

MIE1628 Assignment 5 Part B Report

Mengyang Liu, ID: 1006702739

Ziyun Weng, ID: 1005351986

1. [Marks: 15] Clearly define the problem you intend to address using this dataset. Present a comprehensive problem statement that includes:
 - a. A detailed description of the meaningful issue you're tackling
 - b. An outline of all necessary steps, including:
 - i. Data preprocessing
 - ii. Data cleaning
 - iii. Modelling approach

Your problem statement should be thorough, spanning approximately half to one full page. If you determine that data cleaning is unnecessary, please provide a justification for why this dataset doesn't require cleaning. In such a case, allocate more attention to other crucial aspects such as EDA and the modelling process.

Ensure your problem statement is well-structured, coherent, and provides a clear roadmap for your data analysis project.

Phishing attacks are one of the most pervasive threats in cybersecurity today. They involve deceptive websites that impersonate legitimate domains to steal sensitive information such as passwords, credit card numbers, and login credentials. With the increasing reliance on digital services, the frequency and impact of phishing attacks have intensified. This project aims to address the critical issue of detecting phishing websites in real-time by developing a robust machine learning model capable of accurately distinguishing between phishing and legitimate websites based on behavioural and structural web features.

The dataset used in this project originates from the UCI Machine Learning Repository in **.arff** format and includes numerous categorical features that describe various characteristics of websites, such as use of pop-ups, URL structure, and security certificates. To simplify classification and address class imbalance, the original target variable, which included three classes: legitimate (1), suspicious (0), and phishing (-1), was converted into a binary format. The modified target variable now consists of two classes: phishing (1) and not phishing (0), where suspicious entries were merged with the legitimate class. This adjustment enhances the clarity of the classification task and ensures a more balanced label distribution for model training.

The key steps in our analytical pipeline include: Data Preprocessing: Loading the .arff file and converting it to a structured DataFrame. The categorical labels were encoded into numerical representations, and the class distribution was visualized and adjusted to binary. Data Cleaning:

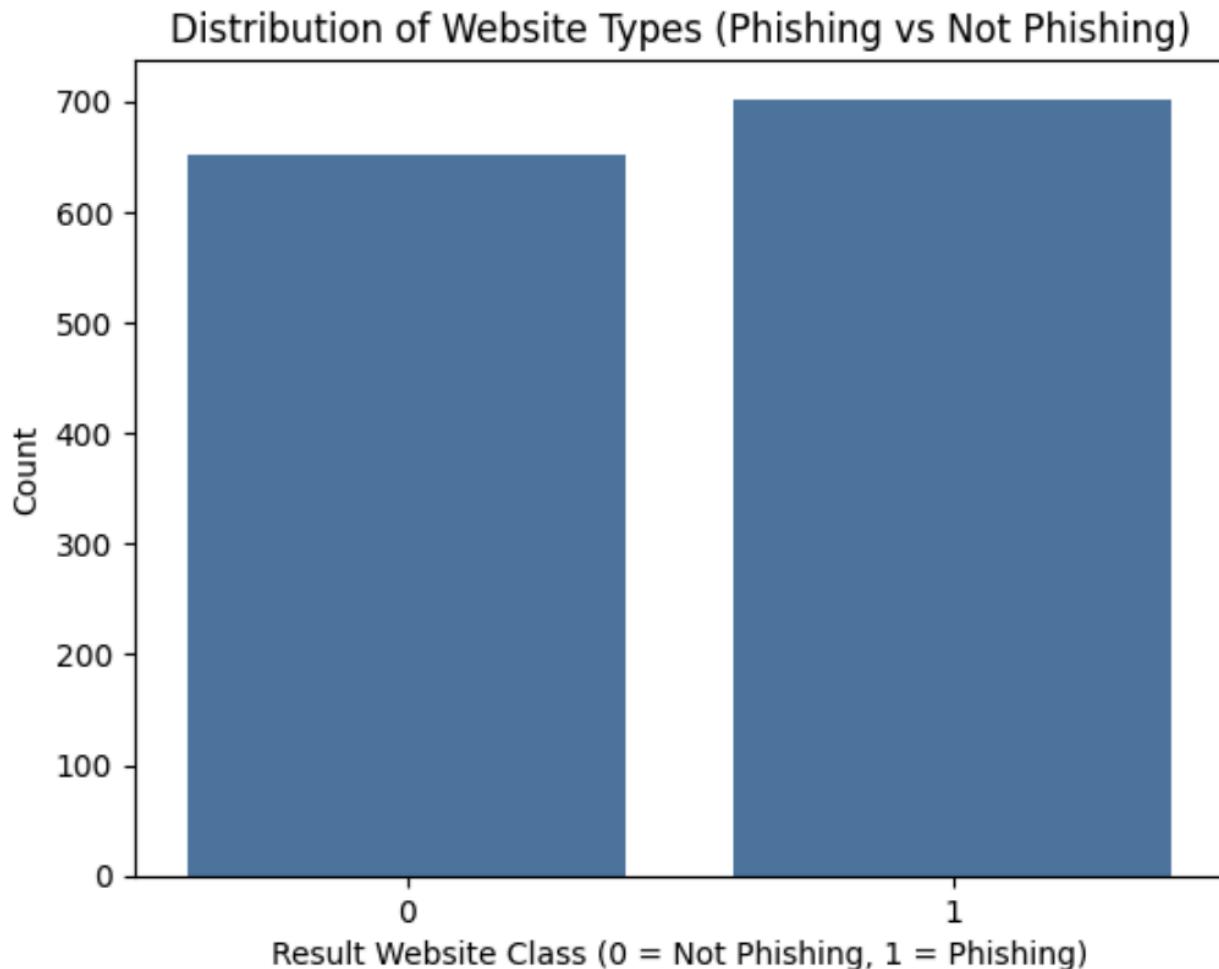
After thorough inspection, the dataset was found to have no missing values or inconsistent data types. Therefore, no imputation or removal of rows/columns was necessary.

Exploratory Data Analysis (EDA): Visualizations, such as bar charts, were used to understand the relationship between specific features (e.g., pop-up windows) and the likelihood of phishing.

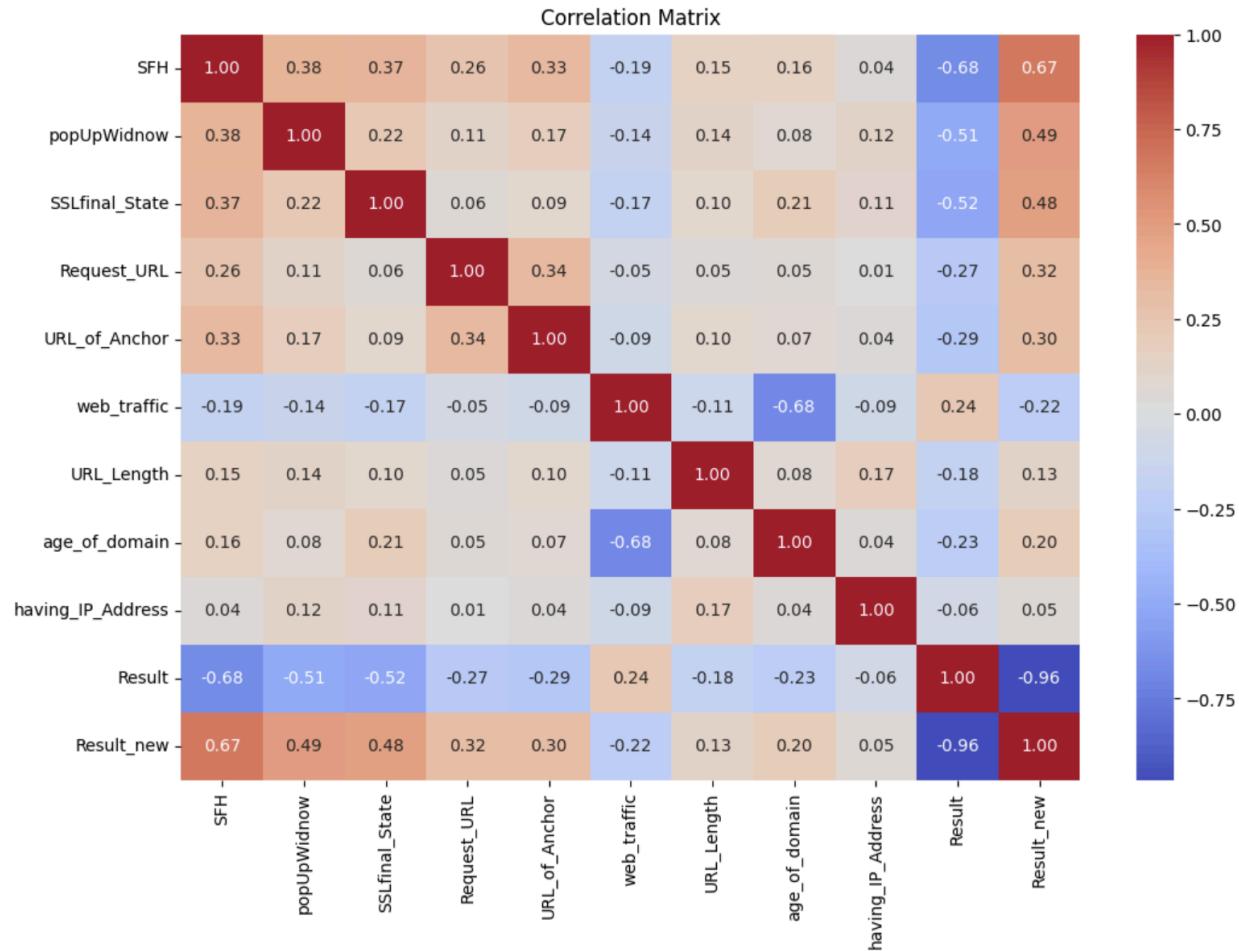
Modelling Approach: The classification task is approached using several machine learning models, including Decision Trees, Random Forest, Logistic Regression, and XGBoost. The models are trained and evaluated using metrics like accuracy, ROC-AUC, and confusion matrices to assess classification performance.

By executing this pipeline, we aim to build a reliable, interpretable system that can be integrated into browser security tools or used by cybersecurity analysts to proactively flag potentially harmful websites.

2. [Marks: 10] Explore your dataset and provide at least 5 meaningful charts/graphs with an explanation.



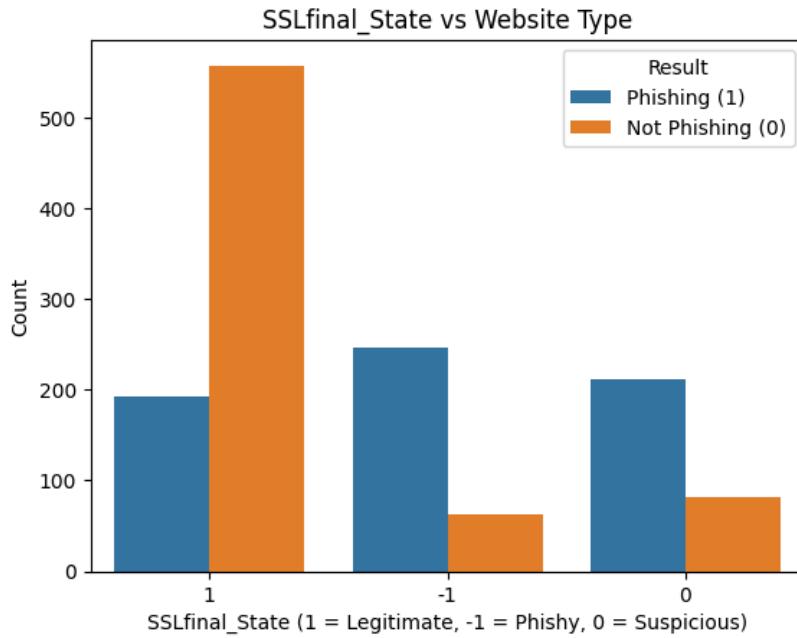
After preprocessing, we re-mapped the target variable Suspicious (0) merged into Legitimate (0) as negative Phishing (-1) converted to Phishing (1) as positive. The classes are now nearly balanced, which is ideal for most machine learning models later. Binary labels simplify model training and make metrics like accuracy, precision, and recall more straightforward to interpret.



Result vs. Result_new with correlation = -0.96 between Result and Result_new. This very strong negative correlation confirms that the modified label is an inversion of the original, due to label remapping. It validates that the transformation was applied correctly, converting a 3-class label into a binary label.

The following features are most correlated with phishing behaviour:

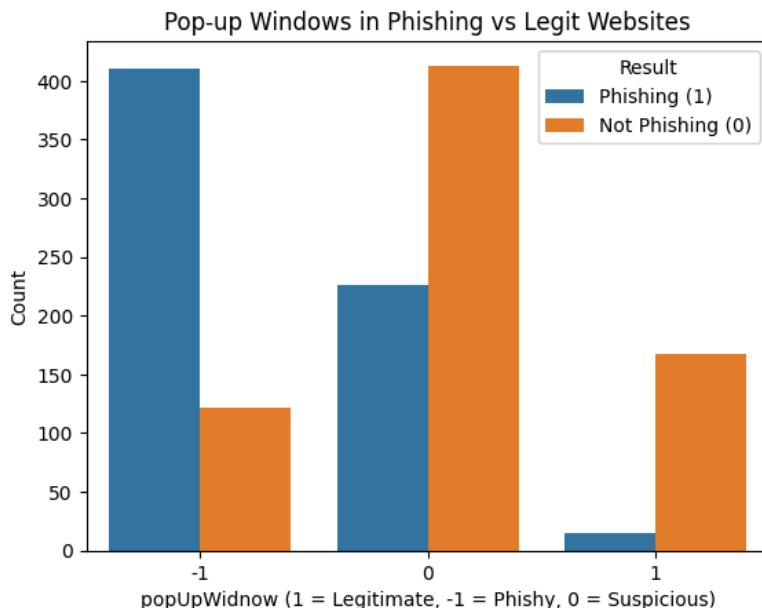
- SFH +0.67 Insecure/missing form handling is highly indicative of phishing
- popUpWindow +0.49 Use of pop-ups is strongly associated with phishing
- SSLfinal_State +0.48 Poor or missing SSL is a phishing red flag
- Request_url +0.32 Phishing sites often use external or suspicious request URLs
- URL_of_Anchor +0.30 Fake/misleading anchor links appear frequently on phishing sites



Websites with an SSL state of 1 (valid SSL certificate) are mostly legitimate.

Websites with -1 (invalid/missing SSL) or 0 (suspicious certificate) are primarily phishing.

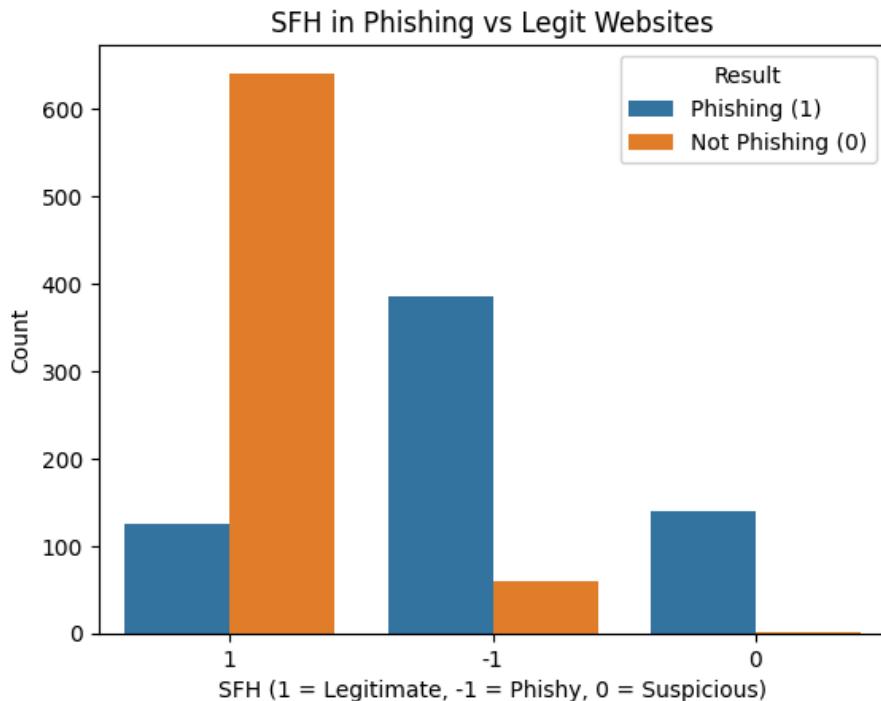
Phishing websites frequently lack proper SSL certification, or they attempt to spoof SSL by using expired or self-signed certificates. This feature is a strong signal for phishing detection.



Strongest signal: `popUpWidnow = -1` is highly associated with phishing.

Moderate signal: `popUpWidnow = 1` leans toward legitimate sites.

Weakest signal: `popUpWidnow = 0` is not clearly indicative either way.



when SFH = 1 (legitimate handler): Most common in legitimate websites

when SFH = -1 (phishing handler): Strongly associated with phishing websites

when SFH = 0 (suspicious): Primarily seen in phishing cases

The SFH attribute indicates how form data is handled. Phishing websites often submit data to blank or external URLs, which is unusual for legitimate websites. A malformed or empty SFH field is a major phishing red flag.

3. [Marks: 10] Do data cleaning/pre-processing as required and explain what you have done for your dataset and why?

i.

- Load the dataset and inspect its structure

Data columns (total 10 columns):			
#	Column	Non-Null Count	Dtype
0	SFH	1353 non-null	object
1	popUpWidnow	1353 non-null	object
2	SSLfinal_State	1353 non-null	object
3	Request_URL	1353 non-null	object
4	URL_of_Anchor	1353 non-null	object
5	web_traffic	1353 non-null	object
6	URL_Length	1353 non-null	object
...			
9	Result	1353 non-null	object

dtypes: object(10)

- Check for data cleaning and pre-processing

ii.

After loading the dataset and converting it into a DataFrame from .arff, we perform the following checks:

- Missing values
- Incorrect or inconsistent data types
- Irrelevant or redundant features
- Class imbalance

```
Missing values: 0
Result
-1    702
1     548
0     103
Name: count, dtype: int64
```

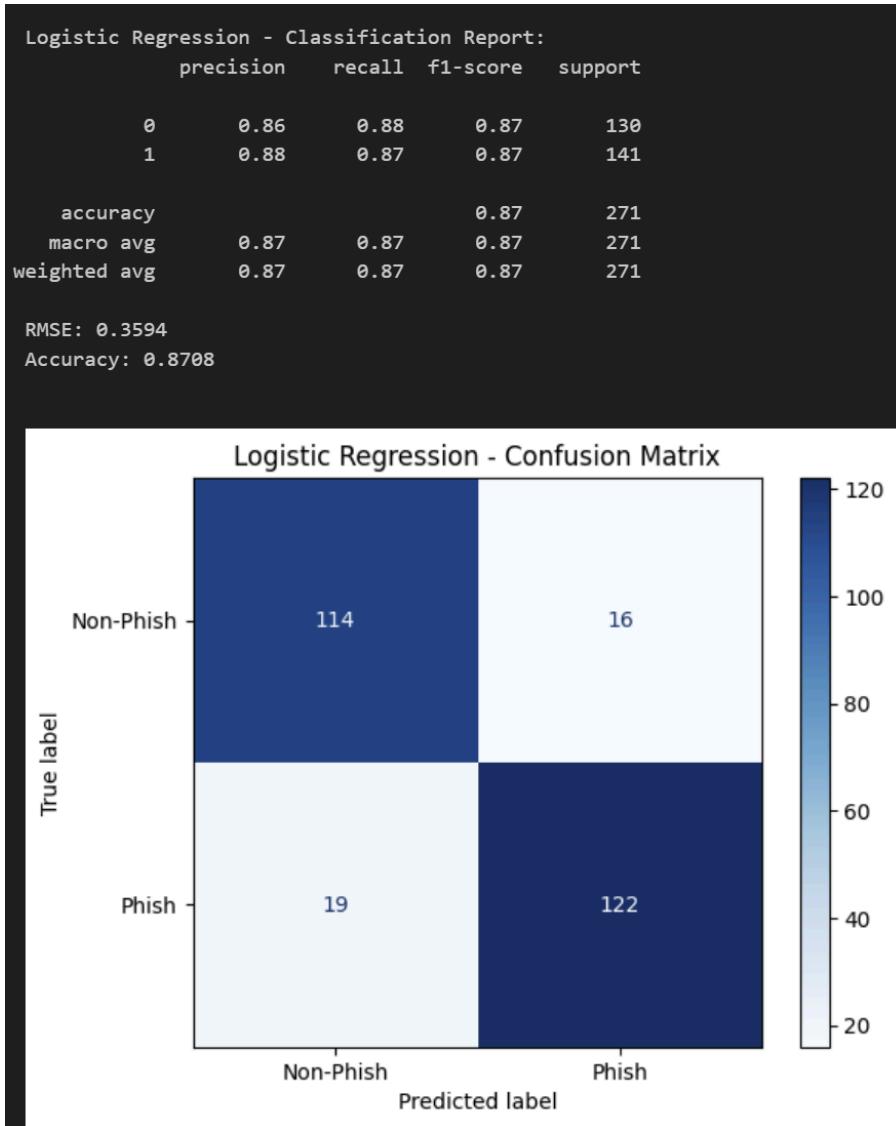
There are no missing values, so there is no need to deal with NA problems. However, for Legitimate, Suspicious and Phishy, these values have been replaced with numerical values 1,0 and -1, respectively. As the data was distributed unevenly for the phishing and legitimate classes (our target), we chose to modify the result class. By moving all suspicious classes to legitimate classes it makes the dataset binary and more balanced. In the new result class, 0 = Not Phishing, 1 = Phishing

Data structure after modification

```
Result_new
1    702
0    651
Name: count, dtype: int64
Result_new
1    51.884701
0    48.115299
Name: proportion, dtype: float64
```

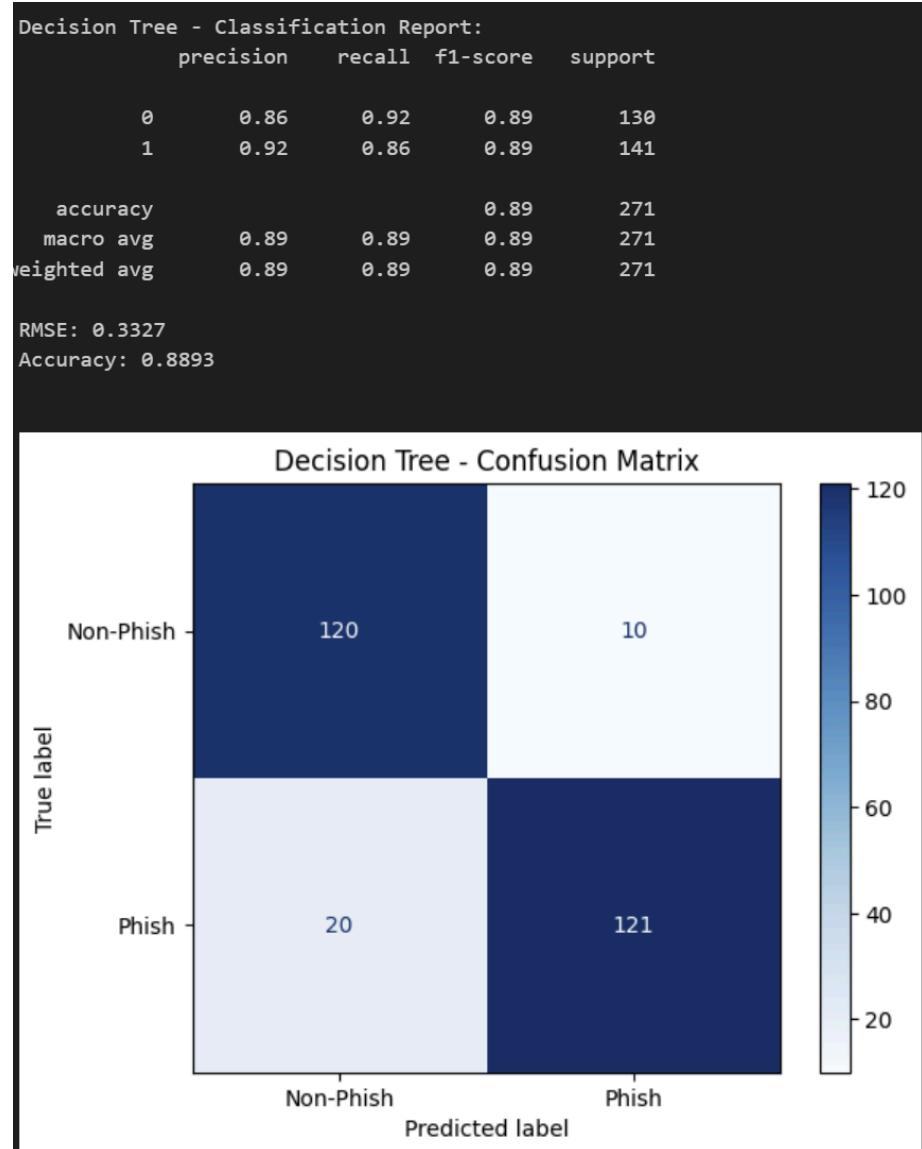
4. [Marks: 15] Implement 2 machine learning models and explain which algorithms you have selected and why. Compare them and show success metrics (Accuracy/RMSE/Confusion Matrix) as per your problem. Explain results.

- Logistic model



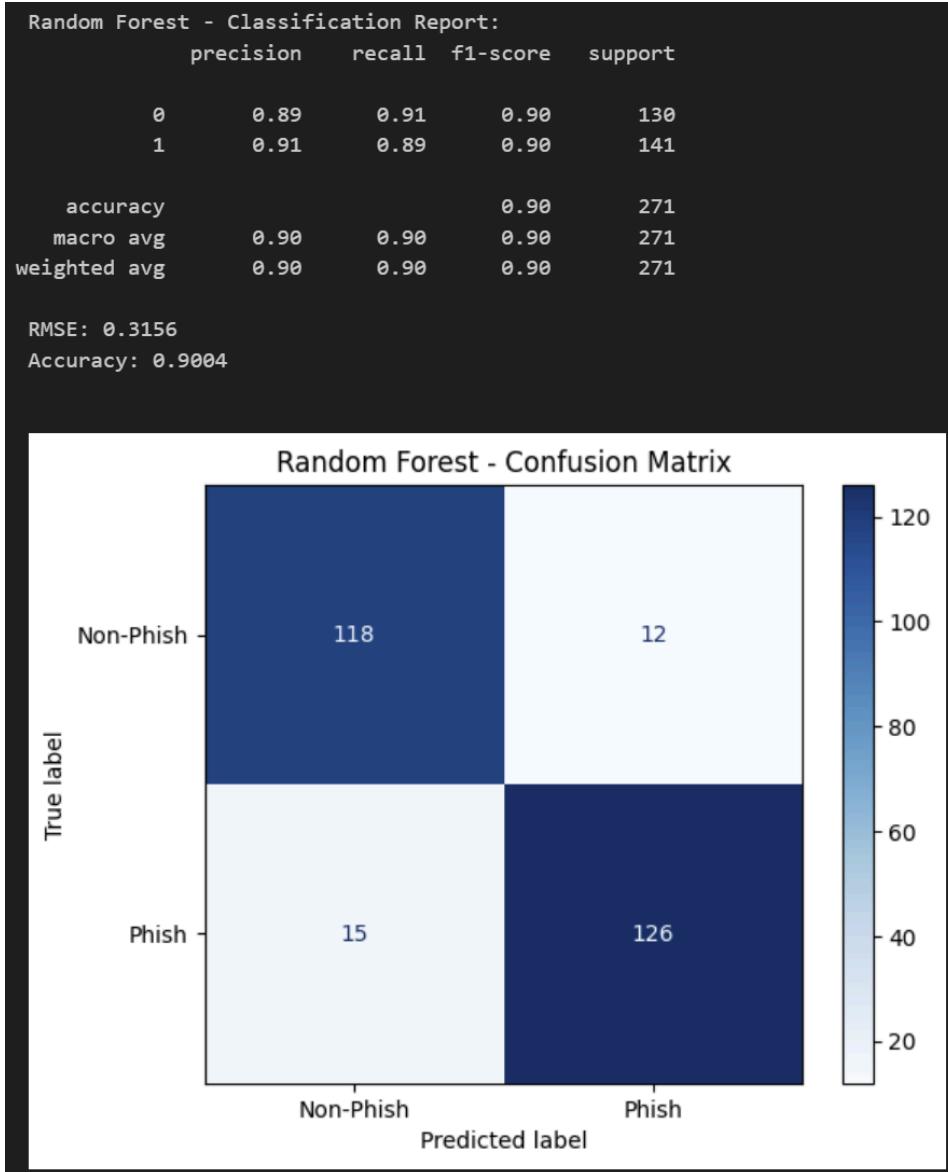
The logistic regression model performs well, with balanced precision and recall, a high accuracy of 87%, and relatively low misclassification errors. It can effectively differentiate phishing websites from legitimate ones, making it a reliable model for real-world phishing detection scenarios.

- Decision Tree



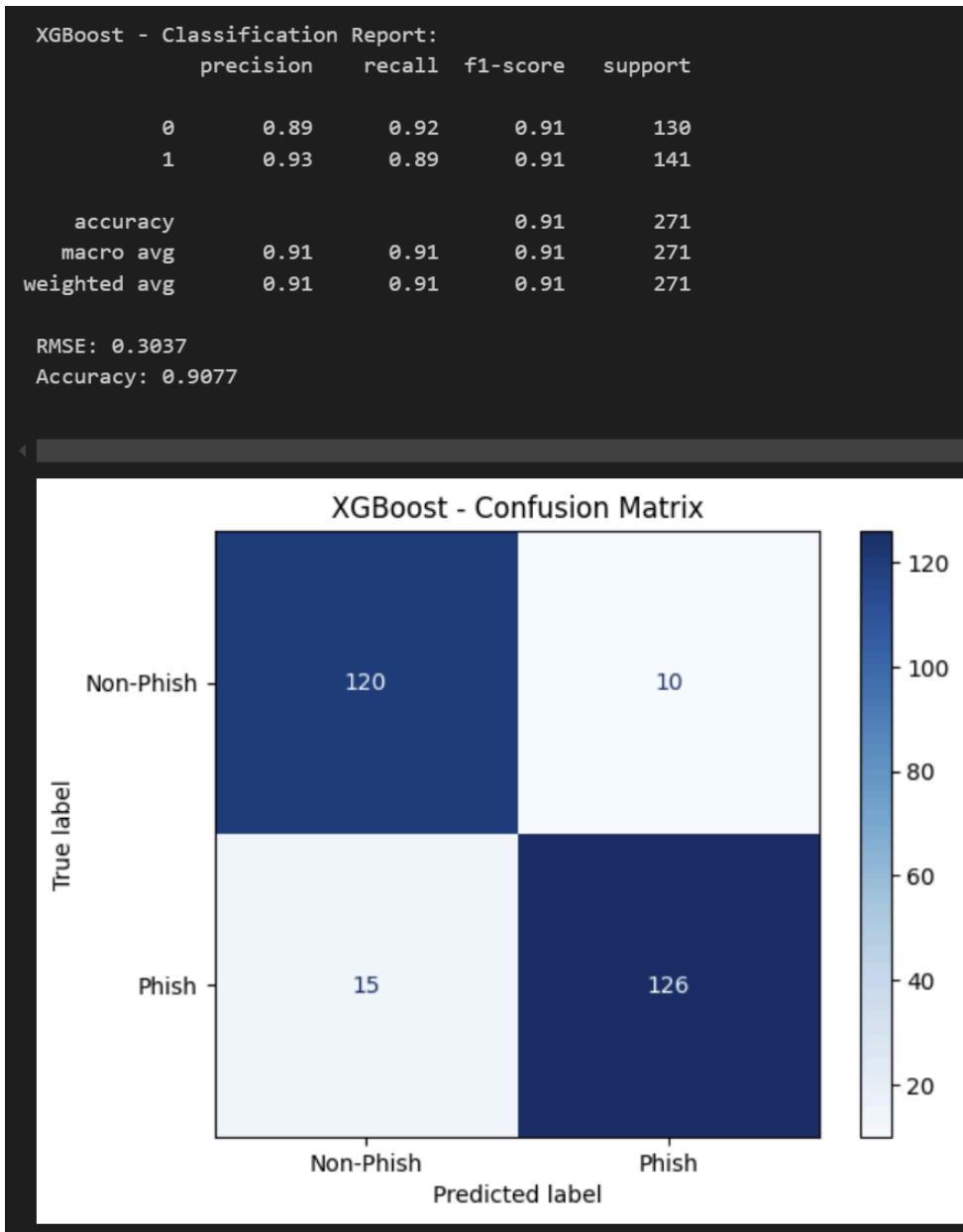
The Decision Tree model performs slightly better than Logistic Regression with higher accuracy (0.8893) and lower RMSE. It makes fewer false positives (10 vs. 16) while maintaining comparable performance across both classes. This model is well-suited for phishing detection due to its high interpretability and strong performance.

- Random Forest



The Random Forest model outperforms Logistic Regression and Decision Tree models across all metrics, with the highest accuracy (90.04%), the lowest RMSE, and balanced precision/recall for both classes. It is the most reliable and robust classifier among those evaluated for detecting phishing websites in this task.

- XGBoost



XGBoost delivers the best overall performance, achieving 91% accuracy, balanced precision/recall, and the lowest RMSE across all tested models. This makes it the strongest candidate for phishing detection, offering high reliability with minimal false predictions. Its slight edge over Random Forest is due to better probabilistic accuracy and reduced prediction error.

5. [Marks: 15] Deploy a run-time pipeline for your dataset using Azure Designer Studio.
Or do hyperparameter tuning for your algorithms. Explain your results.
Or use Automated ML for your data set. Explain the best model results.

Deploy a run-time pipeline for your dataset using Azure Designer Studio

1. Go to “Machine Learning Studio” and click “create a new pipeline”

The screenshot shows the Azure AI | Machine Learning Studio interface. The left sidebar has sections for Home, Model catalog, Authoring (Notebooks, Automated ML, Designer, Prompt flow), Assets (Data, Jobs, Components, Pipelines, Environments). The Designer section is selected. The main area is titled 'Designer' and 'New pipeline'. It shows two tabs: 'Classic prebuilt' (selected) and 'Custom'. A large central area has a blue plus sign and the text 'Create a new pipeline using classic prebuilt components'. Below this is a 'Pipelines' section with tabs for 'Pipeline drafts' (selected) and 'Pipeline jobs'.

2. In the new pipeline, we need to create a data asset of our data (if it doesn't exist)

The screenshot shows the Azure AI | Machine Learning Studio interface. The left sidebar has sections for Home, Model catalog, Authoring (Notebooks, Automated ML, Designer, Prompt flow), Assets (Data, Jobs, Components, Pipelines, Environments). The Designer section is selected. The main area is titled 'Authoring' and shows a search bar and filter options. A message says 'Success: phishing data asset created succ...'. Below is a table with one row, showing a 'phishing' asset by 'Bella Weng' with a 'Version 1' button. The right side of the screen shows pipeline-related controls like Undo, Redo, Validate, Save, Pipeline interface, and a status bar at the bottom.

3. Create a new data asset

a. Select a data type

Azure AI | Machine Learning Studio

Create data asset

1 Data type

2 Data source

3 Destination storage type

4 Destination storage type

5 File or folder selection

6 Settings

7 Schema

8 Review

Name * phishing

Description Data asset description

Type * Tabular

Use cases for data types

When should I use File type?

The File type is recommended in most scenarios when you are working with a single data file of any type (including tabular data). This type allows you to specify a file location by URI in a storage location on your local computer, an attached Datastore, blob/ADLS storage, or a publicly available http(s) location. There are many types of supported URLs. In the Azure Machine Learning CLI v2 or Python SDK v2, this data type is called `uri_file`. [Learn more about the uri_file type](#)

When should I use Folder type?

Back Next Cancel

b. Select a Data Source, in our pipeline, we selected “From Local Files”

Create data asset

1 Data type

2 Data source

3 Destination storage type

4 Destination storage type

5 File or folder selection

6 Settings

7 Schema

8 Review

Choose a source for your data asset

Choose the data source you want to create your asset from. A data source can be from a local storage location on your computer, from an attached datastore, from Azure storage, or from a publicly available web location.

From Azure storage

Create a data asset from registered data storage services including Azure Blob Storage, Azure file share, and Azure Data Lake.

From local files

Create a data asset by uploading files from your local drive.

From SQL databases

Create a dataset from Azure SQL database and Azure PostGreSQL database.

Back Next Cancel

c. Select a datastore

Create data asset X

Step 3: Destination storage type

Datastore type * Azure Blob Storage [Create new datastore](#)

Name	Storage name	Create
workspaceblobstore	assignment50932992050	Apr 12
workspaceartifactstore	assignment50932992050	Apr 12

Search datastore Filter Columns

Page 1 of 1 25/Page

[Back](#) [Next](#) [Cancel](#)

d. In the datastore, find the path to our csv file and upload file

Create data asset X

Step 4: File or folder selection

Choose a file or folder

Choose files or folders to upload from your local drive. If you upload multiple folders or files, they will be stored in a containing folder.

Upload path azureml://subscriptions/d7f748cb-9faf-4e94-ab42-0eddede7eefe/resourcegroups/ziyun.wen...

[Upload files or folder](#)

Overwrite if already exists [Upload files](#)

Information

What file types can I Supported file types incl (such as csv or tsv), Parq and plain text.

Where are files uplo: Files will be uploaded to datastore and made ava workspace.

[Back](#) [Next](#) [Cancel](#)

e. Config source file settings

Create data asset X

Progress: 1. Data type (Completed) 2. Data source (Completed) 3. Destination storage type (Completed) 4. File or folder selection (Completed) 5. Settings (In Progress) 6. Schema (Not Started) 7. Review (Not Started)

Settings
These settings determine how the data is parsed. The initial settings are automatically detected; you can change them as needed to reparse the data.

File format	Delimiter	Example	Encoding
Delimited	Comma	Field1,Field2,Field3	UTF-8

Column headers	Skip rows
All files have same headers	None

Dataset contains multi-line data ⓘ

ⓘ Note: Processing tabular files with multi-line data is slower because multiple CPU cores cannot be used to ingest the data in parallel. Checking this option may result in slower processing times.

Data preview

SFH	popUp...	SSLfina...	Reques...	URL_of...	web_tr...	URL_Le...	age_of...	having...	Result	Result
1	-1	1	-1	-1	1	1	1	0	0	0
-1	-1	-1	-1	-1	0	1	1	1	1	0
1	-1	0	0	-1	0	-1	1	0	1	0

Back Next Review Cancel

f. config table schema

Create data asset X

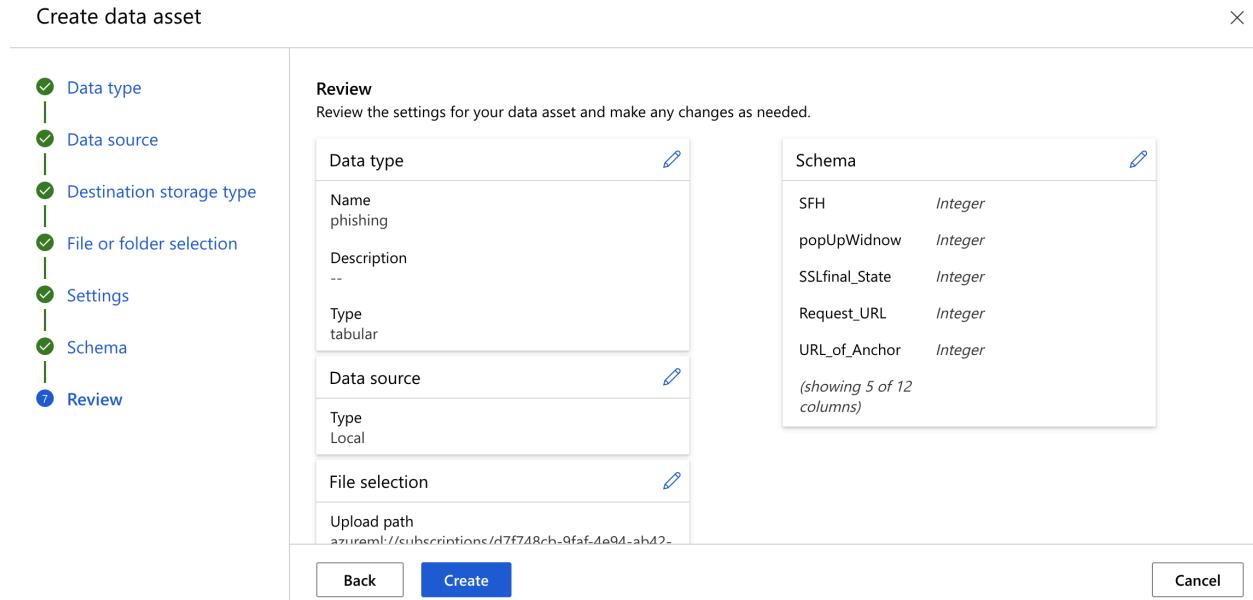
Progress: 1. Data type (Completed) 2. Data source (Completed) 3. Destination storage type (Completed) 4. File or folder selection (Completed) 5. Settings (Completed) 6. Schema (In Progress) 7. Review (Not Started)

Schema
Column types are auto-detected based on the initial subset of the data and can be updated here. Values not aligning with the specified column type will fail conversion and would be either null-filled or replaced with error value. Any conversions preview errors are non-blocking and you can proceed.

Incl...	Column name	Type	Example values	Date format <small>ⓘ</small>	Properties <small>ⓘ</small>
<input checked="" type="checkbox"/>	Path	String		Not applicable to s...	Not applicable t...
<input checked="" type="checkbox"/>	SFH	Integer	1, -1, 1	Not applicable to s...	Not applicable t...
<input checked="" type="checkbox"/>	popUpWidnow	Integer	-1, -1, -1	Not applicable to s...	Not applicable t...
<input checked="" type="checkbox"/>	SSLfinal_State	Integer	1, -1, 0	Not applicable to s...	Not applicable t...

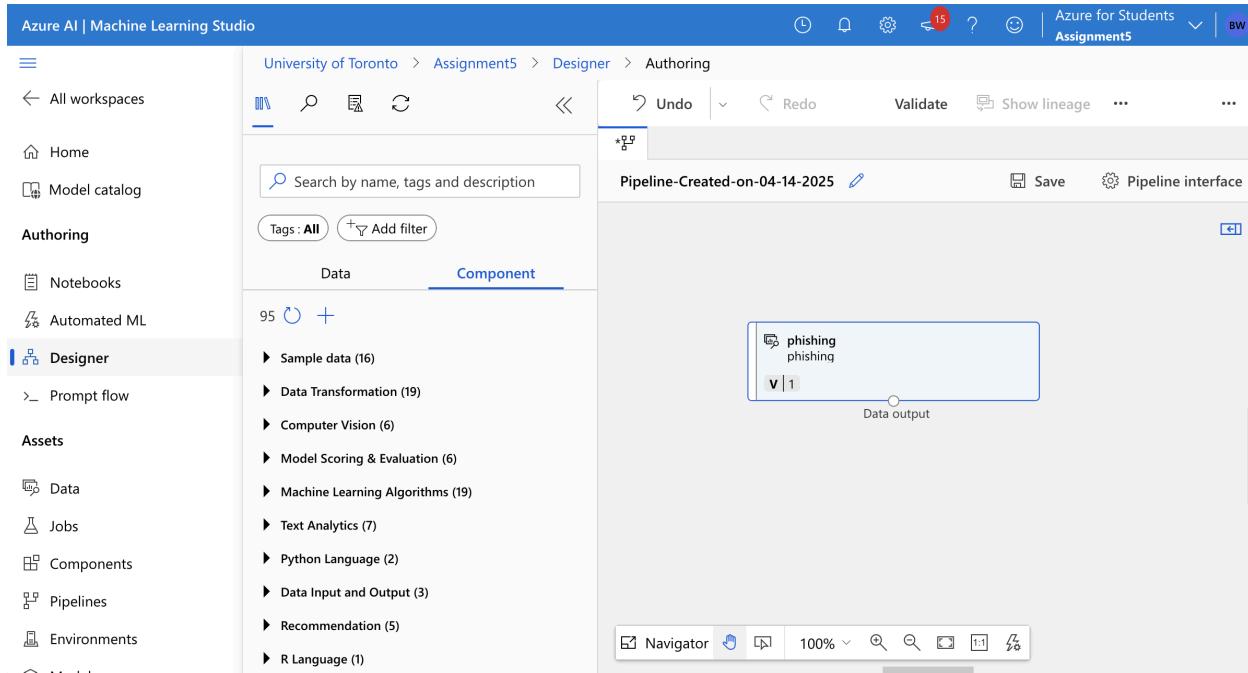
Back Next Cancel

g. Now, review and submit

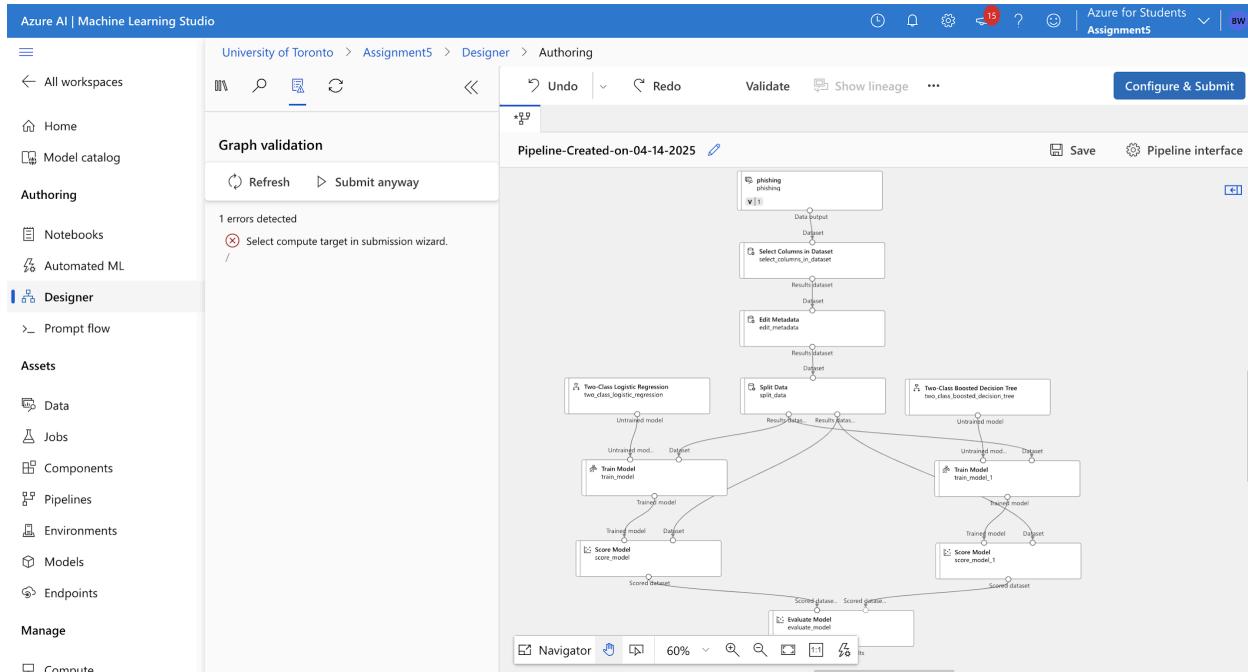


4. After adding an asset, you should be able to see it under the data tab. Drag it to our work space.

5. Now, you can add more components to fulfill your business logic



6. This Azure Machine Learning pipeline is built to classify phishing websites using two supervised models: Logistic Regression and Boosted Decision Tree. The process starts with importing the dataset, selecting relevant features, and editing metadata to define the target column. The data is then split into training and testing sets. Both models are trained separately and scored using the test data. Finally, the Evaluate Model module compares their performance using metrics like accuracy, precision, and AUC. This setup allows for an effective comparison to determine which model is better suited for phishing detection.



7. Now, we can config and submit our pipeline

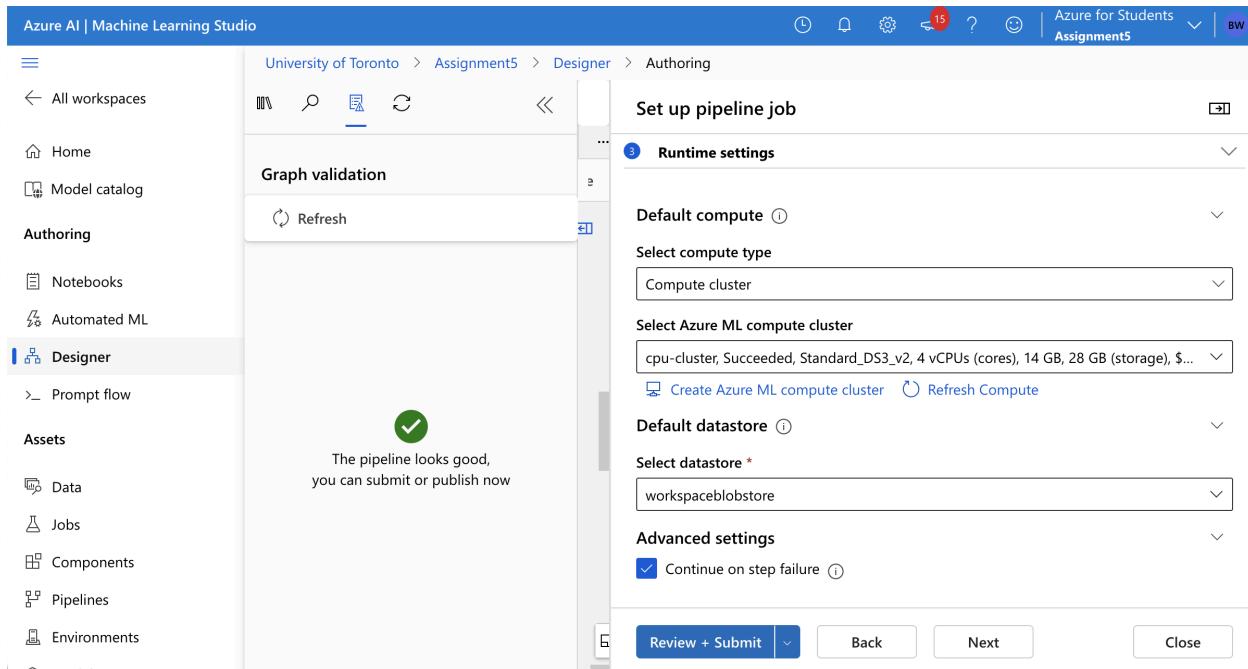
a. Basics

The screenshot shows the 'Set up pipeline job' interface in the 'Basics' step. The left sidebar is titled 'Designer'. The main area displays a green checkmark icon with the message 'The pipeline looks good, you can submit or publish now'. The right panel contains fields for 'Experiment name' (radio buttons for 'Select existing' and 'Create new', with 'Create new' selected and 'assignment5' entered), 'Job display name' ('Pipeline-Created-on-04-14-2025'), 'Job description' ('Pipeline created on 20250414'), and 'Job tags' (empty). At the bottom are buttons for 'Review + Submit', 'Back', 'Next', and 'Close'.

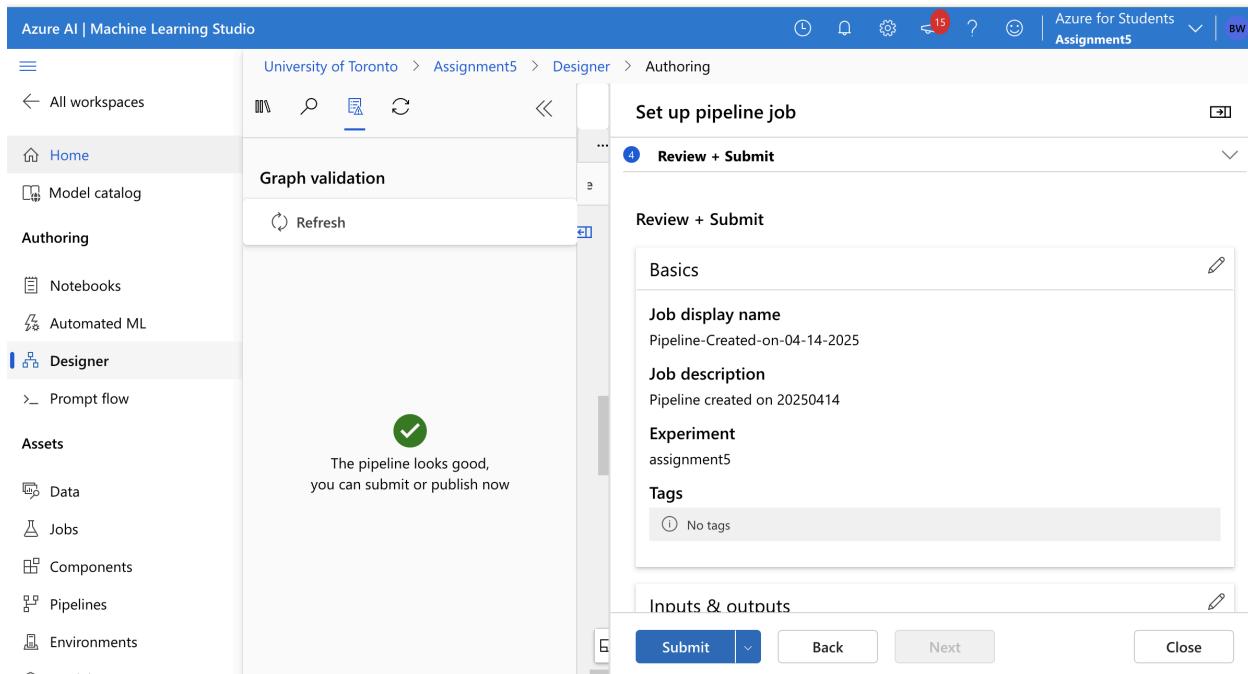
b. inputs and outputs

The screenshot shows the 'Set up pipeline job' interface in the 'Inputs & outputs' step. The left sidebar is titled 'Designer'. The main area displays a green checkmark icon with the message 'The pipeline looks good, you can submit or publish now'. The right panel contains sections for 'Inputs' ('No inputs') and 'Outputs' ('No outputs'). At the bottom are buttons for 'Review + Submit', 'Back', 'Next', and 'Close'.

c. runtime settings



d. submit



Outputs:

8. After we submit, we can go back to the Designer page. Under “Pipeline Jobs” we can see our new pipeline job.

Azure AI | Machine Learning Studio

University of Toronto > Assignment5 > Designer

Create a new pipeline using classic prebuilt components ⓘ

Pipelines

Pipeline drafts Pipeline jobs

Refresh Reset view

Search Filter Columns

Display name	Experiment	Status	Description
Pipeline-Created-on-04-14-2025	assignment5	Completed	Pipeline created
Pipeline-Created-on-04-14-2025	assignment5	Canceled	Pipeline created

<https://ml.azure.com/?wsid=/subscriptions/d7f748cb-9faf-4e94-a...>

9. We can click the job to view the status of each component.

Azure AI | Machine Learning Studio

University of Toronto > Assignment5 > Jobs > assignment5 > Pipeline-Created-on-04-14-2025

Outline Refresh Clone Resubmit View profiling Publish ...

Pipeline-Created-on-04-14-2025 Completed Share Add to compare Job overview

Type node name, comment or comp... Add filter

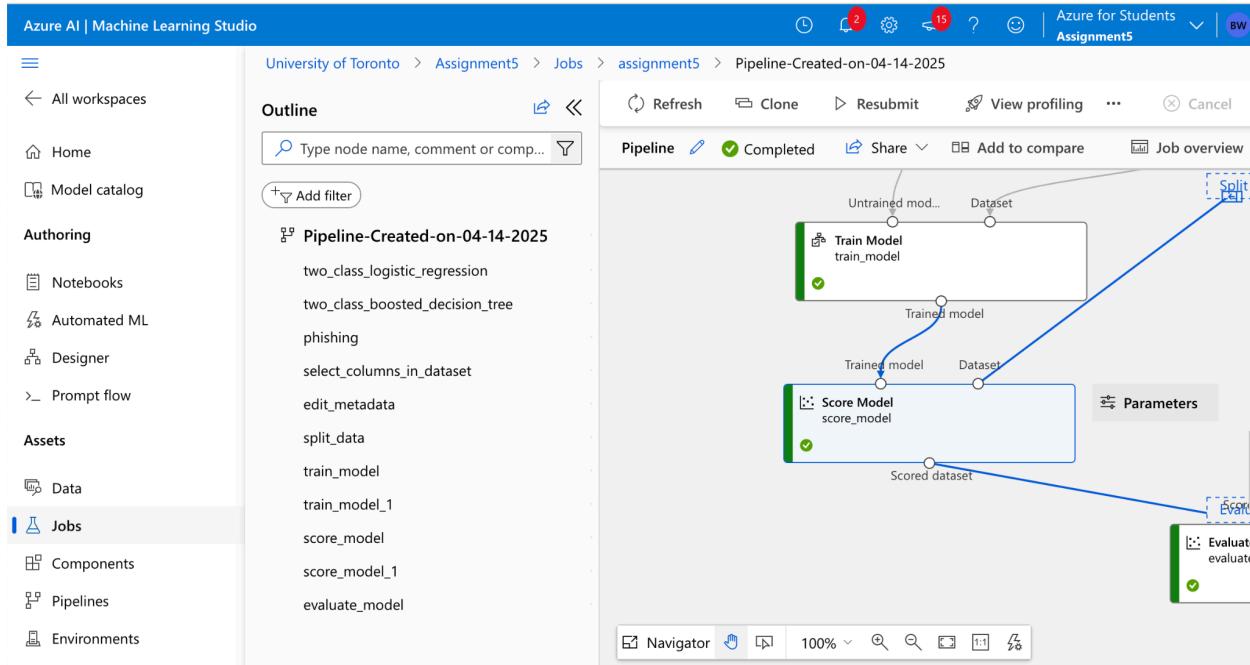
Pipeline-Created-on-04-14-2025

```

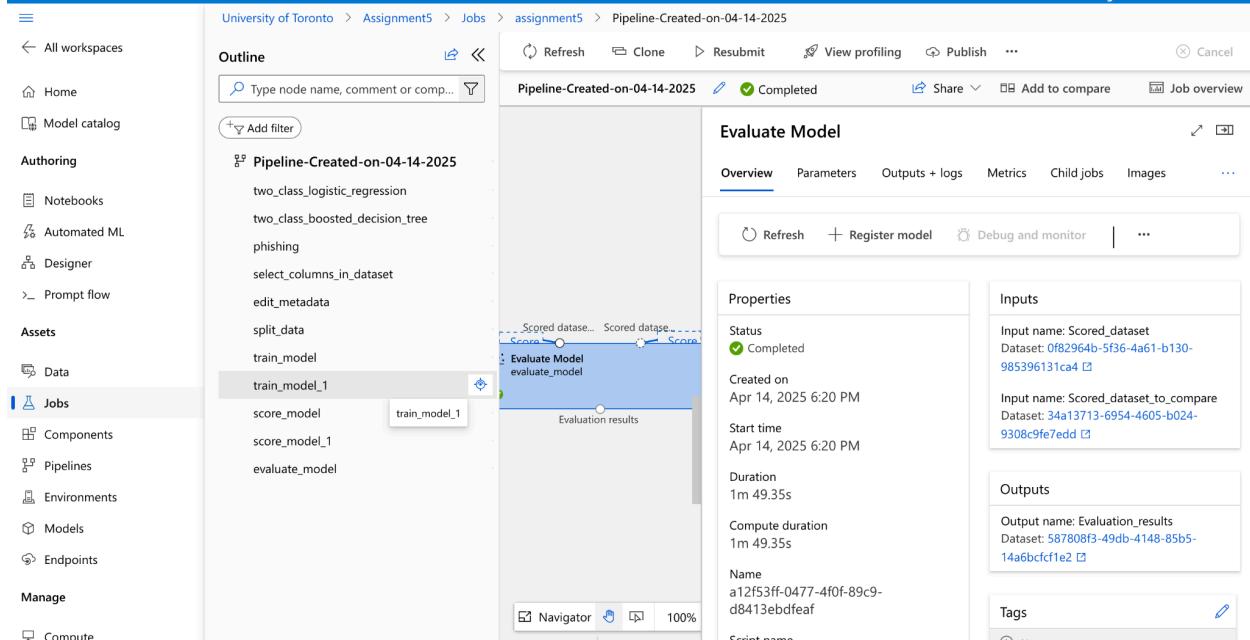
graph TD
    A[phishing] --> B[Select Columns in Dataset]
    B --> C[Edit Metadata]
    C --> D[Two-Class Logistic Regression]
    C --> E[Split Data]
    D --> F[Train Model train_model]
    E --> G[Train Model train_model_1]
    F --> H[Score Model score_model]
    G --> I[Score Model score_model_1]
    H --> J[Evaluate Model evaluate_model]
    I --> J
  
```

Navigator 60% 60%

10. We can also select a component and click “view in canvas” for more details.



11. Evaluate Model details



12. In the canvas, we can go to the “Outputs + logs” tab, click “Hide data outputs”, and click “Preview data” to view the result

Azure AI | Machine Learning Studio

University of Toronto > Assignment5 > Jobs > assignment5 > Pipeline-Created-on-04-14-2025

Outline

Type node name, comment or comp...

Pipeline-Created-on-04-14-2025

+ Add filter

two_class_logistic_regression
two_class_boosted_decision_tree_regression
phishing
select_columns_in_dataset
edit_metadata
split_data
train_model
train_model_1
score_model
score_model_1
evaluate_model

Evaluate Model

Overview Parameters Outputs + logs Metrics Child jobs Images ...

Refresh Register model Debug and monitor ...

Data outputs Hide data outputs

Evaluation results +

Other outputs

std.log.txt

logs module_statistics system_logs user_logs std.log.txt

```

155 Cleaning up all out
156 1 items cleaning up
157 Cleanup took 0.0548
158 Traceback (most rec
159 File "uridecode_i
160 execute(decoded
161 File "uridecode_i
162 exit(retval)
163 File "/azurerm/en
164 raise SystemExit(
165 0
166
167

```

13. Here's the output

Azure AI | Machine Learning Studio

Evaluation_results

Left port Right port

Scored dataset (left port) Scored dataset to compare (right port)

ROC curve

True positive rate
False positive rate

Precision-recall curve

Precision
Recall

Lift curve

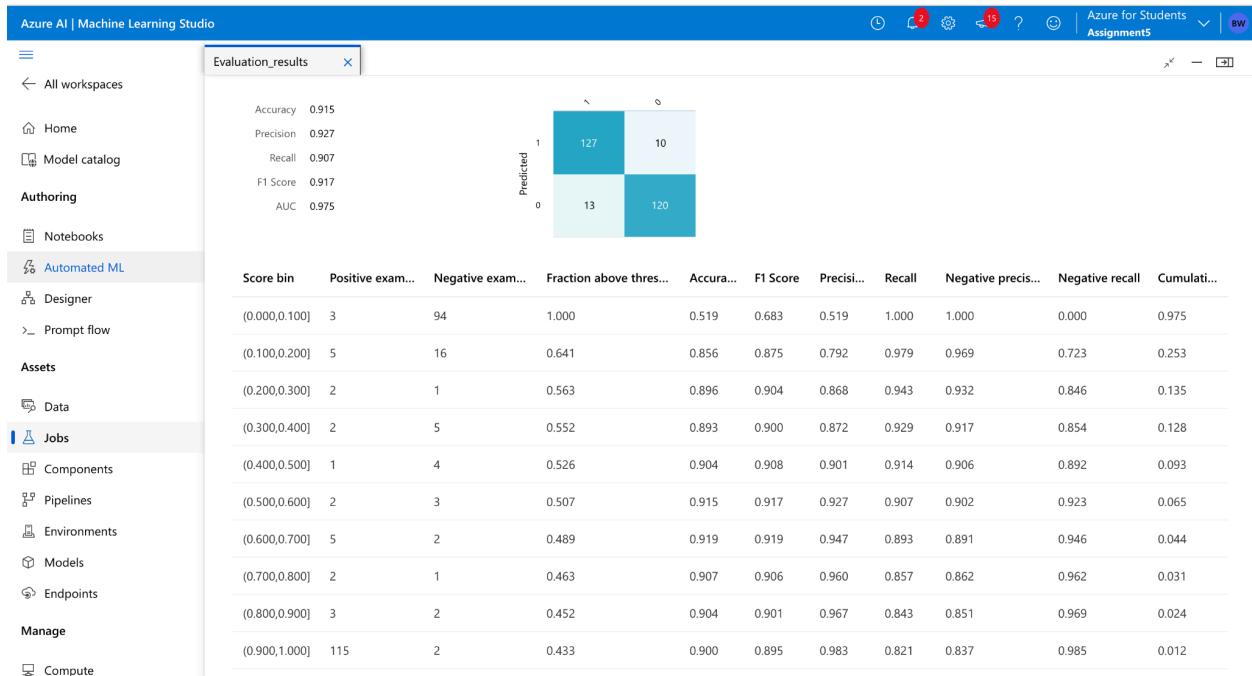
Number of true positives
Positive rate

Threshold: 0.5

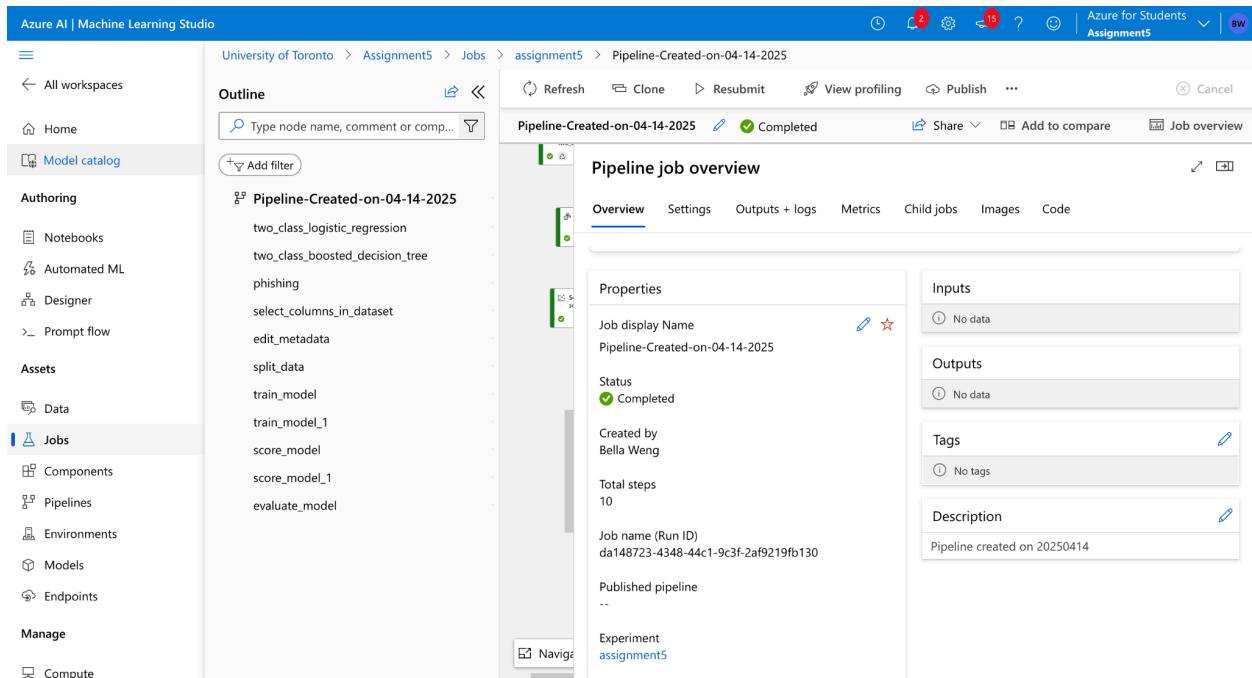
Actual

		1	0
1	127	10	
0	13	120	

Accuracy: 0.915
Precision: 0.927
Recall: 0.907
F1 Score: 0.917
AUC: 0.975



14. We can also click “Job overview” on the top right for more details about the pipeline job.



Display name (2 visualized)	Parent job name	Experiment	Status	Created on
Pipeline-Created-on-04-1(10)		assignment5	Completed	Apr 14, 2025 6:13 PM
evaluate_model	da148723-4348-44c1...	assignment5	Completed	Apr 14, 2025 6:20 PM
score_model_1	da148723-4348-44c1...	assignment5	Completed	Apr 14, 2025 6:19 PM
score_model	da148723-4348-44c1...	assignment5	Completed	Apr 14, 2025 6:18 PM
train_model_1	da148723-4348-44c1...	assignment5	Completed	Apr 14, 2025 6:18 PM
train_model	da148723-4348-44c1...	assignment5	Completed	Apr 14, 2025 6:18 PM
split_data	da148723-4348-44c1...	assignment5	Completed	Apr 14, 2025 6:17 PM
edit_metadata	da148723-4348-44c1...	assignment5	Completed	Apr 14, 2025 6:16 PM

6. [Marks: 15] Summarize your project's key findings and overall conclusions in a brief paragraph. Ensure your summary is firmly grounded in the data and analysis you've presented throughout your project. Offer meaningful insights that not only encapsulate your work but also lay a foundation for potential future research in this area. Your conclusions should be well-reasoned and directly supported by your results.

Our project successfully addressed the critical issue of phishing website detection by leveraging a real-world dataset and applying machine learning techniques.

After converting the original multi-class labels into a binary classification task to improve balance and interpretability, we evaluated several models, including Logistic Regression, Decision Tree, Random Forest, and XGBoost. Among them, XGBoost emerged as the best performer, achieving a 91% accuracy and the lowest RMSE, indicating strong predictive power with minimal classification errors. Through exploratory analysis, we identified key indicators of phishing behaviour such as insecure form handling (SFH), invalid SSL certificates, and malicious pop-up usage, which significantly contributed to model performance. These findings not only validate the model's reliability but also offer practical insights into the characteristics of phishing websites.

The deployed Azure ML pipeline further demonstrates the model's readiness for real-world applications. This work provides a solid foundation for future research, which could explore real-time detection using live URLs, feature engineering from HTML content, or integration with browser extensions for proactive user protection.