The project's domain background is the prediction of the success of Starbucks offers. It aims to develop a system that can predict whether an offer will be successful based on transaction, demographic, and offer data. The project utilizes the dataset from the Starbucks app to analyze customer behavior and determine the factors that contribute to the success of offers.

The problem statement is to build a predictive model that can determine the success of Starbucks offers. By analyzing transaction data, demographic information, and offer details, the goal is to predict whether an offer will be completed or not by customers. This prediction can help Starbucks optimize their offer strategy and target specific demographic groups more effectively.

The datasets and inputs used for the problem include:

portfolio.json: Contains information about various offers such as offer type, duration, and reward.
profile.json: Contains demographic data for customers, including their age, gender, income, and membership start date.
transcript.json: Contains customer transaction and offer event data, including offer received, viewed, completed, and transaction amounts.
The proposed solution for the problem involves the following steps and highlights the usage of AWS:

Data gathering and cleaning: The code reads the data files and preprocesses them by handling missing values, renaming columns, converting data types, and performing one-hot encoding.
Exploratory Data Analysis: Analyzes the distribution of gender, age, and income in the customer profile dataset and visualizes the distribution of event types in the transcript dataset.

Building the Offer Dataset: Combines the transcript, portfolio, and profile datasets to create an offer dataset. It calculates the completion rate for each offer and filters out incomplete data and unnecessary columns.

Model Training and Evaluation: Splits the dataset into training, validation, and test sets. Builds an XGBoost classifier model and trains it on the training set. Evaluates the model's performance using accuracy, precision, recall, F1 score, and ROC AUC. Displays a confusion matrix and ROC curve.

Prediction: Makes predictions using the trained model and deploys it as an endpoint using AWS Sagemaker. The predictions can be accessed via the Lambda function or in the notebook instance.

The evaluation metrics for the solution include accuracy, precision, recall, F1 score, and ROC AUC. These metrics will be used to assess the performance of the trained XGBoost classifier model.

The project design involves utilizing AWS services for data processing, model training, and deploying the prediction endpoint. Screenshots in the readme show the usage of AWS services such as AWS Notebook Instance, S3 Bucket, Training Job, and Deployed Endpoint. The results will not be displayed in a terminal but can be accessed through a Lambda function. The code files and dataset files need to be downloaded and placed in the same directory. The code is executed by running the Python code file using the command python code_file.py. The project encourages exploration, modification, and experimentation with different models or techniques to improve the prediction of offer success.