collection, cleaning, and preprocessing.
3- **Machine Learning Engineer –** Develops and fine-tunes predictive models.
4- **Backend Developer –** Builds the API for deployment.
5- **Frontend Developer –** Develops dashboards for visualizing predictions.
6- **Model Evaluation & Performance Analyst –** Tests and validates model accuracy, optimizes performance, and ensures reliable predictions.

Functional Requirements

1.1 Data Collection and Preprocessing

- The system must allow healthcare providers to **upload patient records** manually or through an API.
- The system must **check the uploaded data** to ensure important details (e.g., age, medical history, test results) are included.
- The system must handle **missing data** using methods like filling missing values (e.g., mean, median) or removing incomplete records if necessary.
- The system must **normalize numerical data** (e.g., lab results, BMI, blood pressure) to keep values consistent.
- The system must convert **categorical data** (e.g., diagnosis, gender, medications) into a format that machine learning models can understand.

1.2 Exploratory Data Analysis (EDA)

- The system must provide **summary statistics** to describe the dataset (e.g., averages, counts, distributions).
- The system must **detect and highlight unusual values** (e.g., outliers, missing data, inconsistencies).
- The system must generate **visualizations** (e.g., histograms, scatter plots, heatmaps) to help users explore trends.
- The system must provide an **interactive dashboard** to display key insights in a user-friendly format.

1.3 Predictive Model Development

- The system must allow users to **choose different machine learning models** (e.g., Logistic Regression, Random Forest, Neural Networks).
- The system must **automatically split the dataset** into training, validation, and test sets.
- The system must use **cross-validation** to improve model reliability.
- The system must calculate and display key **performance metrics** (e.g., accuracy, precision, recall, F1-score, confusion matrix).
- The system must allow **hyperparameter tuning** to optimize model performance.
- The system must support both **classification tasks** (e.g., disease risk prediction) and **regression tasks** (e.g., recovery time estimation).
- The system must analyze which **features (patient factors) impact predictions** the most.

1.4 Model Deployment

- The system must **deploy the trained model** so healthcare professionals can input patient data and receive predictions.
- The system must allow users to **export predictions** in formats like JSON, CSV, or PDF for further analysis.

1.5 Model Monitoring and MLOps

- The system must **track model performance over time** to detect changes in accuracy.
- The system must store key details such as **training date, dataset version, and evaluation results**.
- The system must **alert users** if model accuracy decreases significantly.
- The system should support **automatic retraining** when performance drops below a certain level.
- The system must generate **performance reports** summarizing how well the model works overtime.

Non-Functional Requirements

1. Performance Requirements

- The system should provide predictions **within a few seconds** for a single patient input.
- The system should be able to handle **large datasets** efficiently.
- The API should support multiple users at the same time without slowdowns.

2. Security and Compliance

- Patient data must be **protected and confidential**.
- The system should use **secure login mechanisms** (e.g., username & password).
- Access to patient data should be **restricted** based on user roles (e.g., doctors vs. admin users).
- The system should follow **basic healthcare data privacy regulations**.

3. Scalability and Availability

- The system should be designed to **handle more data and users in the future**.
- The model should be **easy to update** when new healthcare data becomes available.
- The system should be **accessible online** and have minimal downtime.

4. Usability and Accessibility

- The system should have a **simple and user-friendly interface** for healthcare professionals.
- The interface should be easy to navigate, even for users with minimal technical knowledge.
- The system should provide **clear and understandable predictions** with explanations.

5. Maintainability and Extensibility

- The system should be **well-documented** so future improvements can be made easily.

- The code should be **organized and modular**, allowing easy updates or adding new features

Git-Hub Repository:

https://github.com/omarashour04/Rowad_Projiikt