**Project Scope Document:

Credit Card Qualification App**

Kyle Davidson GitHub | Email | LinkedIn

1. Project Overview:

The goal of this project is to build a simple application that predicts whether a user would qualify for a credit card based on their answers to a few specific questions. The application will utilize a machine learning model trained on a dataset sourced from Kaggle. The project will involve developing a Python script to handle all the data processing, model training, and exporting the trained model for use in the application.

2. Objectives:

- Develop a user-friendly application interface where users can input their responses a few simple questions.
- Utilize a pre-existing dataset from Kaggle to train a machine learning model.
- Create a Python script that imports the dataset, preprocesses the data, trains a machine learning model, and exports the trained model in a reusable format.
- Implement a decision-making logic in the application that utilizes the trained model to provide a binary ("YES" or "NO") response based on user inputs.
- Document project accordingly, with a focus on transparency and workflow

3. Deliverables:

- ****User Interface (UI):****
  - Simple UI allowing users to input their answers to four predetermined questions.
  - Clear display of the result ("YES" or "NO") based on the prediction.

- ****Python Script (ML Pipeline):****
  - Single Python script (`credit_card_ml.py`) that performs the following tasks:
    - Imports the credit card dataset.
    - Preprocesses and cleans the dataset as needed.
    - Performs EDA and Feature Selection
    - Trains a machine learning model using the processed data.
    - Exports the trained model in a format suitable for later use (e.g., Pickle, Joblib).

- **Documentation:**
  - README.md file on GitHub repository detailing:
    - Project Scope - overview and objectives
    - Setup instructions for running the application and the Python script.
    - Description of the dataset used and any preprocessing steps.
    - Explanation of the machine learning model and its usage in the application.

  - Credit_app.ipynb Jupyter Notebook on GitHub repository detailing:
    - Project Scope - overview and objectives
    - Pipeline in a logical and concise manner.
    - Explanation of the machine learning model and its usage in the application.

4. Constraints and Considerations:

- **Dependencies:**
  - Minimize external dependencies to keep the project lightweight.
  - Utilize essential libraries such as pandas, scikit-learn for data processing and model training.

- **Hosting:**
  - Host the project repository on GitHub for version control and collaboration.

5. Timeline:

- Estimated project completion: 2 weeks (subject to change based on complexity and unforeseen challenges).

6. Future Enhancements (Optional):

- Explore deploying the application on a web platform for wider accessibility.
- Implement additional features such as user authentication or enhanced model performance metrics.

1. **Project Scope Document**:
   - This document outlines the goals, objectives, and deliverables of your project.
   - Include a description of what you aim to achieve with your UI/application.
   - Define the features and functionalities you plan to implement.
   - Specify any constraints or limitations (e.g., time, resources, technical constraints).

2. **Technical Design Document**:
   - Describe the architecture of your application.
   - Outline the technologies, tools, and frameworks you plan to use (minimal dependencies).
   - Detail the UI design and user experience considerations.
   - Define the data model and any APIs or services your application will interact with.

3. **Implementation Plan**:
   - Break down the project into manageable tasks or milestones.
   - Define timelines and deadlines for each task.
   - Allocate resources and responsibilities (if working in a team).
   - Identify potential risks and mitigation strategies.

4. **User Documentation**:
   - Create user guides or manuals for using your application.
   - Include installation instructions, if applicable.
   - Provide information on how to navigate and utilize the features of the application.

5. **Testing and Quality Assurance Document**:
   - Outline your testing strategy (e.g., unit testing, integration testing).
   - Detail test cases and scenarios to validate the functionality of your application.
   - Document any bugs or issues encountered during testing and their resolutions.

6. **Deployment Guide**:
   - Explain how to deploy your application.
   - Include instructions for setting up the development environment.
   - Describe how to configure and run the application locally or on a free hosting platform.

7. **Project Report or Summary**:
   - Compile a summary of your project, including its objectives, methodology, and outcomes.
   - Reflect on challenges faced and lessons learned.
   - Share any future plans or improvements for the project.