Statement of Work -Heart Disease Prediction

Brief Introduction of the Project:

The Heart Disease Prediction using Machine Learning project aims to leverage clinical data to predict the likelihood of a patient developing heart disease. This is achieved using various machine learning algorithms trained on medical datasets containing information such as age, sex, cholesterol levels, blood pressure, and other health indicators. The goal is to create a predictive model that can assist in early detection, enabling timely medical intervention. Additionally, the project includes the deployment of a user-friendly web application, built using Streamlite, to allow users to input their medical information and receive instant predictions about their heart health.

This project is significant as it addresses one of the leading causes of mortality globally, heart disease, by using advanced data-driven techniques to aid in preventive healthcare.

2. Project Scope

- Data Collection and Preprocessing: Gather data from reliable datasets (e.g., UCI Heart Disease dataset), clean the data to handle missing values, and preprocess it for model building.
- Exploratory Data Analysis (EDA): Conduct EDA to uncover trends and patterns in the data, visualizing key features affecting heart disease outcomes.
- Model Building: Train several machine learning models, such as Logistic Regression, Random Forest, and Support Vector Machines (SVM). Evaluate model performance using metrics like accuracy, precision, recall, and F1 score.
- Model Optimization: Use techniques such as hyperparameter tuning, cross-validation, and feature selection to improve model accuracy.
- Deployment: Develop a user-friendly web interface using Streamlit, enabling users to input data and receive predictions in real time.

Work Breakdown Structure (WBS)

1. Data Collection & Preprocessing

- 1.1 Identify Data Sources
    - Obtain clinical data relevant to heart disease (e.g., UCI Heart Disease Dataset).
- 1.2 Data Cleaning
    - Handle missing values, remove duplicates, and address outliers.
- 1.3 Data Transformation

    - Convert categorical features, normalize numerical features, and ensure consistency in data format.
- 1.4 Data Splitting
    - Split data into training and testing sets for model development.

2. Exploratory Data Analysis (EDA)

- 2.1 Statistical Summary
    - Generate summary statistics of the dataset (mean, median, standard deviation, etc.).
- 2.2 Data Visualization
    - Plot key graphs like histograms, correlation heatmaps, and box plots to understand the relationships between features.
- 2.3 Feature Correlation
    - Analyze correlations to identify the most impactful features.
- 2.4 Identify Outliers
    - Detect outliers that could affect model performance.

3. Feature Engineering

- 3.1 Feature Selection
    - Identify key features using techniques like Recursive Feature Elimination (RFE) or feature importance from models.
- 3.2 Feature Scaling
    - Apply normalization or standardization to ensure model performance is not impacted by feature scaling.
- 3.3 Feature Encoding
    - Convert categorical data into numeric formats (e.g., One-Hot Encoding).

4. Model Development

- 4.1 Model Selection
    - Evaluate multiple algorithms (Logistic Regression, Random Forest, Support Vector Machines, etc.).
- 4.2 Model Training
    - Train each model on the training data set.
- 4.3 Model Evaluation
    - Use evaluation metrics such as accuracy, precision, recall, F1 score, and confusion matrix.

- 4.4 Cross-Validation
    - Apply K-fold cross-validation to assess model generalization.
- 4.5 Hyperparameter Tuning
    - Optimize model parameters to achieve the best performance (using Grid Search, Random Search, etc.).

5. Model Optimization
    - 5.1 Performance Comparison
        - Compare all trained models to select the best-performing one based on accuracy and other evaluation metrics.
    - 5.2 Final Model Selection
        - Choose the model with the best performance and generalizability for deployment.

6. System Deployment
    - 6.1 Web Application Development (Streamlit)
        - Build an interface for users to input their data.
    - 6.2 Backend Integration
        - Integrate the final trained model with the web app backend for real-time predictions.
    - 6.3 Testing Web Application
        - Perform user testing and debugging to ensure smooth user experience and accurate prediction outputs.
    - 6.4 Deployment
        - Deploy the web app on a cloud platform (e.g., Heroku, AWS).

7. Testing
    - 7.1 Model Testing
        - Test the model on the test dataset to validate its performance.
    - 7.2 System Testing
        - Test the entire system (web app + model) for proper functioning.
    - 7.3 User Feedback
        - Gather user feedback and make necessary improvements.

8. Documentation & Reporting
    - 8.1 Model Documentation

    - Document the model development process, including data preparation, model evaluation, and final selection.
- 8.2 System Documentation
    - Provide system-level documentation covering deployment steps, dependencies, and functionality.
- 8.3 Final Report
    - Create a final report summarizing the project outcomes, insights, and recommendations for future improvements.

Work Distribution

- Data Scientists/Engineers: Responsible for data collection, EDA, feature engineering, and model development.
- Machine Learning Engineers: Focus on model training, optimization, and evaluation.
- Web Developers: Handle Streamlit application development, system integration, and deployment.
- Test Engineers: Conduct system and model testing, ensuring accuracy and performance.

Project Milestones: Heart Disease Prediction Using Machine Learning

1. Milestone 1: Data Collection and Preprocessing Completion
    - Task: Gather the heart disease dataset, clean the data, and perform necessary preprocessing (handling missing values, feature encoding, etc.).
    - Expected Outcome: A clean, well-structured dataset ready for exploratory analysis and model development.
2. Milestone 2: Exploratory Data Analysis (EDA) and Feature Selection
    - Task: Perform EDA to understand data distribution, feature importance, and correlations. Select relevant features for model training.
    - Expected Outcome: Insights from the data, visualizations, and a reduced feature set based on statistical analysis.
3. Milestone 3: Model Training and Comparison
    - Task: Train multiple machine learning models (Logistic Regression, Random Forest, etc.) using the selected features. Evaluate their performance using key metrics.
    - Expected Outcome: Trained models with initial performance results (accuracy, precision, recall, F1 score, etc.).

4. Milestone 4: Model Optimization and Selection
    - Task: Optimize the models using hyperparameter tuning, cross-validation, and feature engineering techniques to improve accuracy.
    - Expected Outcome: A final optimized model that outperforms others, ready for deployment.
5. Milestone 5: Web Application Development (Streamlit)
    - Task: Develop the user interface with Streamlit, allowing users to input clinical data and receive heart disease predictions in real time.
    - Expected Outcome: A functional web application that integrates with the machine learning model for real-time predictions.
6. Milestone 6: System Testing and Final Deployment
    - Task: Conduct system-level testing to ensure the model works accurately in the deployed web application. Perform user acceptance testing and make any necessary fixes or adjustments.
    - Expected Outcome: A fully deployed system available for use, providing accurate heart disease predictions via a web interface.
7. Milestone 7: Project Documentation and Reporting
    - Task: Prepare the final project documentation and report, summarizing the data preparation, model building process, evaluation metrics, system design, and deployment strategy.
    - Expected Outcome: A complete report covering the entire lifecycle of the project, ready for submission or presentation.

1. Data Collection and Preprocessing

- **Deliverable**: A cleaned and pre-processed dataset ready for modelling.
- **Tasks**:
    - Collect data from relevant sources (e.g., UCI Heart Disease Dataset).
    - Clean data by handling missing values, outliers, and duplicates.
    - Preprocess data through scaling, encoding categorical features, and splitting the dataset.

2. Exploratory Data Analysis (EDA)

- **Deliverable**: Comprehensive EDA report with key insights and visualizations.

- **Tasks**:
    - Provide a statistical summary of the dataset.
    - Visualize data distributions and patterns (e.g., histograms, heatmaps).
    - Analyse feature correlations and identify impactful features.

3. Feature Selection and Engineering

- **Deliverable**: Final set of selected features optimized for model training.
- **Tasks**:
    - Engineer new features if applicable (e.g., derived metrics).
    - Perform feature selection using statistical methods or model-based techniques, such as Recursive Feature Elimination (RFE).

4. Machine Learning Model Development

- **Deliverable**: Trained machine learning models (e.g., Logistic Regression, Random Forest, SVM).
- **Tasks**:
    - Train various algorithms on the pre-processed data.
    - Evaluate model performance using metrics such as accuracy, precision, recall, and F1 score.
    - Optimize models through hyperparameter tuning.

5. Model Optimization and Final Selection

- **Deliverable**: Optimized model ready for deployment.
- **Tasks**:
    - Compare models and select the best-performing one.
    - Refine the final model with additional tuning and feature selection.

6. Front-End Design (User Interface)

- **Deliverable**: Web application interface using Streamlit for user input.
- **Tasks**:
    - Design a user-friendly UI to collect clinical data from users.

- Implement Streamlit for smooth user interaction and input capture.

7. Backend Development

- **Deliverable**: Backend setup to connect user inputs to the prediction model.
- **Tasks**:
  - Design the backend architecture to handle user input.
  - Create an API to connect the front end with the predictive model.

8. Backend Integration

- **Deliverable**: Fully integrated system to process user inputs and return predictions.
- **Tasks**:
  - Integrate the trained model with the backend to enable predictions.
  - Ensure seamless communication between front end, backend, and model for real-time prediction generation.

9. Database Setup (Optional)

- **Deliverable**: Database for storing user inputs or prediction outcomes.
- **Tasks**:
  - Design a database schema and create necessary tables.
  - Integrate the database with the backend for data storage and retrieval if needed.

10. System Testing and Validation

- **Deliverable**: System validation report with test results.
- **Tasks**:
  - Validate model predictions using test data.
  - Conduct thorough system testing to ensure UI responsiveness and accurate predictions.
  - Address bugs and optimize based on testing feedback.

11. Deployment

- **Deliverable**: Deployed web application accessible for real-time heart disease predictions.
- **Tasks**:
    - Deploy the application on a cloud platform (e.g., Heroku, AWS).
    - Ensure scalability and stability in a live environment.

12. Final Documentation and Reporting

- **Deliverable**: Comprehensive project report and user guide.
- **Tasks**:
    - Document the entire development process, from data collection to deployment.
    - Provide a user guide for the web application.
    - Summarize project findings and workflows in the final report.