Members

Jo-Anne van der Wath (577394)

Henry Roux (577440)

Armandre Erasmus (577311)

Chaleigh Storm (577716)

Bin381

Project Milestone 6

Table of Contents

[Introduction 2](#_Toc181219845)

[Deployment Plan 2](#_Toc181219846)

[Review of Data Mining Results 2](#_Toc181219847)

[Best Approach for Implementation 2](#_Toc181219848)

[Deployment Strategy 3](#_Toc181219849)

[Deployable Results 3](#_Toc181219850)

[Alternative Deployment Plans 3](#_Toc181219851)

[Information Distribution to Users 3](#_Toc181219852)

[Monitoring of Model 3](#_Toc181219853)

[Integration with Organizational Systems 4](#_Toc181219854)

[Deployment Tools 4](#_Toc181219855)

[Monitoring and Maintenance Plan 4](#_Toc181219856)

[Monitoring and Maintenance Strategy 4](#_Toc181219857)

[Dynamic Aspects 4](#_Toc181219858)

[Accuracy Metrics and Monitoring Protocols 4](#_Toc181219859)

[Re-evaluation and Model Discontinuation Criteria 5](#_Toc181219860)

[Update Criteria and Mechanisms 5](#_Toc181219861)

[Documenting the Initial Problem and Assessing Changes in Objectives 6](#_Toc181219862)

[User Guide for Application 7](#_Toc181219863)

[Bibliography 10](#_Toc181219864)

# Introduction

This milestone contains the sixth phase of the CRISP-DM (Cross Industry Standard Process for Data Mining) methodology, the deployment and monitoring phase. The model created during the modelling phase will need to be deployed into a working environment, where the model can be used to make predictions based on new data given through the application. The goal of this phase is to deploy the model in an environment where it can be used to the benefit of the business.

# Deployment Plan

## Review of Data Mining Results

The Random Forest model was approved for deployment in the previous stage of the CRISP-DM framework (phase 5 – Evaluation) based on its strong performance across accuracy, precision, recall, and F1 score, aligning with business objectives and data mining goals. The model successfully predicts customer eligibility using demographic and financial data and does not use the annual salary attribute during the training of the model.

The model effectively addresses the business problem by incorporating additional predictive features beyond income, improving the accuracy of eligibility predictions and potentially expanding the eligible customer base without increasing credit risk.

## Deployment Strategy

### Deployable Results

Final Model: The random forest model is the final model that will be deployed with the web application. This is the model that will be used to make eligibility predictions for the application.

Primary Output: The model provides a classification score (eligible/ineligible) for each customer.

Feature Importance Insights: The model outputs the most influential features, which can assist in future business decisions by identifying key eligibility factors.

Dashboard for Data Exploration: The deployment tool will display real-time predictions and graphical representations of trends, model evaluation metrics (accuracy, precision, recall and f1-score).

### Alternative Deployment Plans

Any software delivery team's success depends on having a well-thought-out software deployment strategy. It guarantees reliable, repeatable deployment, lowers mistakes and downtime, permits simple rollbacks, permits controlled deployment to various environments, facilitates success tracking, and incorporates contemporary techniques. Additionally, it guarantees minimal interruption, safety, and speed. Below are four alternative deployment strategies (Berclaz, 2024):

Big Bang Deployment: When using big bang deployment, software is deployed as a whole at once. There are no increments in this strategy, only a single deployment of the complete software.

Continuous Deployment: New versions of the software will be released at any given time, without the need for manual processes. This method results in the quick distribution of software.

Blue/Green Deployment: In blue/green deployment, multiple versions of the same software are running at the same time. The load balancer switches the traffic from the old version to the new version when the new version satisfies the requirements.

Shadow Deployment: In shadow deployment, the new version is deployed alongside the old version, however user access to the new version is restricted. Copies of the requests that are sent to the old version will be sent to the new version for the purpose of testing the new version.

### Information Distribution to Users

Dashboard: Users will access a dashboard where they can enter new customer data, view eligibility scores, and analyze key predictor trends.

Automated Reports: Weekly summary reports will be generated and shared with stakeholders, outlining model performance, new eligible customers, and trends in predictor values.

User Guides and Training: Onboarding and training materials will help end-users interpret results accurately, emphasizing predictor importance and potential use cases.

### Monitoring of Model

User Activity Tracking: Integrate usage tracking (e.g., Google Analytics for Shiny) to monitor user interactions, such as login frequency and session duration, to understand how the model is being used.

Performance Metrics: Monitor key model metrics like accuracy, precision, recall, and F1 score monthly (the metrics will change as the model is also updated and trained on newer data), with data aggregated into monthly performance summaries for stakeholders.

Feedback Mechanism: Allow users to submit feedback directly through the interface, which can help identify usability issues or areas needing clarification.

### Integration with Organizational Systems

Data Integration: Set up regular data feeds from “CustData2.csv” or a similar database, automating weekly data updates. Implement data validation steps to ensure input data is formatted correctly for the model.

User Access and Permissions: Configure access controls to restrict usage of the app to authorized users. Store predictions and usage logs securely within the organization’s data infrastructure.

## Deployment Tools

Organizations may effectively incorporate trained models into their production systems and make use of machine learning's advantages in practical applications by using machine learning model deployment tools, which provide strong features and capabilities to expedite the deployment process. Below are a few of the best tools for deploying ML models:

Amazon SageMaker: Amazon SageMaker is a fully managed service that combines a wide range of tools to enable machine learning (ML) for every use case at a cheap cost and with great performance. Using notebooks, debuggers, profilers, pipelines, MLOps, and other tools in a single integrated development environment (IDE), SageMaker enables you to create, train, and implement machine learning models at scale. With streamlined access control and transparency over your machine learning initiatives, SageMaker satisfies governance standards. With specially designed tools to optimize, test, retrain, and implement FMs, you may even create your own FMs, which are big models that were trained on enormous datasets. You may install hundreds of pretrained models, including publically available FMs, with a few clicks using SageMaker (Amazon Web Services Inc., 2024).

Google Vertex AI: Vertex AI is a machine learning (ML) platform that enables you to modify large language models (LLMs) for use in your AI-powered applications, as well as train and implement ML models and AI applications. Vertex AI integrates data science, ML engineering, and data engineering processes, allowing your teams to work together with a shared toolkit and scale your apps with Google Cloud's advantages (Google Cloud, 2024).

Microsoft Azure ML: Azure ML is a deployment tool which allows users to deploy their models at a faster rate, which in turn increases production rates as well as efficiency and scaling. Azure ML can support many machine learning models and frameworks. Additionally, Azure ML has tools for every stage in the machine learning lifecycle. Azure ML also supports automated machine learning which allows users to create machine learning models with simple codes. This tool easily integrates with Microsoft power platform and azure services, which support end-to-end development. A con of Azure ML is that it may be difficult to use if the user is not knowledgeable about Microsoft products (TrueFoundry, 2024).

TensorFlow Serving: TensorFlow Serving is one of the most commonly used deployment tools in machine learning. It provides an adaptable framework and is built specifically for TensorFlow models. TensorFlow can support many model versions at once and allows for smooth integration with TensorFlow frameworks. Unfortunately, TensorFlow Serving does not support security features and may not be compatible with platforms outside of TensorFlow (TrueFoundry, 2024).

Shiny: The model can be deployed through the use of a shiny web application. Shiny is an R package which is used to build web applications using R programming. Shiny applications consist of a user interface and a server side where all the processing occurs. Users are able to interact with the user interface and view any results from processing (Shiny, 2024). Shiny allows for the deployment of machine learning models that are built in R. Users are able to host shiny applications on a shiny server or may deploy the model to the publishing platform, Posit Connect. Posit Connect allows the deployment and accessing of Shiny apps and dashboards, as well as markdown codes. Using this, users will be able to secure the application and implement authentication. The scaling of R processes can be done, which also results in faster loading times. Finally, performance and recourse metrics can also be measured using Posit Connect (posit, 2024).

Flask + Docker: If Shiny’s deployment limitations are encountered, Flask (Python-based) with Docker for containerization can serve as an alternative. Flask is lightweight and can easily integrate machine learning models, though it may require converting the R model into Python. Please add references

Power BI Integration: For broader visualization options, Power BI could be used as a supplemental tool to display the model’s outputs. Power BI can provide additional analytical insights but doesn’t natively support model hosting. Please add references

## Recommended deployment tool

Shiny Application: Deploying the model through a Shiny web application ensures interactivity, accessibility, and ease of use for business users. Shiny is particularly effective because it allows direct integration with R and can dynamically display predictions and insights.

Alternative Cloud Hosting Options: For scalability, consider hosting the Shiny app on a cloud platform (e.g., Shinyapps.io or an internal server) to manage secure access and potentially large data volumes.

## Final model deployment

# Monitoring and Maintenance Plan

## Monitoring and Maintenance Strategy

To ensure the model remains effective post-deployment, monitoring is crucial. This involves continuous performance evaluation to capture any shifts in data patterns, user behavior, or external conditions that may impact the model's predictive power or relevance. According to the business understanding phase (Milestone 1), the model is intended to predict service eligibility based on a mix of demographic and financial factors to improve on the annual salary-only approach used by LangaSat .

### Dynamic Aspects

Given that demographic and financial data patterns can shift over time, it's essential to account for various dynamic factors such as changes in economic conditions, shifts in the distribution of salaries or demographics, or modifications in eligibility criteria. Monitoring tools should include data drift detection, which helps to identify if the input data significantly deviates from the original training data used to build the model. For instance, a shift in the economic climate impacting average household incomes could necessitate a recalibration of model parameters.

### Accuracy Metrics and Monitoring Protocols

In Project Milestone 3, accuracy metrics like precision, recall, F1 score, and confusion matrices were used to evaluate initial models . These metrics should continue to be tracked post-deployment to identify declines in performance.

Specifically, the following should be monitored:

Accuracy: to measure the overall performance of the model.

Precision and Recall: to assess the model's reliability in classifying eligible vs. non-eligible customers, particularly important for minimizing credit risk for LangaSat.

F1 Score**:** to balance precision and recall in cases where eligibility prediction impacts are critical.

Any decline in these metrics beyond acceptable thresholds (e.g., a 5% drop in F1 Score or accuracy) should trigger a model review or recalibration. Thresholds should be defined based on the initial model performance benchmarks from Milestone 3 evaluations .

### Re-evaluation and Model Discontinuation Criteria

The model should be re-evaluated under conditions such as:

Data Drift: A significant shift in the distribution of variables (e.g., annual salaries, household sizes) can indicate that the model's assumptions may no longer hold true.

Model Performance Degradation: A consistent decline in accuracy or F1 score below a specified threshold suggests the model may be underfitting or overfitting due to evolving data.

Business Objective Shifts: If LangaSat changes its eligibility criteria or credit risk assessment standards, the model should be revisited to ensure alignment with the updated business objectives.

The conditions for discontinuation may include persistent underperformance despite adjustments or changes in the business environment that make the model irrelevant.

### Update Criteria and Mechanisms

Model updates should occur when:

Accuracy thresholds are not met over a set period (e.g., quarterly reviews indicating consistent decline).

Model Validity Expiration: The model's training dataset may lose relevance over time due to environmental changes, requiring re-training on recent data.

Shift in Application Domain**:** As new variables (e.g., economic indicators) become relevant or are better predictive of eligibility, they should be included in re-training .

Regular re-training cycles, such as annual or semi-annual updates, should be established based on analysis of data drift trends.

### Documenting the Initial Problem and Assessing Changes in Objectives

The model's purpose was initially defined in Milestone 1 as a classifier to improve the eligibility determination by adding demographic and lifestyle factors beyond annual salary. This context is essential for evaluating any future changes in objectives; any adjustments should enhance the model's alignment with the broader goal of reducing credit risk while expanding the pool of eligible customers for LangaSat.

# User Guide for Application

The web application will consist of three pages. The main page will be the page where eligibility predictions occur.

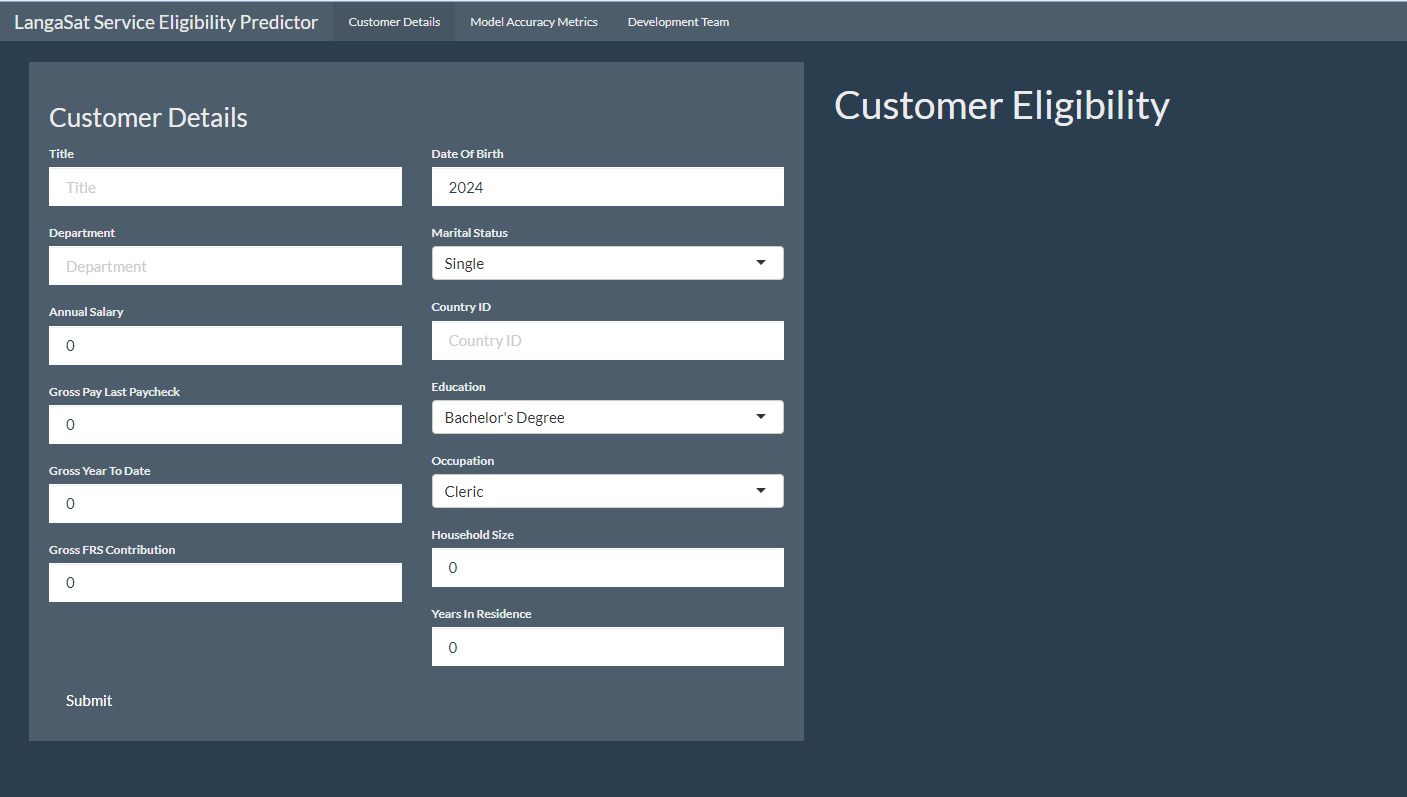


Figure 1 Web Application Predictor

The main page, called Customer Details, will take the customer details as input. The user will need to enter the values for all the empty cells. After they have entered all the details, they will be able to submit them. After they have submitted, the model will be used to make a prediction, and the classification of the user will be displayed on the right-hand side of the web application.

An example of a sample record, and its prediction can be seen in the following figure:

A screenshot of a computer

Description automatically generated

Figure 2 Customer Eligibility Prediction

The second page of the web application will contain the evaluation metrics of the model. Metrics such as the accuracy, precision, recall and f1 score will be displayed. Additionally, the percentage of eligible customers for the baseline model as well as the random forest model will be available. The increase in the number of eligible customers will be calculated and displayed as well. Furthermore, a feature importance plot will be shown on this page to allow the user to view the most relevant feature at a given time. This will allow for informed decisions to be made regarding the features that affect eligibility.

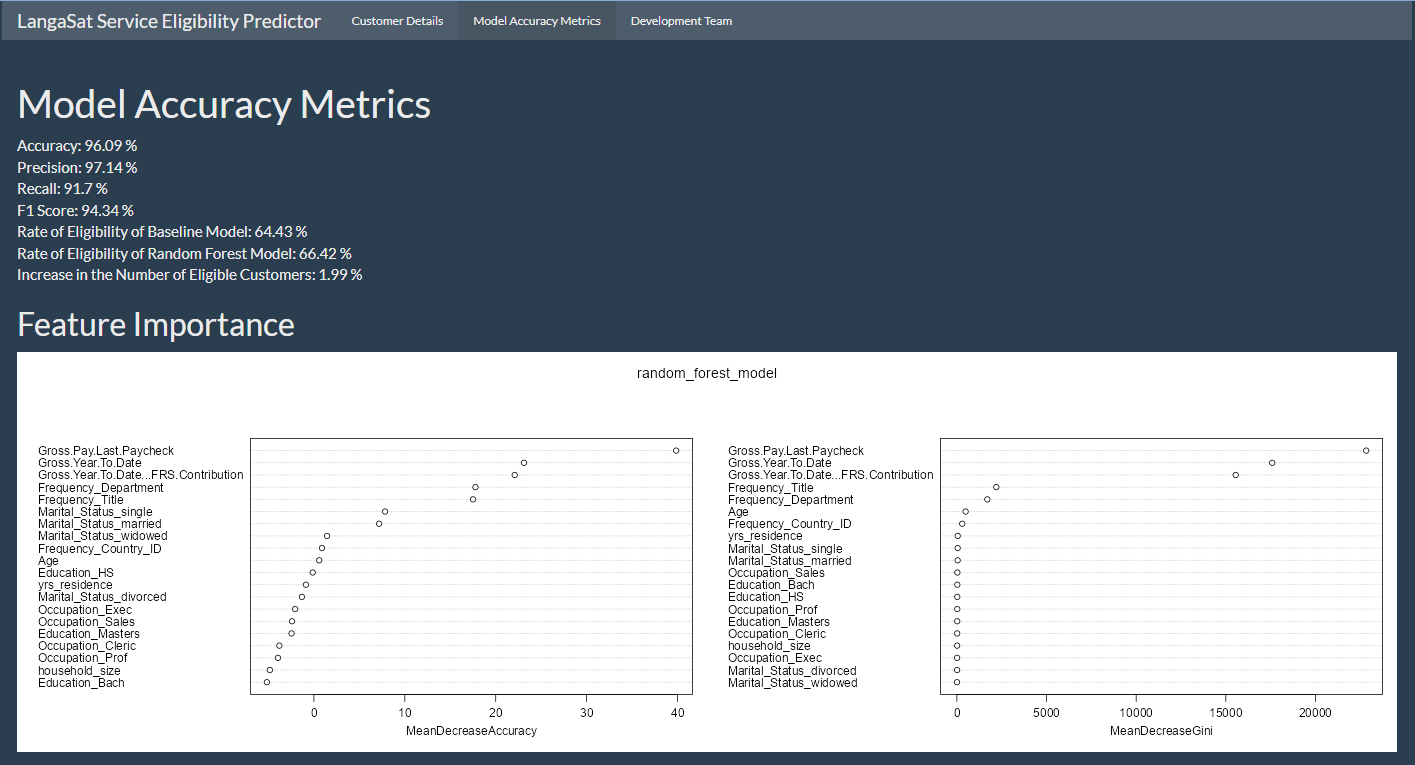


Figure 3 Web Application Model Accuracy Metrics

As an added bonus, the names of the members of the development team will be displayed on the third page, called “Development Team”.

A screenshot of a computer

Description automatically generated

Figure 4 Web Application Development Team

The interface of the web application is simple and easy to use, and any user should easily be able to navigate their way around the application.

# Bibliography

Amazon Web Services Inc., 2024. *Amazon SageMaker.* [Online]   
Available at: https://aws.amazon.com/sagemaker/  
[Accessed 30 October 2024].

Berclaz, D., 2024. *8 Deployment Strategies Explained and Compared.* [Online]   
Available at: https://www.apwide.com/8-deployment-strategies-explained-and-compared/  
[Accessed 30 October 2024].

Google Cloud, 2024. *Introduction to Vertex AI.* [Online]   
Available at: https://cloud.google.com/vertex-ai/docs/start/introduction-unified-platform  
[Accessed 30 October 2024].

posit, 2024. *Get your Shiny apps online.* [Online]   
Available at: https://posit.co/products/open-source/shiny-server/  
[Accessed 31 October 2024].

Shiny, 2024. *Welcome to Shiny.* [Online]   
Available at: https://shiny.posit.co/r/getstarted/shiny-basics/lesson1/index.html  
[Accessed 30 10 2024].

TrueFoundry, 2024. *Best Machine Learning Model Deployment Tools in 2024.* [Online]   
Available at: https://www.truefoundry.com/blog/model-deployment-tools#tensorflow-extended-tfx-serving-tailored-for-tensorflow-models  
[Accessed 30 October 2024].