BIN381

Project Milestone 6

Members

Jo-Anne van der Wath (577394)
Henry Roux (577440)
Armandre Erasmus (577311)
Chaleigh Storm (577716)

Table of Contents

Table of Figures 2
Introduction 3
Running Codes 3
Deployment Plan 3
Review of Data Mining Results 3
Deployment Strategy 3
Deployable Results 3
Alternative Deployment Plans 4
Information Distribution to Users 4
Monitoring of Model 5
Integration with Organizational Systems 5
Deployment Tools 5
Recommended deployment tool 7
Monitoring and Maintenance Plan 8
Monitoring and Maintenance Strategy 8
Dynamic Aspects and Data Drift Detection 8
Accuracy Metrics and Monitoring Protocols 9
Re-evaluation and Model Discontinuation Criteria 9
Update Criteria and Mechanisms 9
Documenting the Initial Problem and Assessing Changes in Objectives 10
User Guide for Application 11
References 14

Table of Figures

Figure 1 Web Application Predictor 11
Figure 2 Customer Eligibility Prediction 12
Figure 3 Web Application Model Accuracy Metrics 13
Figure 4 Web Application Development Team 13

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.

Running Codes

For the code to work correctly, please run the script files in the following order:

1. #1 Data cleaning, preprocessing, pipeline and model creation
2. #2 Making predictions with model and pipeline
3. app.R

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 (jumpingrivers.com, 2020).

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 (Bichiashvili, et al., 2024).

Recommended deployment tool

After analyzing multiple development tools, it was determined that the best development tool for this project will be a Shiny application. The Shiny application will be chosen for the following reasons (Shiny, 2024):

Interactivity:

Shiny offers an interactive interface, allowing users to directly interact with the model and interface. Users will be able to add input, and predictions will be displayed. Model outputs will be monitored, and the model metrics can be evaluated.

Integration with R:

As the data processing and modelling is done in R, Shiny is ideal as it is also implemented is R. This makes for seamless integration between the model and the application. The compatibility between the model and the application allows the formatting and outputs to easily be shared between the two.

Real-Time Updates:

Shiny allows for updates to be made in real-time. This is ideal for entering new records (perspective customers) to be entered into the prediction model. The prediction will be made in real-time, and the output will be stored for later use.

Visualizations:

R is also ideal, as it has visualization capabilities. This is ideal as it will allow for plots such as the importance plots to be displayed, which will give the business insights into which features are relevant to the eligibility of customers.

Collaborative Features of Shiny:

Shiny is collaborative which allows the team members to easily work together on the features. It also makes it easier to deploy the app and allow users to provide feedback.

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

Monitoring and Maintenance Plan

Monitoring and Maintenance Strategy

Post-deployment, continuous monitoring is necessary to maintain model relevance, particularly for predictive models that rely on dynamic demographic and financial data. This monitoring involves the systematic evaluation of performance metrics and regular checks for data drift to capture changes in data patterns, user behavior, or economic conditions. In the business understanding phase (Milestone 1), the model was designed to improve service eligibility prediction by incorporating a broader set of demographic and financial variables, supplementing LangaSat's traditional salary-only model (Milestone 1).

Effective monitoring incorporates tools like Power BI for visualizing model performance and R for continuous metric assessment, as recommended in Milestone 2 for enhancing data quality and insight.

Dynamic Aspects and Data Drift Detection

Since demographic and economic factors can shift rapidly, it's crucial to monitor these data for potential drift, where input distributions deviate from those in the training set. Drift can occur due to changes in the economy, demographic shifts, or updates to eligibility criteria, which impact model accuracy. Data drift detection methods such as divergence metrics and Kolmogorov-Smirnov tests are highly recommended for tracking these shifts (Gama, et al., 2014; Lever, et al., 2016).

For instance, economic downturns could lower household income averages, or policy changes might alter population demographics, necessitating model recalibration. Automated drift detection tools integrated with alert systems will provide early warnings, prompting the team to assess and retrain as needed, in line with LangaSat's focus on responsible and up-to-date decision-making (Kotsiantis, 2011).

Accuracy Metrics and Monitoring Protocols

Project Milestone 3 emphasized tracking model accuracy, precision, recall, F1 score, and confusion matrices. For continuous post-deployment monitoring, these metrics help detect shifts in model performance and allow for timely interventions.

1. **Accuracy**: Evaluates overall model performance across eligibility classifications.
2. **Precision and Recall**: Important for assessing how well the model distinguishes eligible versus non-eligible customers, essential to mitigate credit risk for LangaSat.
3. **F1 Score**: Combines precision and recall, making it crucial for high-stakes predictions like service eligibility, where both false positives and false negatives can impact the business (Hosmer, et al., 2013).

Declines exceeding a 5% threshold in F1 Score or accuracy would trigger model review and recalibration based on the benchmarks established during Milestone 3. This threshold aligns with industry standards for machine learning performance monitoring (Friedman, 2001; Kleinbaum & Klein, 2010).

Re-evaluation and Model Discontinuation Criteria

The model should undergo re-evaluation if it encounters any of the following conditions:

1. **Data Drift**: Significant changes in variables, like income distributions or household size, may indicate that the model assumptions are no longer valid, warranting recalibration.
2. **Performance Degradation**: Declines in accuracy or F1 score below the set thresholds suggest that the model may need retraining with updated data.
3. **Changes in Business Objectives**: If LangaSat's eligibility criteria or credit risk assessment approach is updated, the model should be re-aligned to these new objectives (Breiman, 2001).

Should these adjustments prove ineffective in restoring performance, discontinuation and model replacement may be necessary. Persistent underperformance, even after recalibration, may signal that the model no longer serves its original purpose for LangaSat and should be revisited (Friedman, 2001; Breiman, 2001).

Update Criteria and Mechanisms

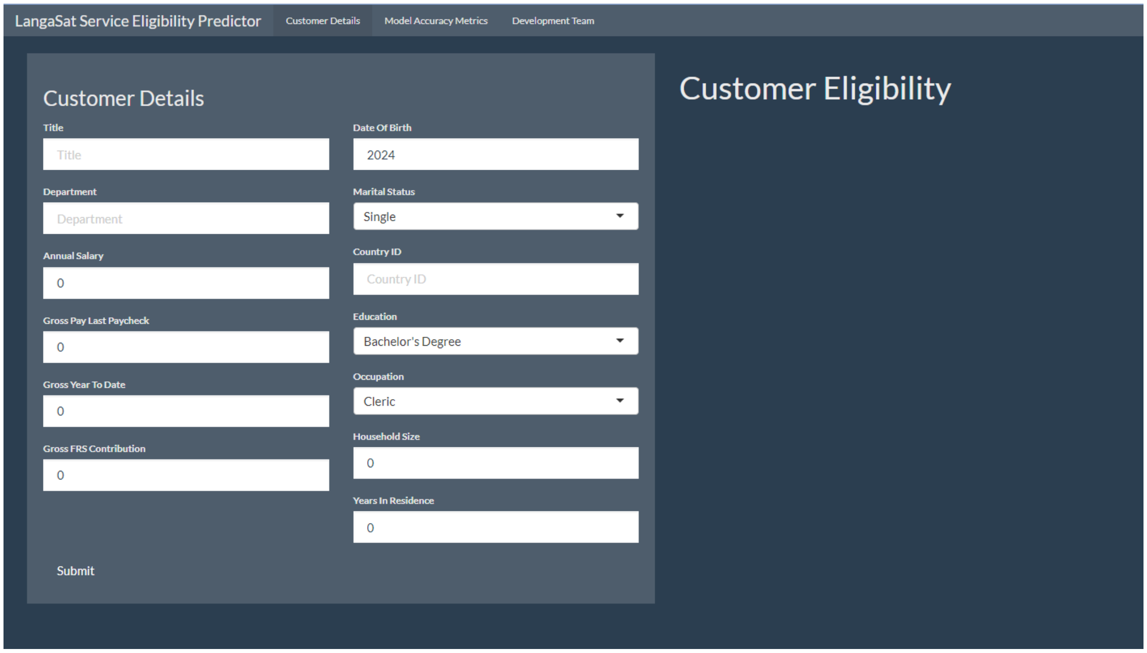
Scheduled updates are recommended if model accuracy and performance consistently fall short of established thresholds. Routine re-training, performed quarterly or semi-annually, ensures that the model reflects current data distributions and business requirements (Hosmer, et al., 2013). Additionally, if new predictors become relevant or economic indicators prove to be strong predictive factors, incorporating them into the retraining dataset will maintain the model's accuracy and alignment with LangaSat’s operational objectives. Milestone 2’s data quality emphasis supports these practices, underscoring the value of up-to-date data in predictive models (Kleinbaum & Klein, 2010; Han, et al., 2012).

Initial Problem and Changes in Objectives

The original project goal, defined in Milestone 1, was to improve LangaSat’s service eligibility assessment by incorporating factors beyond annual salary. This goal remains crucial for future evaluation and model adjustments, ensuring any updates are aligned with LangaSat’s objective of minimizing credit risk while maximizing customer eligibility. This alignment reinforces the CRISP-DM methodology followed throughout the project, providing a structured approach to adjust and evolve the model.

User Guide for Application

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



Customer Details

Title: Title

Date Of Birth: 2024

Department: Department

Marital Status: Single

Annual Salary: 0

Country ID: Country ID

Gross Pay Last Paycheck: 0

Education: Bachelor's Degree

Gross Year To Date: 0

Occupation: Cleric

Gross FRS Contribution: 0

Household Size: 0

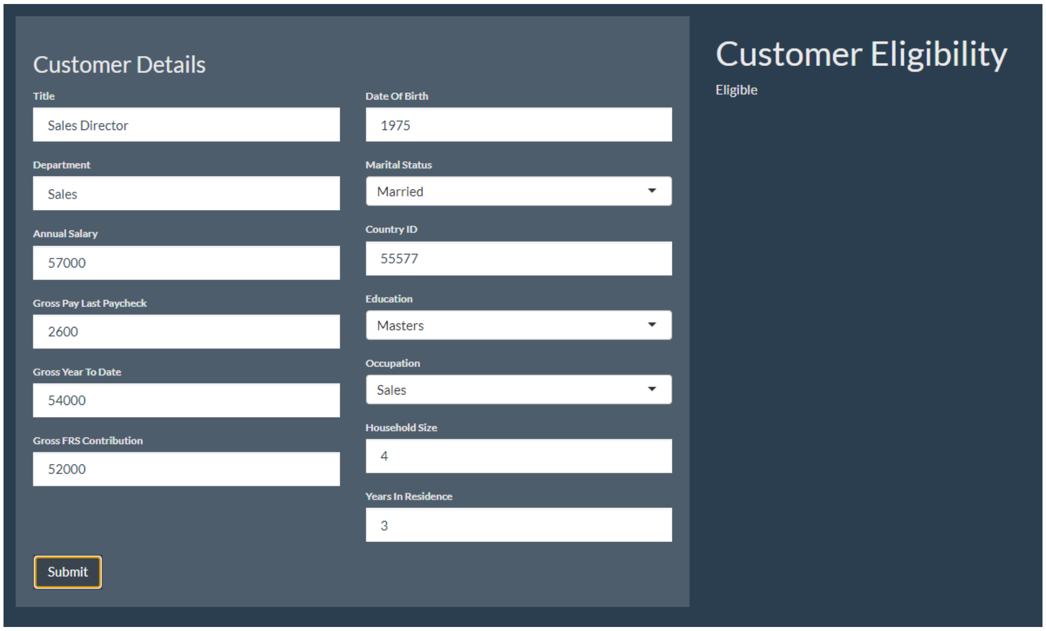
Years In Residence: 0

Customer Eligibility

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:



Customer Details

Title: Sales Director

Department: Sales

Annual Salary: 57000

Gross Pay Last Paycheck: 2600

Gross Year To Date: 54000

Gross FRS Contribution: 52000

Date Of Birth: 1975

Marital Status: Married

Country ID: 55577

Education: Masters

Occupation: Sales

Household Size: 4

Years In Residence: 3

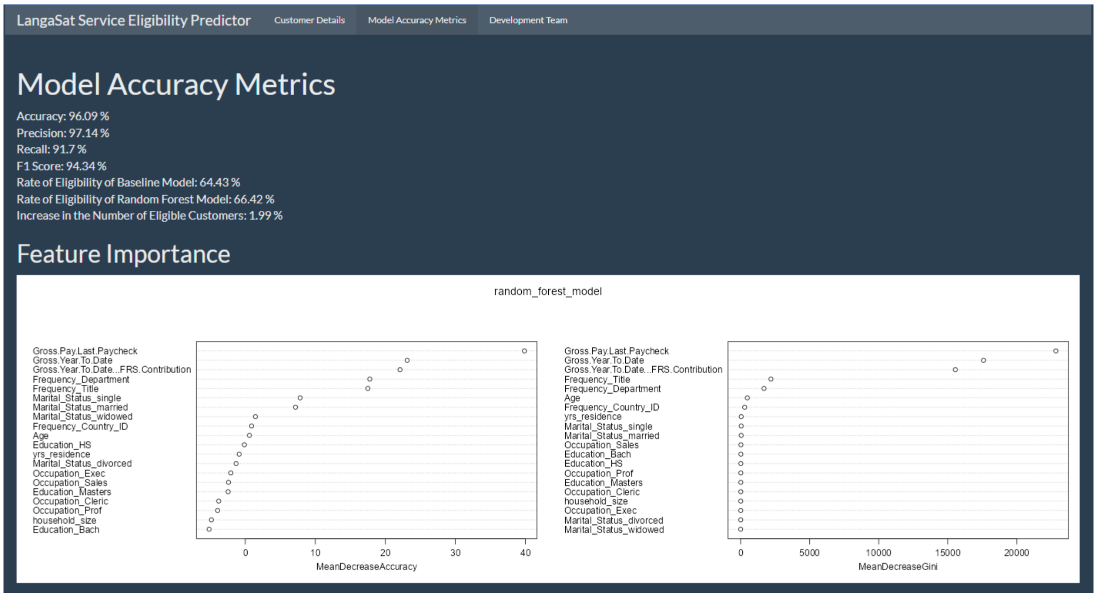
Customer Eligibility

Eligible

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.

The newly created records along with their predictions will be saved to a csv file for further analysis by the business.



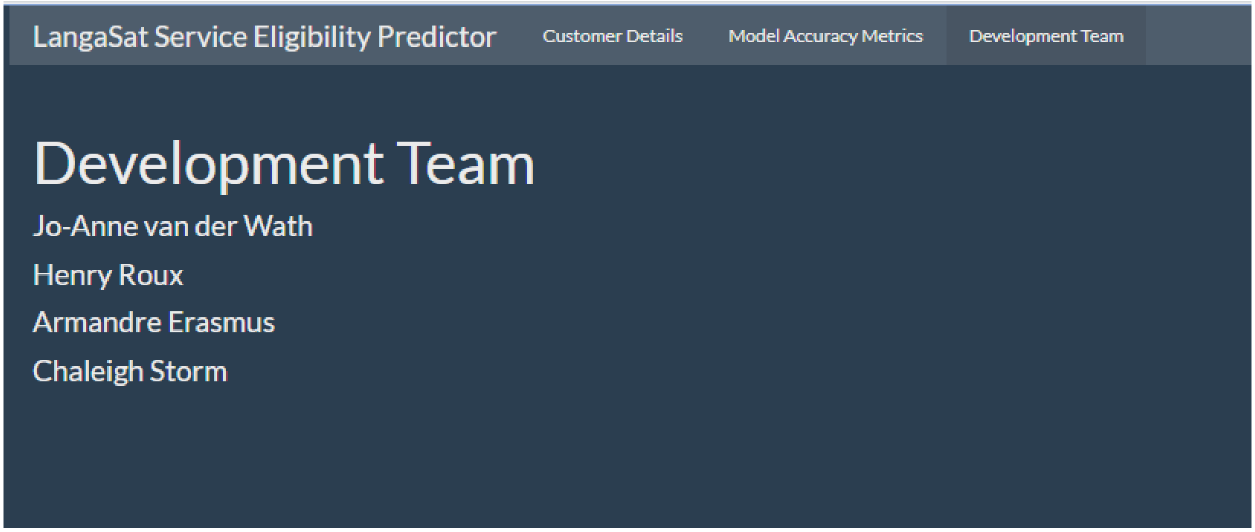
Model Accuracy Metrics

Accuracy: 96.09 %
Precision: 97.14 %
Recall: 91.7 %
F1 Score: 94.34 %
Rate of Eligibility of Baseline Model: 64.43 %
Rate of Eligibility of Random Forest Model: 66.42 %
Increase in the Number of Eligible Customers: 1.99 %

Feature Importance

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".



Development Team

Jo-Anne van der Wath

Henry Roux

Armandre Erasmus

Chaleigh Storm

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.

References

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].

Bichiashvili, O. et al., 2024. *Run Python scripts in Power BI Desktop.* [Online]
Available at: https://learn.microsoft.com/en-us/power-bi/connect-data/desktop-python-scripts
[Accessed 31 October 2024].

Breiman, L., 2001. Random Forests. *Machine Learning,* 45(1), pp. 5-32.

Friedman, J. H., 2001. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics,* 29(5), pp. 1189-1232.

Gama, J. et al., 2014. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR),* 46(4), pp. 1-37.

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].

Han, J., Kamber, M. & Pei, J., 2012. *Data Mining Concepts and Techniques.* Third Edition ed. Waltham: Elsevier.

Hosmer, D. W., Lemeshow, S. & Sturdivant, R. X., 2013. *Applied Logistic Regression.* Third Edition ed. New York, NY: John Wiley & Sons.

jumpingrivers.com, 2020. *Recreating a Shiny App with Flask.* [Online]
Available at: https://www.jumpingrivers.com/blog/r-shiny-python-flask/
[Accessed 31 October 2024].

Kleinbaum, D. G. & Klein, M., 2010. *Logistic Regression.* Third Edition ed. New York, NY: Springer New York.

Kotsiantis, S. B., 2011. RETRACTED ARTICLE: Feature selection for machine learning classification problems: a recent overview. *Springer Science+Business Media,* 42(1), pp. 157-157.

Lever, J., Krzywinski, M. & Altman, N., 2016. Points of Significance: Regularization. *Nature Methods,* 13(10), p. 803+.

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].