Ashton Wilson 008876453

Heart Attack Prediction Deliverable Executive Summary

- ## Opportunity Improvement

  Right now, our healthcare providers do not have a quick, repeatable, data-driven way to assess someone's risk of heart attack. This Heart attack prediction deliverable changes that. This predictive tool uses previous patients' outcomes and their vitals to train a machine model, which can then be fed the current patient's vitals, outputting a prediction of risk or no risk. This model is shown to be 92% accurate.

- ## Customer Needs

  Not only can the product be immediately put to use in our own hospital system, but any other healthcare entity capable of taking the required vitals could also benefit from this product. Every healthcare provider wants to provide timely, accurate data to their patients. This product has the potential to improve patient outcomes through early detection of heart attack risk.

- ## Existing Gaps in the Current Protocol

  Our healthcare facility currently relies on the individual provider's professional opinion and experience to assess heart attack risk. While that does work to an extent, it creates an uneven experience, and it is vulnerable to things like the individual providers' level of fatigue, focus, and any other factors that may bias their judgment. Our product does not aim to replace this but enhance it. Their intuition can still be applied, but now they will have a data-driven recommendation at their fingertips to bolster their assessment. To reiterate, the prediction is tailored to each individual patient based on the current vitals that have just been taken.

- ## Required Data

  Fortunately, the required data are vitals already being collected and charted. These include age, sex, cholesterol, blood pressure, chest pain type, if any, all standard things we are already asking. No additional work is needed beyond putting those values into the prediction model to get a risk assessment.

- ## Methodology for Support, Design, and Development

  The K Nearest Neighbors algorithm has been selected for this project based on its effectiveness and ability to predict based on similarity between patients' vitals. Since we are comparing humans to each other, and they should have high similarity on many data points, it seemed an ideal pick. Additionally, evaluation of its accuracy will include a confusion matrix, heat map, and a ROC-AUC graph to demonstrate its effectiveness.

- ## Deliverables

  1. A fully functional machine learning model that can offer a prediction on whether a patient is at risk of a heart attack.
  2. Documentation, including a user guide
  3. An interactive interface that healthcare professionals can use to interpret patient vitals and obtain a risk assessment.

4. Data visualizations, such as a confusion matrix, demonstrating the model's effectiveness.

- ## Product Implementation Plan and Outcomes

  Take a data set, verify there are no missing values, and clean the set if necessary. Train a machine learning algorithm to predict heart attack risk on a portion of the data and hold back another slice for testing. Build a confusion matrix and ROC curve to verify its effectiveness and examine a heat map of the data to verify a correlation between various health markers. Then, package this in a way that makes it easy for healthcare professionals to input vitals as they are taking them from the patient to get a risk prediction. This will give every healthcare professional an equal and accurate tool to help assess risk, bolstering their ability to provide consistent, excellent care.

- ## Methods for Validation and Verification of Data to Ensure Met Needs

  In the data training phase, we will construct a confusion matrix and an ROC curve to demonstrate proof of accuracy. Then, as we roll this out to healthcare providers, we will ask for feedback on the model prediction and ease of use, allowing us to refine the model further as necessary. Additionally, as more vitals are taken, our training set will grow, allowing us to refine the model further.

- ## Programming Environments and Related Costs

  1. This project will be built in Python, which is an open-source language, so no cost is associated with that; libraries such as SKlearn, Pandas, Matplotlib, and Tkinter will be used.
  2. Our department's current hardware is sufficient for this project, and no new items need to be purchased.
  3. IDEs used will be Pycharm and Jupyter Notebook will be used in the development phase, which are already installed, so there will be no additional cost here either.
  4. Our data scientist will clean the data, and our software engineers will build out the app for data entry by the healthcare staff. As this project is rolled out, we will open a channel for feedback and work out anything that does not meet our customers' needs.

- ## Projected Timeline

  - **Milestone 1: Data Collection (1 Week)**
    Tasks: Data collection, cleaning, and preparation.
    Dependencies: Data must be available.
    Resource: Data Scientist.
    Dates: 9/30 – 10/4.
  - **Milestone 2: Model Development and Initial Testing (1 Week)**
    Tasks: Develop and test the KNN model (with possible consideration of other algorithms); print accuracy scores and create data visualizations.
    Dependencies: Clean data must be available.
    Resource: Data Scientist.
    Dates: 10/7 – 10/11.
  - **Milestone 3: GUI Design (3 Weeks)**
    Tasks: Design the GUI, test for bugs, and implement full functionality.

Dependencies: A trained model that can accept input and make predictions.
Resources: Software Engineers, App Testers, QA Team.
Dates: 10/14 – 11/1.

- **Milestone 4: Product Validation (3 Weeks)**
  Tasks: Validate the product through use and feedback from internal medical practitioners.
  Dependencies: A working application, training documentation for healthcare staff, and tech support for setup if necessary.
  Resources: Help Desk, Healthcare Workers.
  Dates: 11/4 – 11/22.
- **Milestone 5: Final Launch (2 Weeks)**
  Tasks: Incorporate feedback from the validation phase, finalize the product, and roll out the final version.
  Dependencies: Feedback from initial testing.
  Resources: Software Engineers.
  Dates: 11/25 – 12/6.