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.

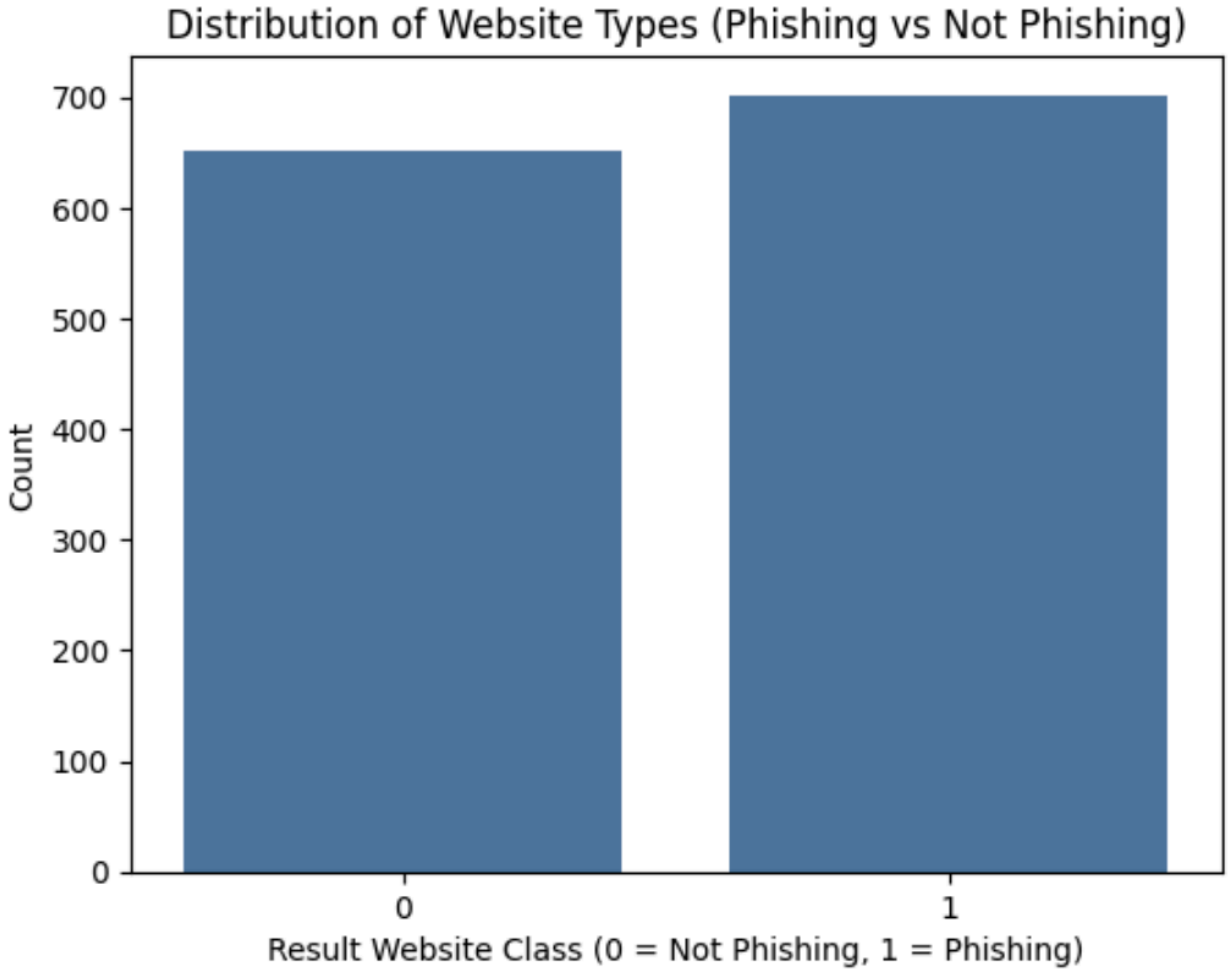
The dataset used in this project originates from the UCI Machine Learning Repository in `.arff` format and includes numerous categorical features that describe various characteristics of websites, such as use of pop-ups, URL structure, and security certificates. To simplify classification and address 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:

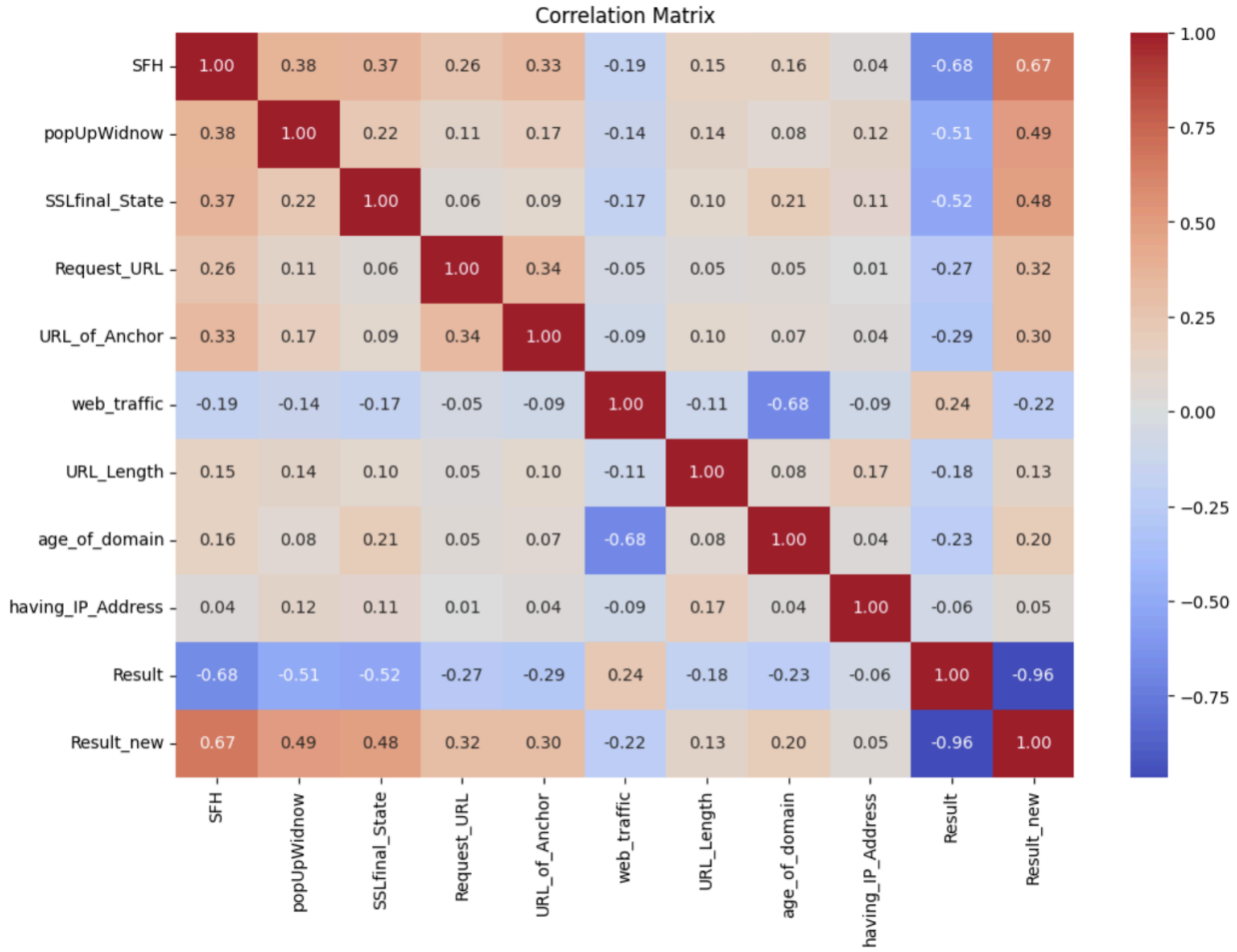
After thorough inspection, the dataset was found to have no missing values or inconsistent data types. Therefore, no imputation or removal of rows/columns was necessary. Exploratory Data Analysis (EDA): Visualizations, such as bar charts, were used to understand the relationship between specific features (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.



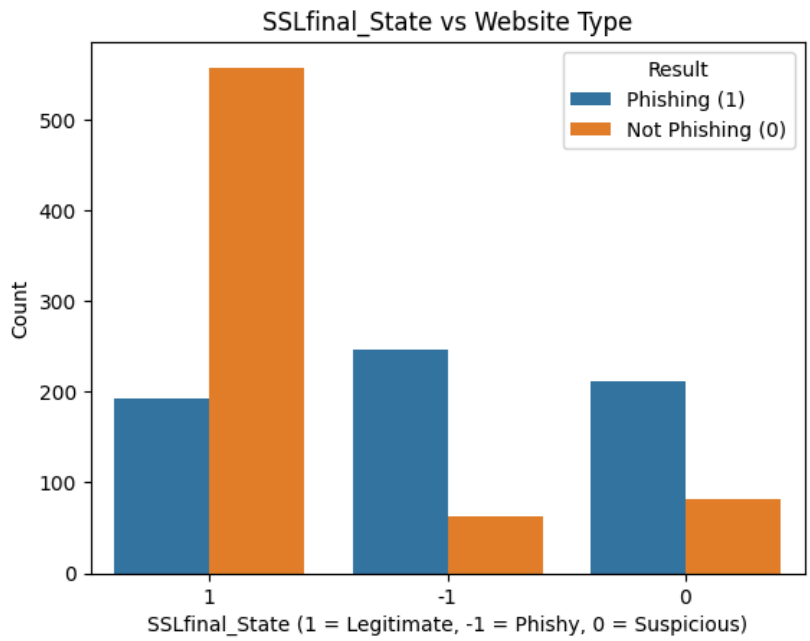
After preprocessing, we re-mapped the target variable Suspicious (0) merged into Legitimate (0) as negative Phishing (-1) converted to Phishing (1) as positive. The classes are now nearly balanced, which is ideal for most machine learning models later. Binary labels simplify model training and make metrics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



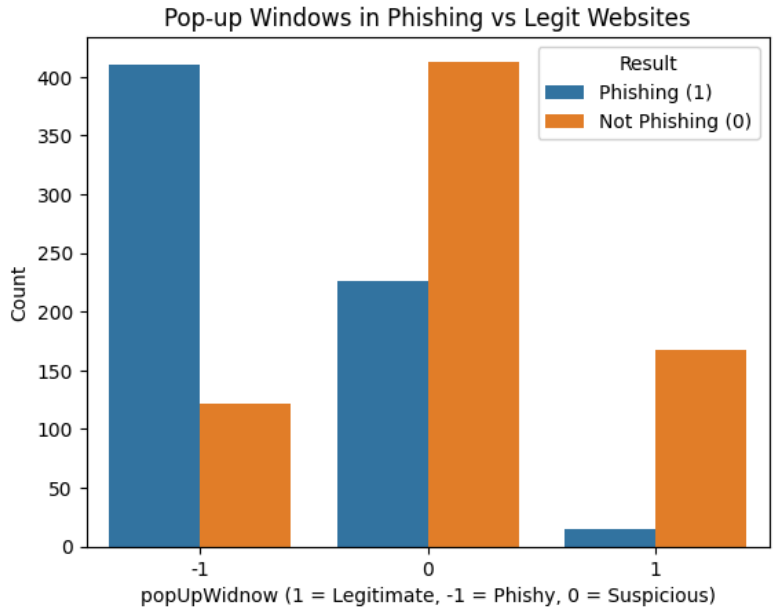
Result vs. Result_new with correlation = -0.96 between Result and Result_new. This very strong negative correlation confirms that the modified label is an inversion of the original, due to label remapping. It validates that the transformation was applied correctly, converting a 3-class label into a binary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

The following features are most correlated with phishing behaviour:

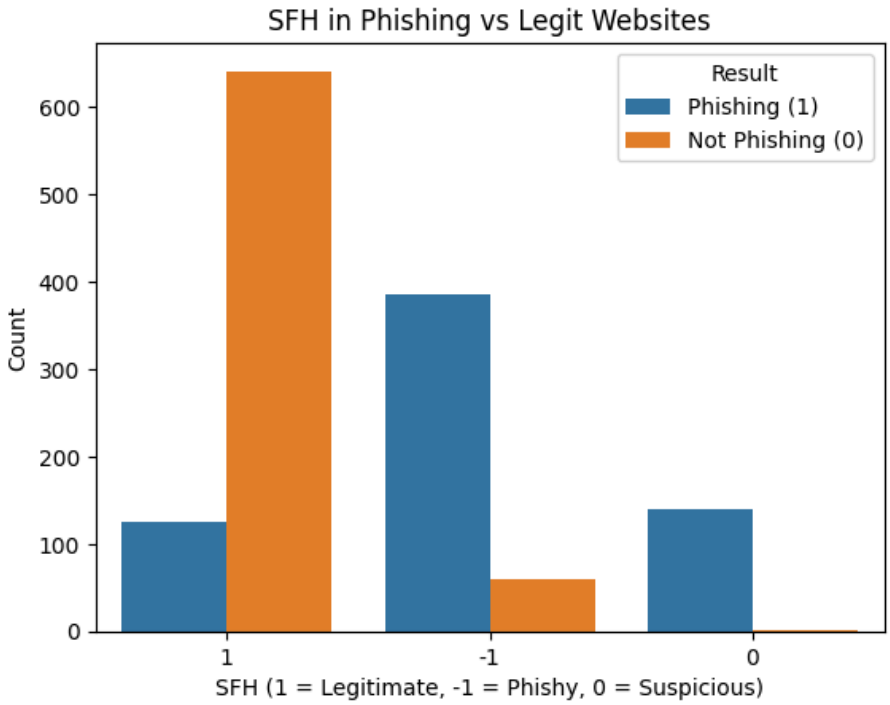
- SFH +0.67 Insecure/missing form handling is highly indicative of phishing
- popUpWindow +0.49 Use of pop-ups is strongly associated with phishing
- SSLfinal_State +0.48 Poor or missing SSL is a phishing red flag
- Request_url +0.32 Phishing sites often use external or suspicious request URLS
- URL_of_Anchor +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.

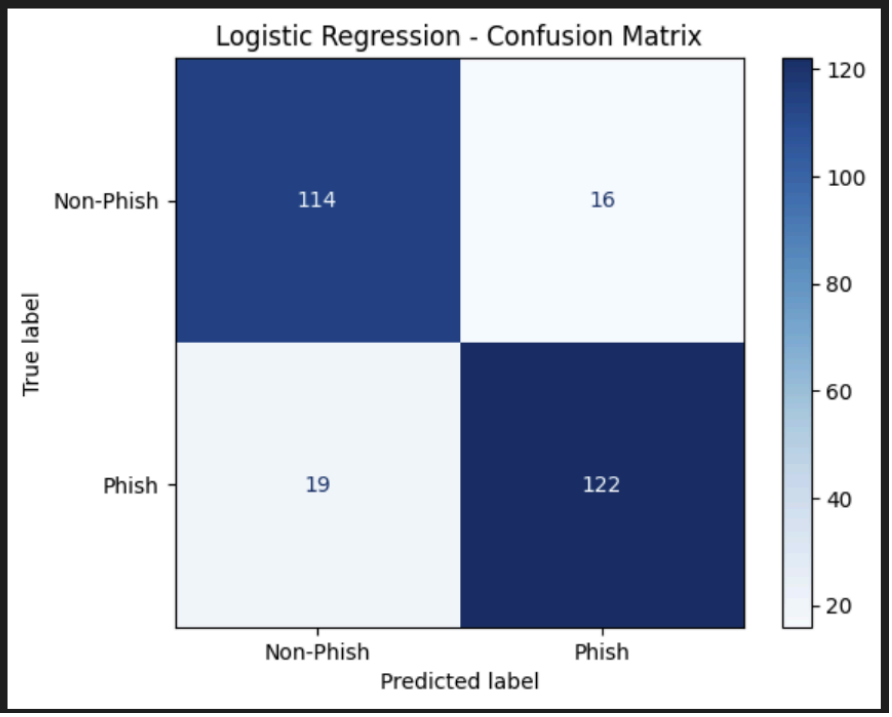
- Logistic model

```
Logistic Regression - Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.88      0.87       130
           1       0.88      0.87      0.87       141

    accuracy                           0.87       271
   macro avg       0.87      0.87      0.87       271
weighted avg       0.87      0.87      0.87       271

RMSE: 0.3594
Accuracy: 0.8708
```



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.

-

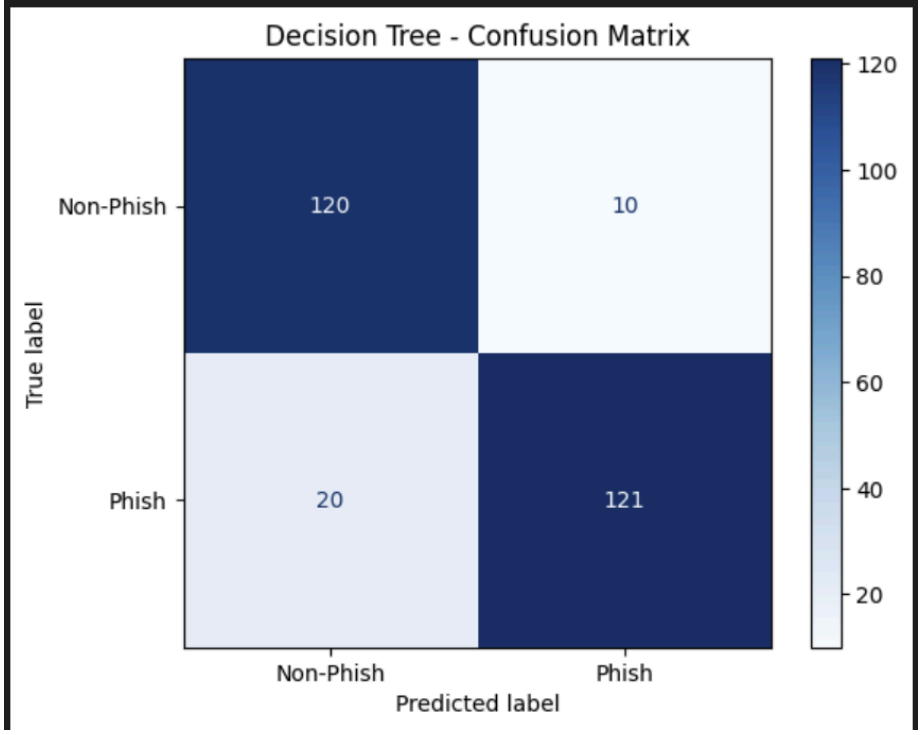
- Decision Tree

```
Decision Tree - Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.92      0.89       130
           1       0.92      0.86      0.89       141

    accuracy                           0.89       271
   macro avg       0.89      0.89      0.89       271
weighted avg       0.89      0.89      0.89       271

RMSE: 0.3327
Accuracy: 0.8893
```



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.

-

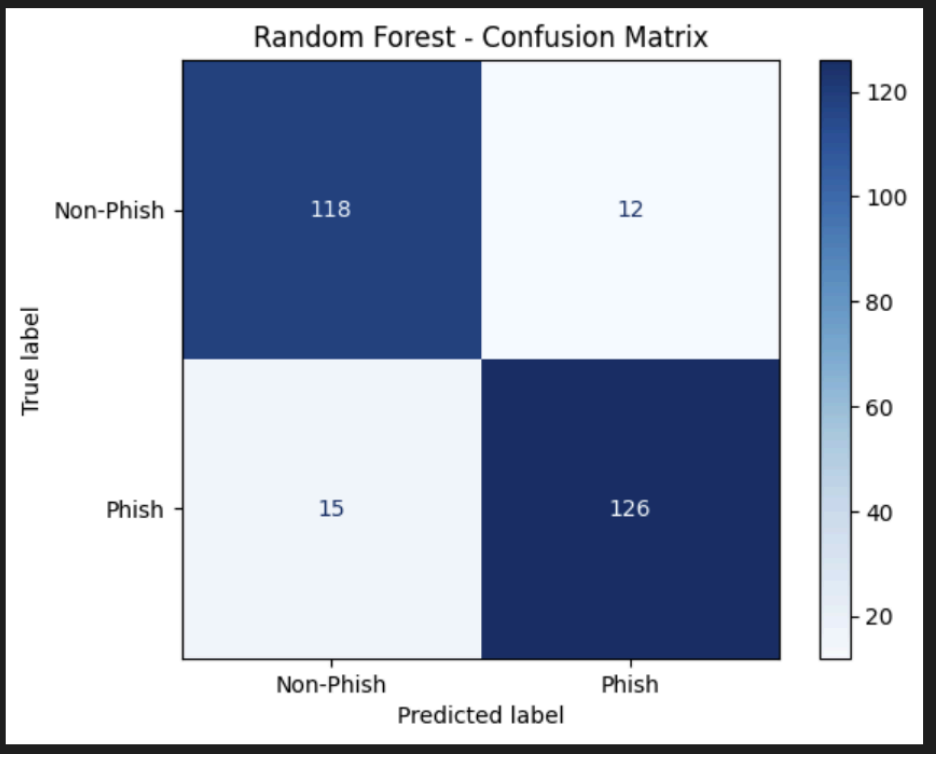
- Random Forest

```
Random Forest - Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.91      0.90       130
           1       0.91      0.89      0.90       141

    accuracy                           0.90       271
   macro avg       0.90      0.90      0.90       271
weighted avg       0.90      0.90      0.90       271

RMSE: 0.3156
Accuracy: 0.9004
```



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.

-

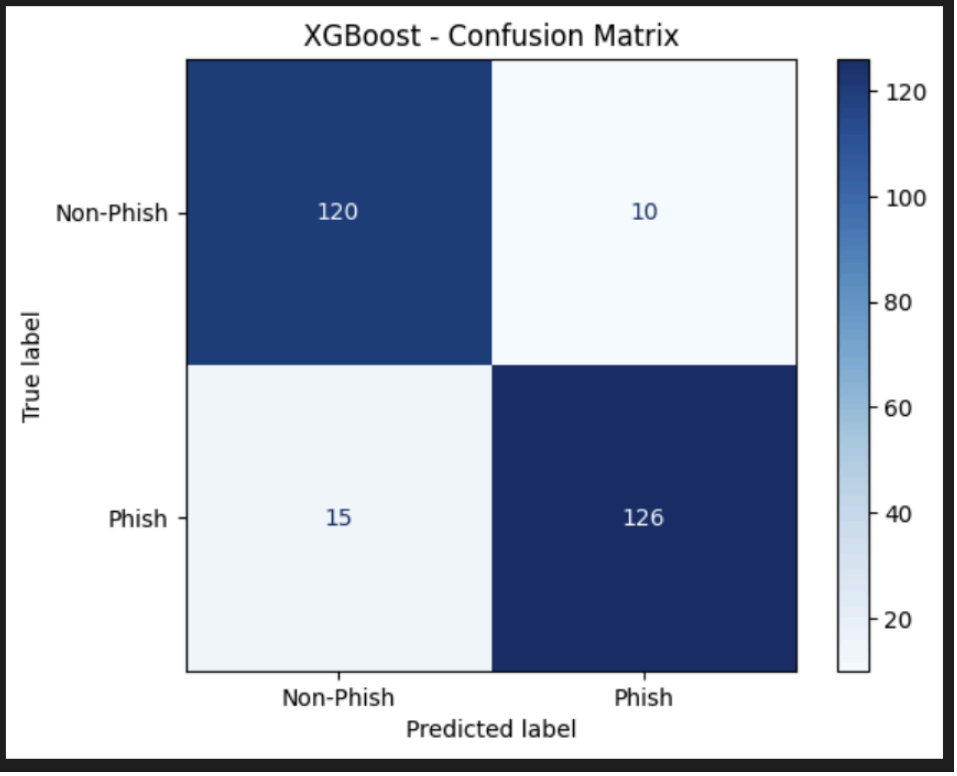
- XGBoost

```
XGBoost - Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.92      0.91       130
           1       0.93      0.89      0.91       141

    accuracy                           0.91       271
   macro avg       0.91      0.91      0.91       271
weighted avg       0.91      0.91      0.91       271

RMSE: 0.3037
Accuracy: 0.9077
```

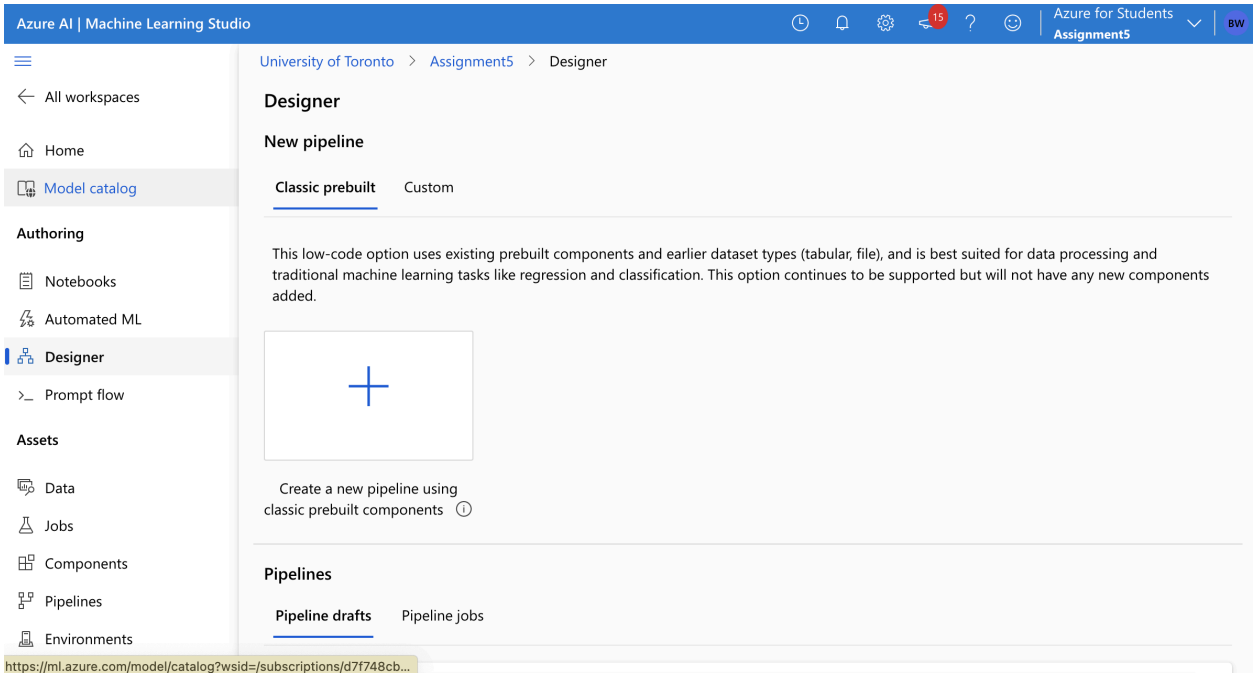


XGBoost delivers the best overall performance, achieving 91% accuracy, balanced precision/recall, and the lowest RMSE across all tested models. This makes it the strongest candidate for phishing detection, offering high reliability with minimal false predictions. Its slight edge over Random Forest is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

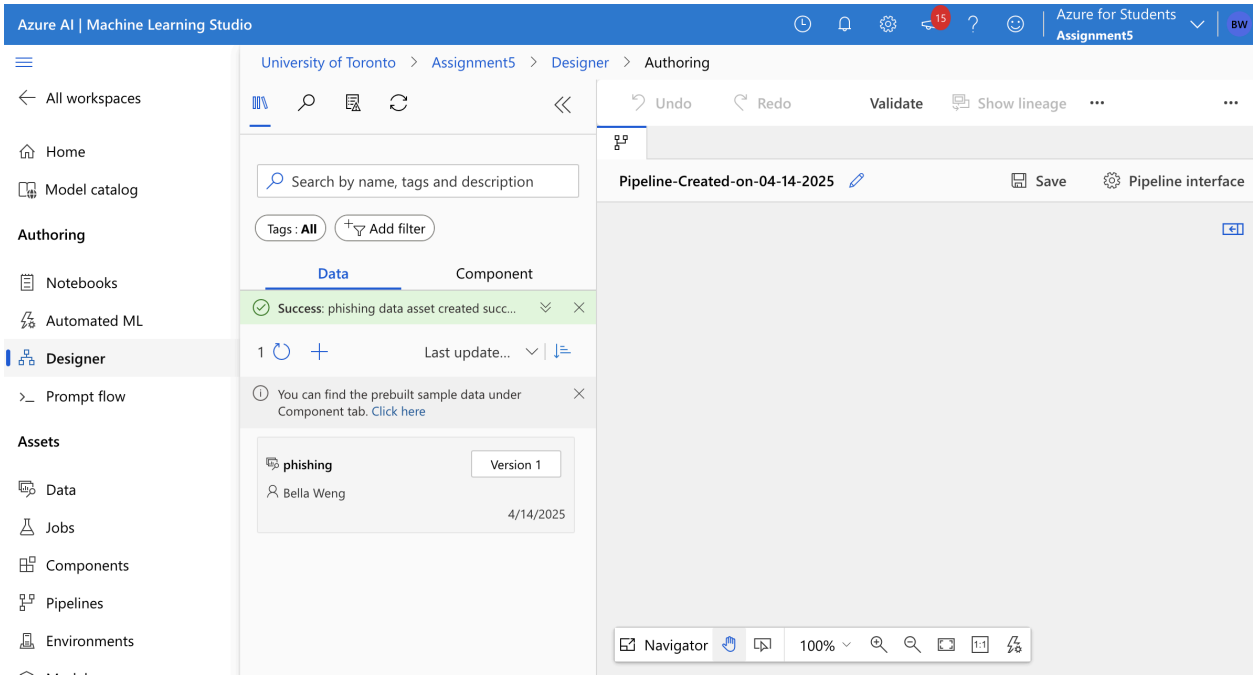
5. [Marks: 15] Deploy a run-time pipeline for your dataset using Azure Designer Studio.
Or do hyperparameter tuning for your algorithms. Explain your results.
Or use Automated ML for your data set. Explain the best model results.

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

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

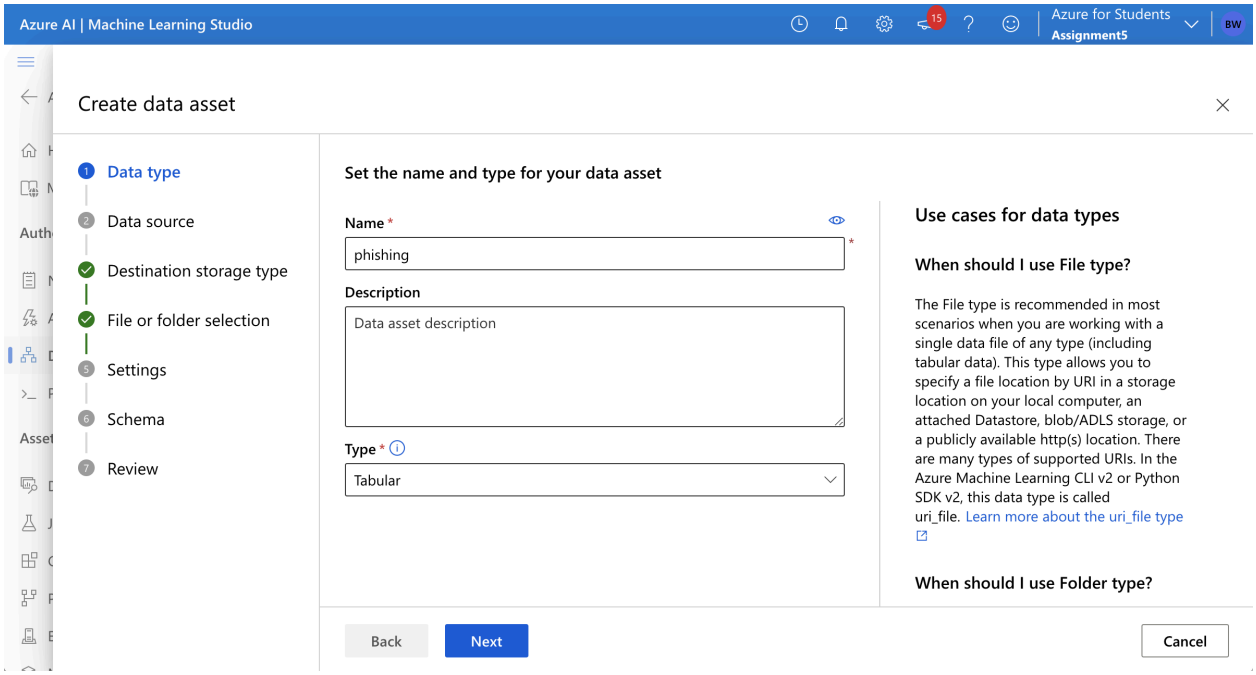


2. In the new pipeline, we need to create a data asset of our data (if it doesn’t exist)

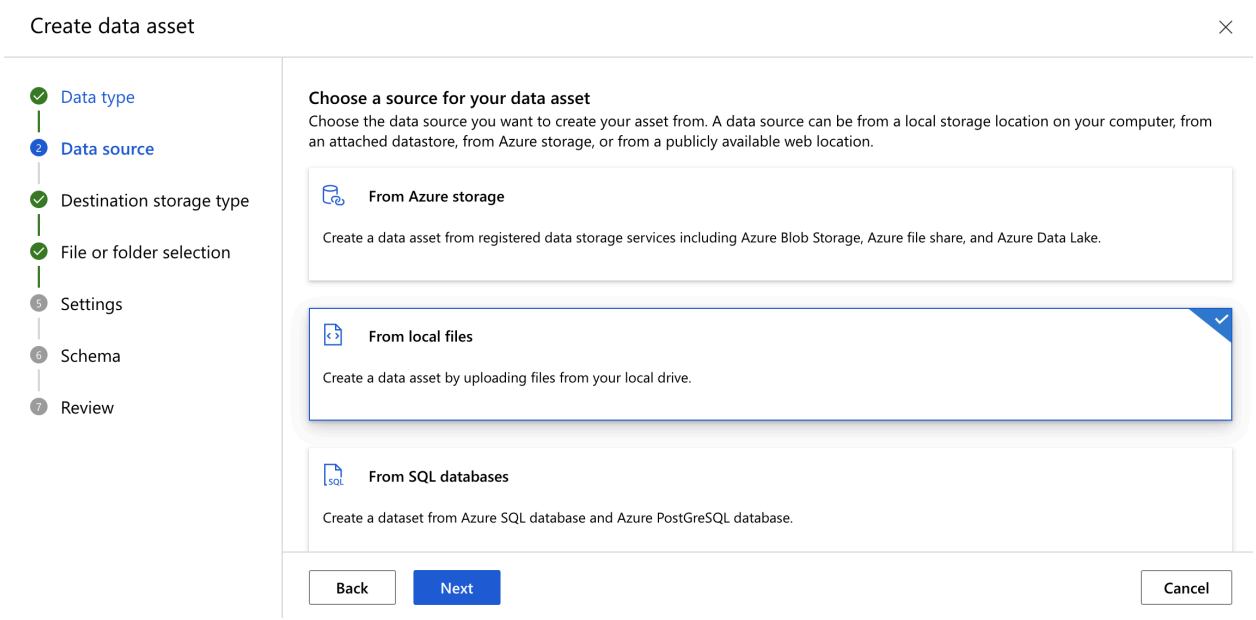


3. Create a new data asset

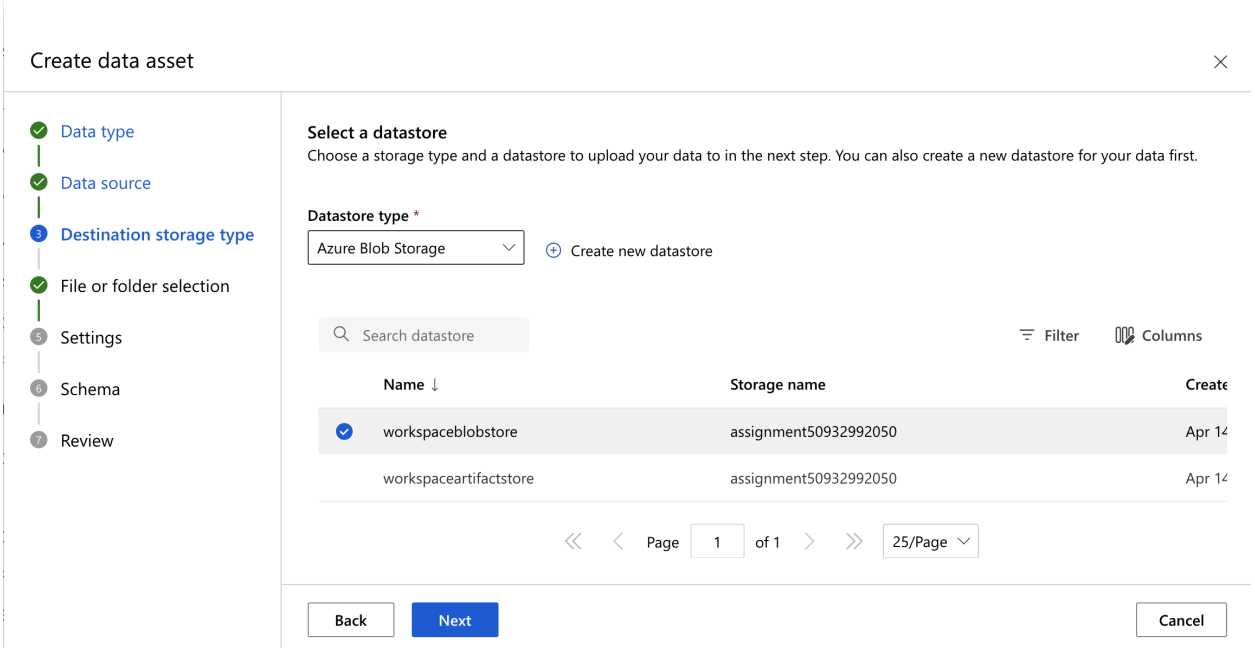
a. Select a data type



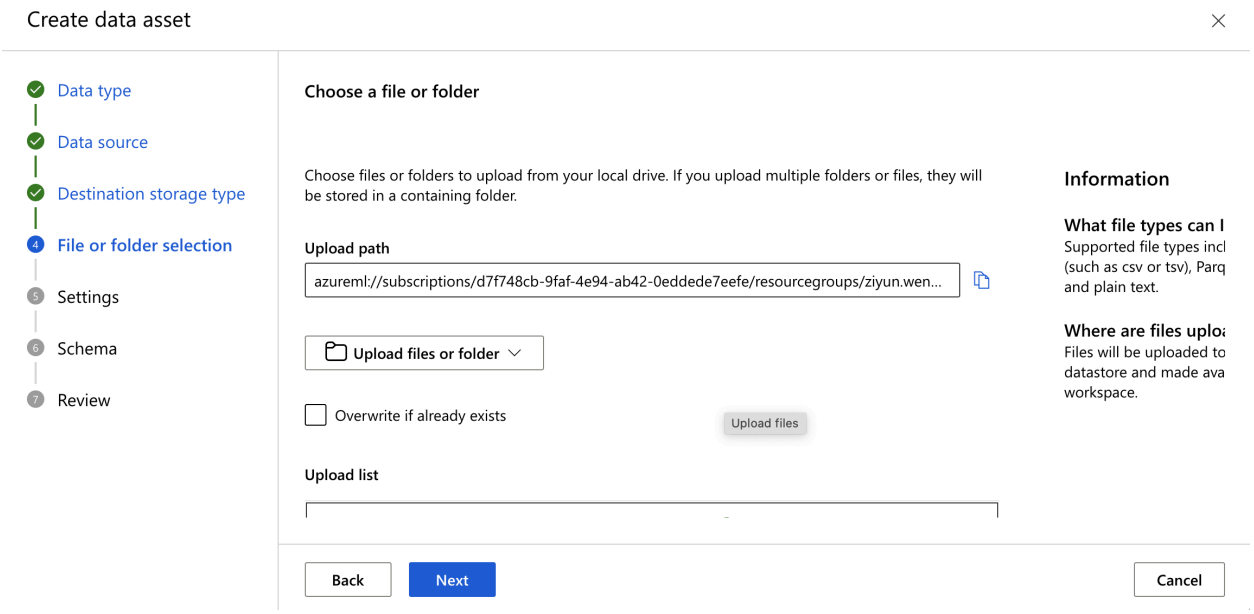
b. Select a Data Source, in our pipeline, we selected “From Local Files”



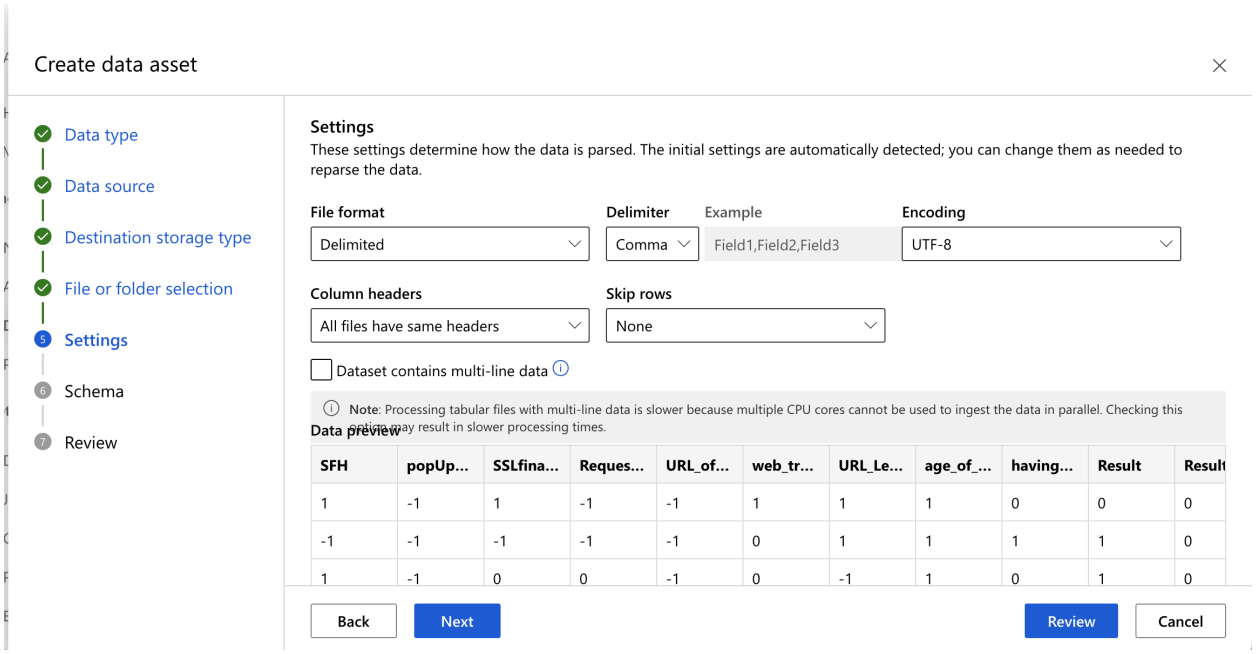
c. Select a datastore



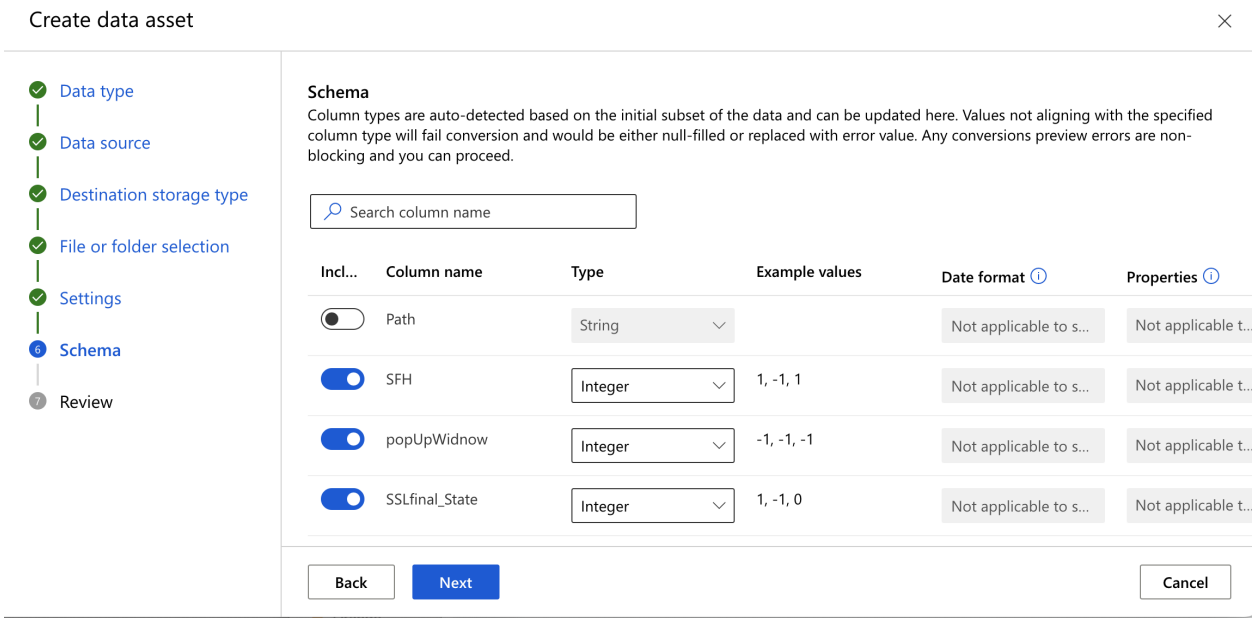
d. In the datastore, find the path to our csv file and upload file



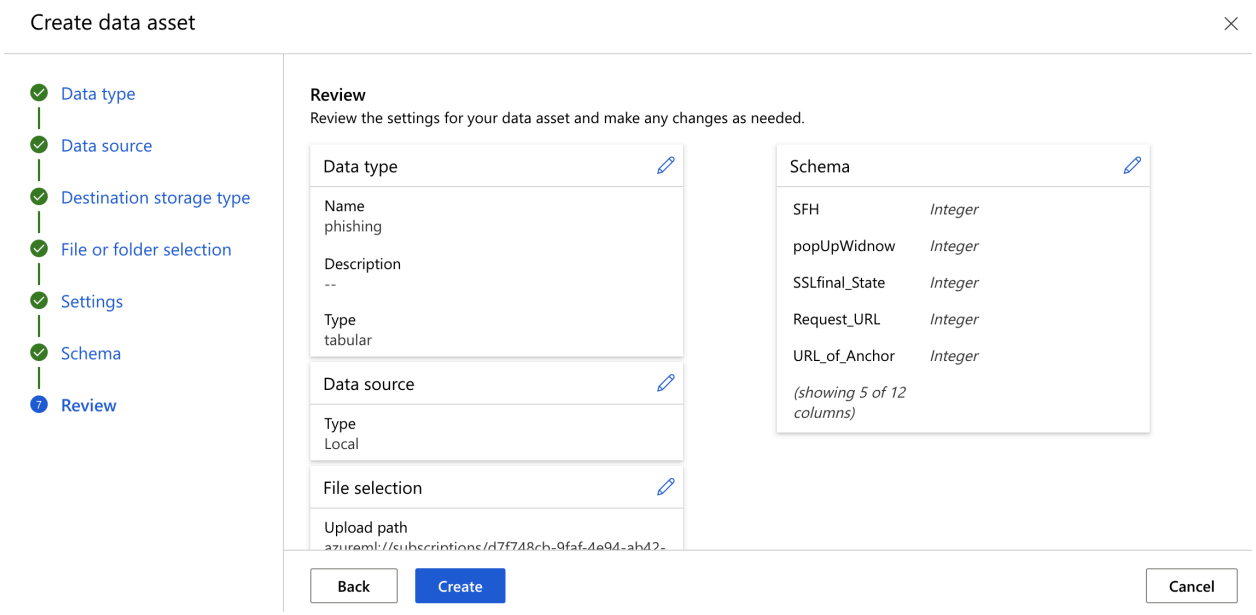
e. Config source file settings



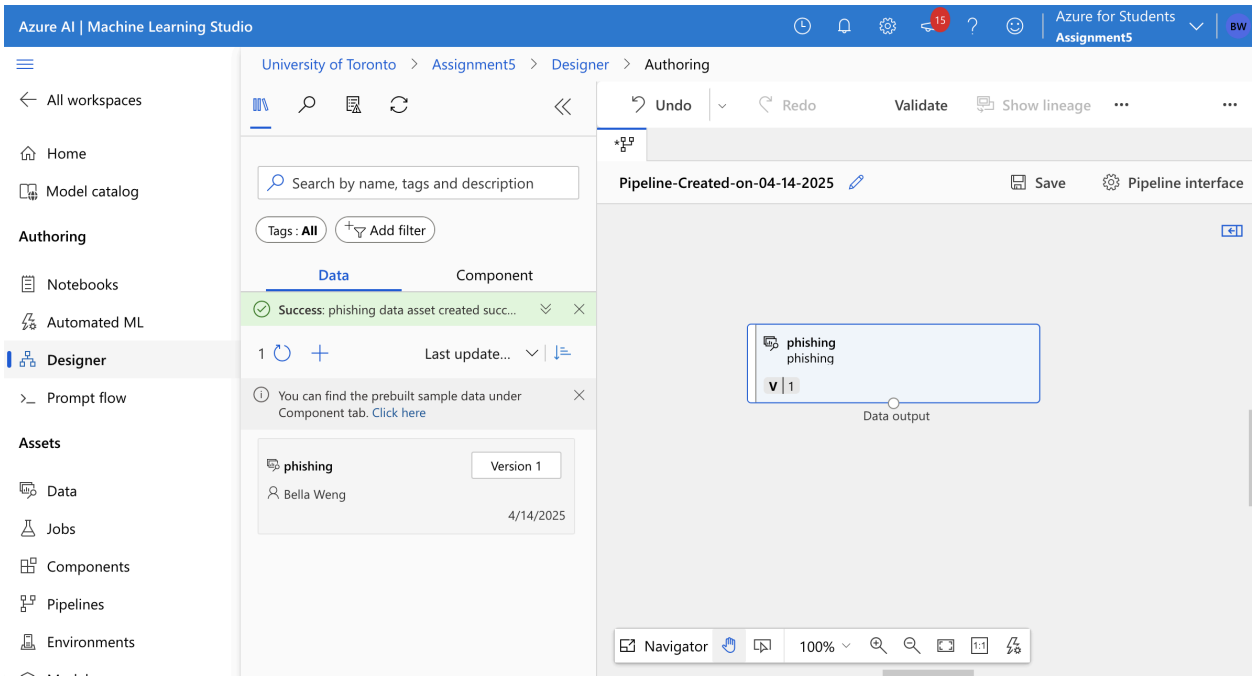
f. config table schema



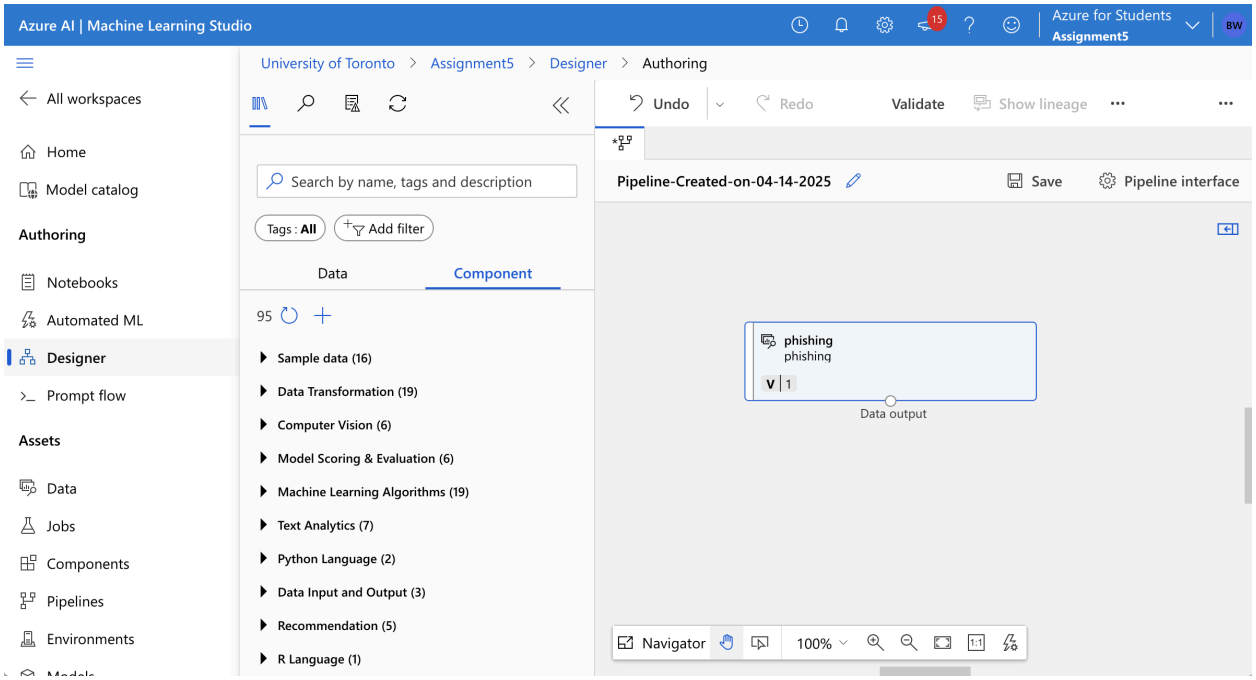
g. Now, review and submit



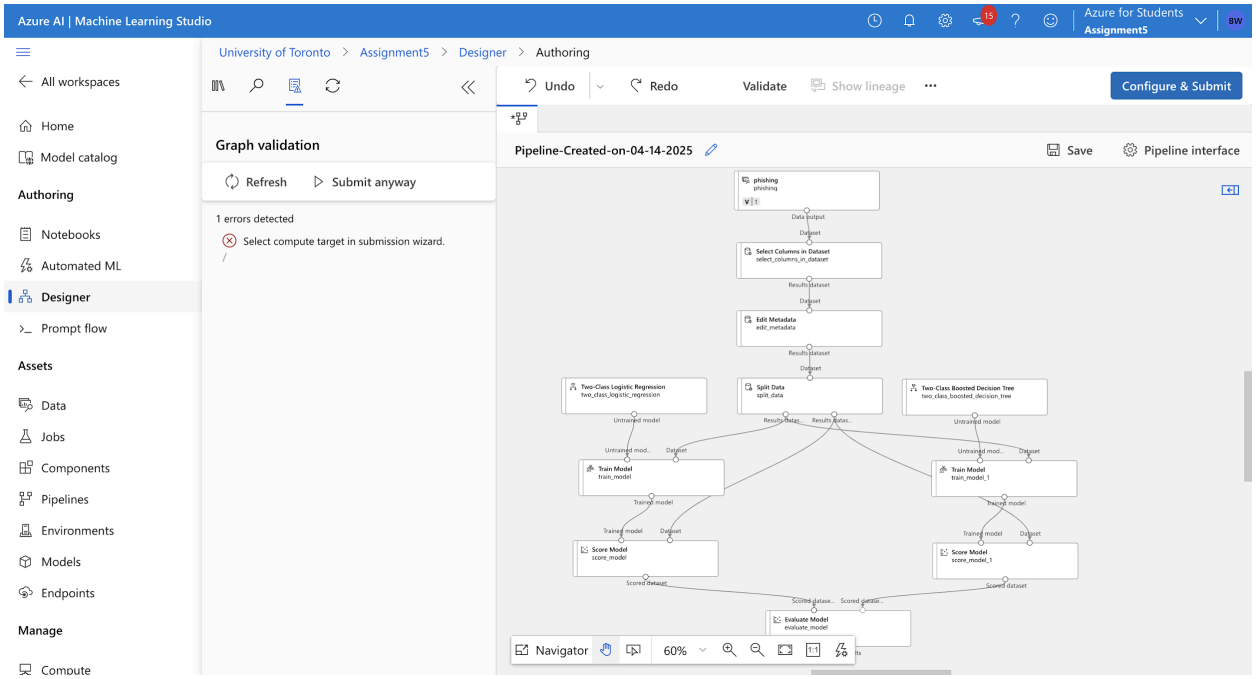
4. After adding an asset, you should be able to see it under the data tab. Drag it to our work space.



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

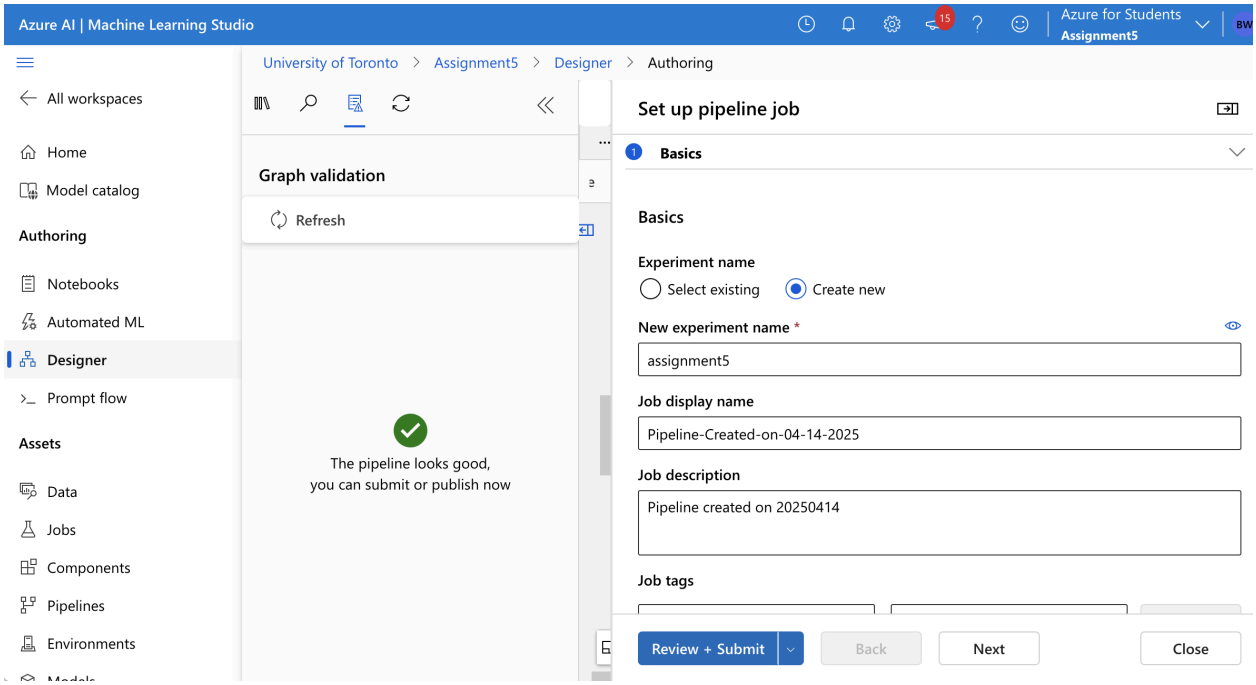


6. This Azure Machine Learning pipeline is built to classify phishing websites using two supervised models: Logistic Regression and Boosted Decision Tree. The process starts with importing the dataset, selecting relevant features, and editing metadata to define the target column. The data is then split 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.

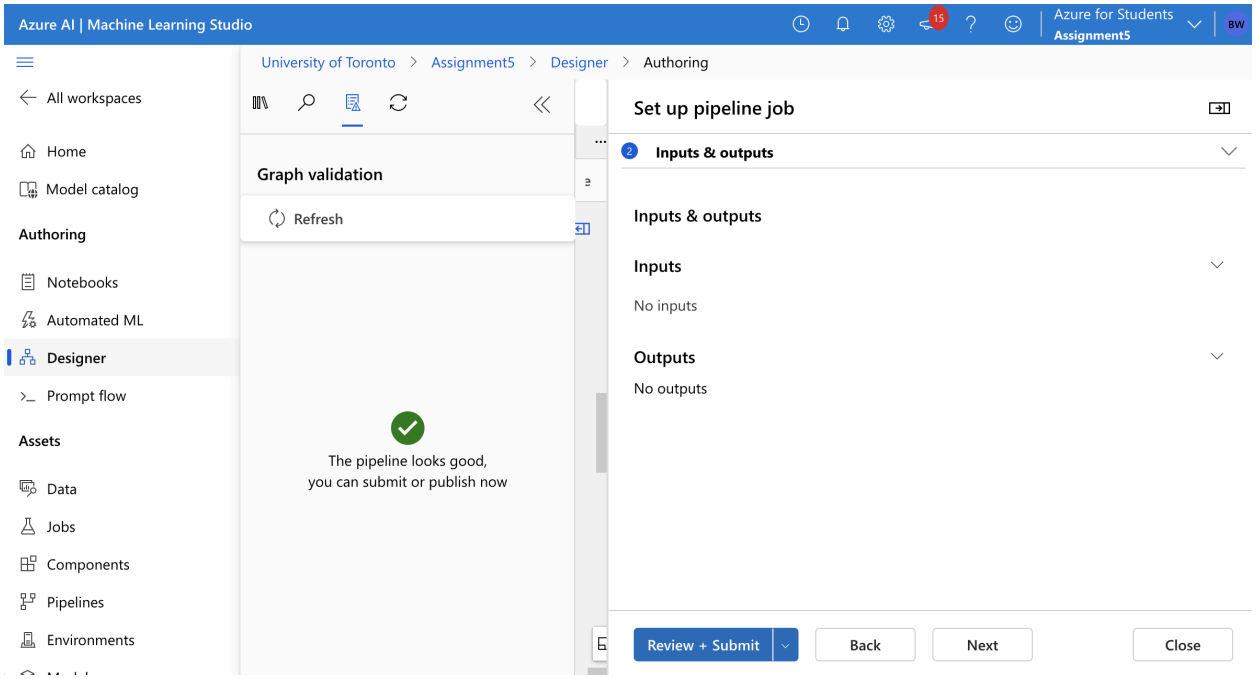


7. Now, we can config and submit our pipeline

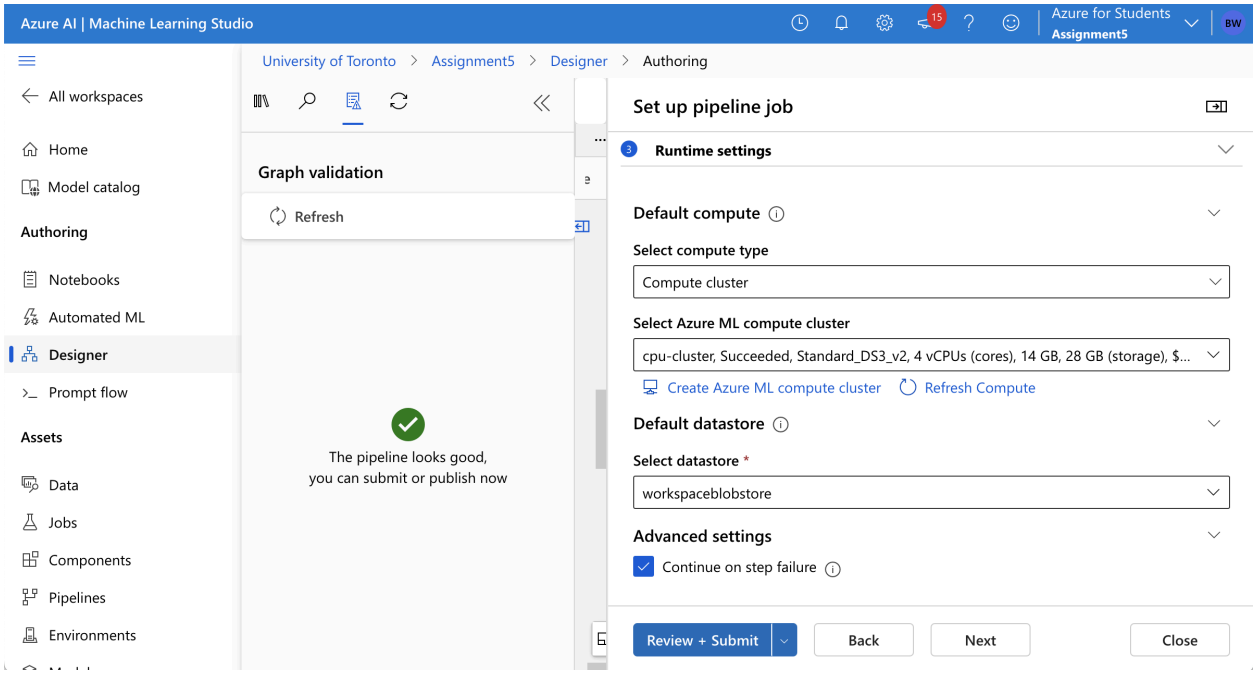
a. Basics



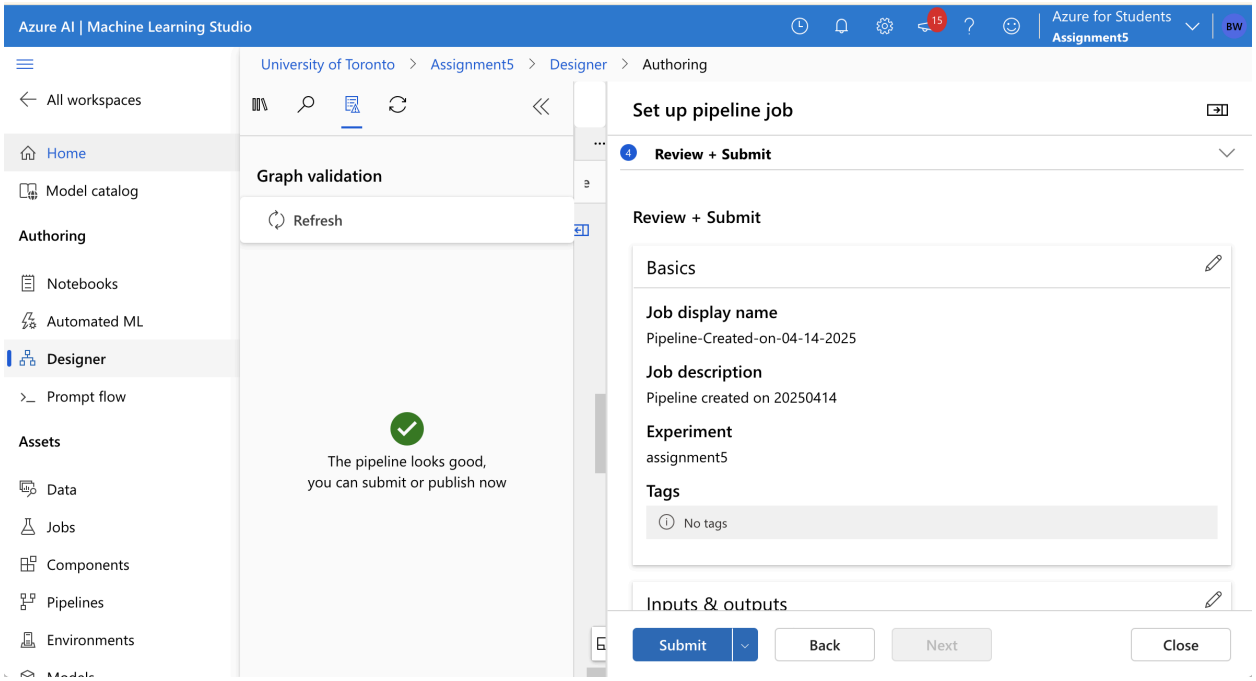
b. inputs and outputs



c. runtime settings

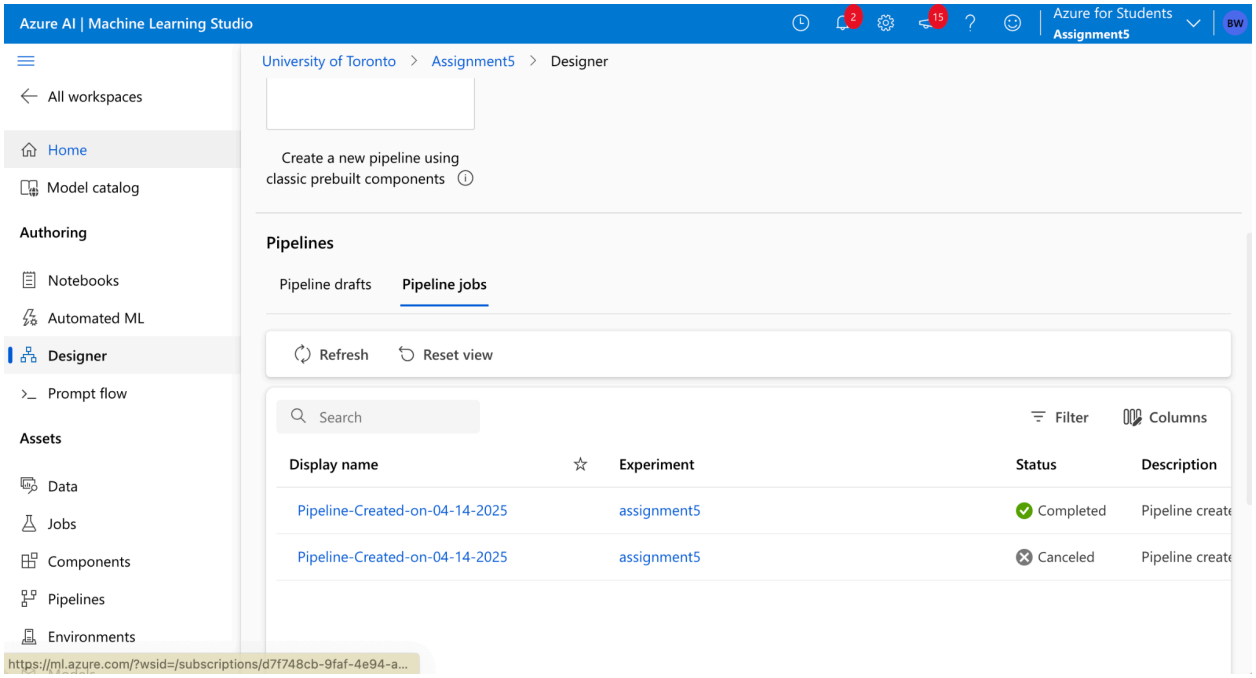


d. submit

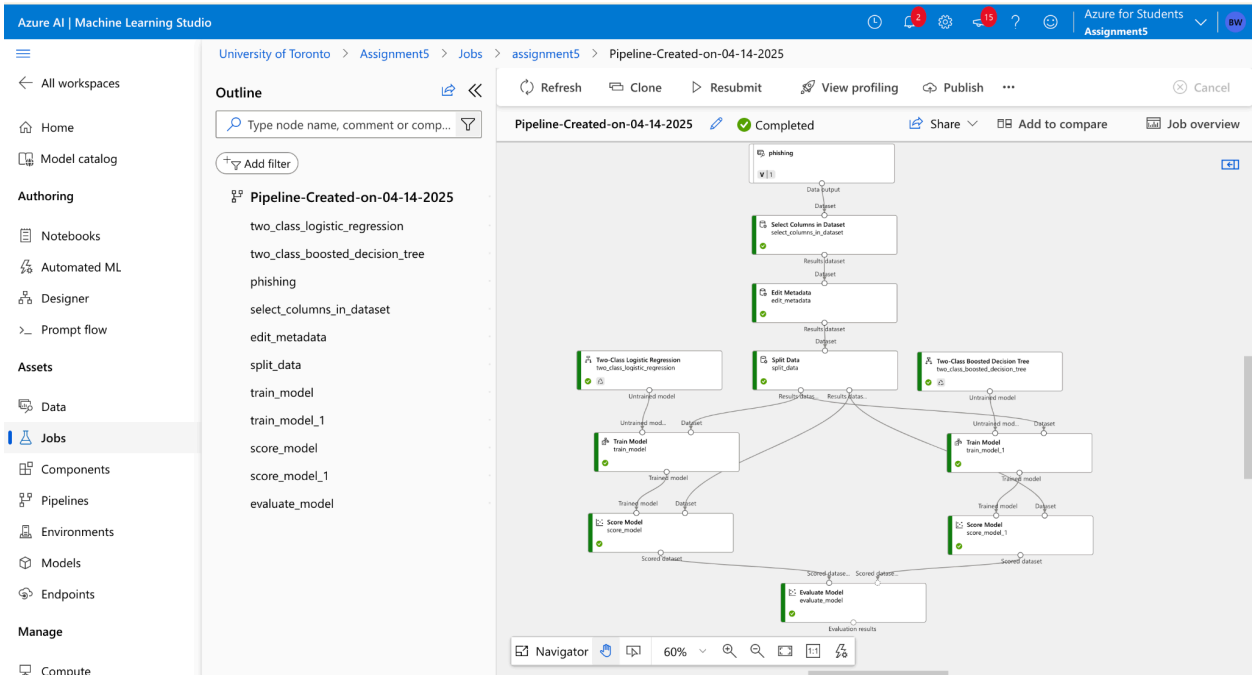


Outputs:

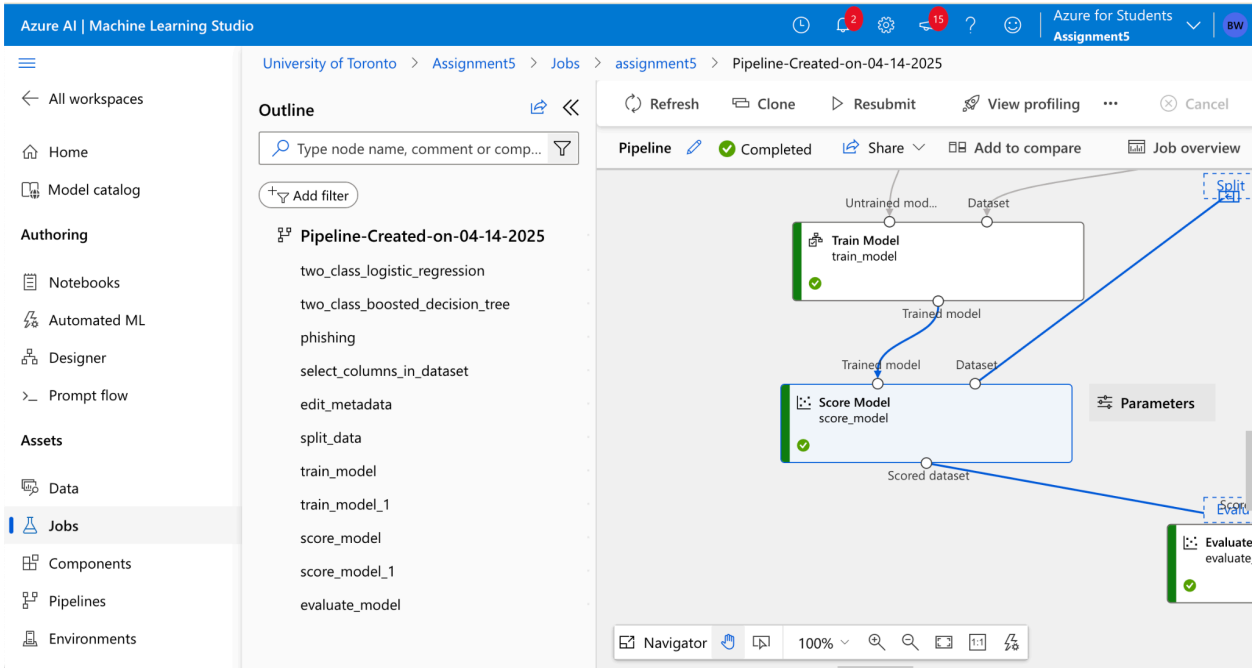
8. After we submit, we can go back to the Designer page. Under “Pipeline Jobs” we can see our new pipeline job.



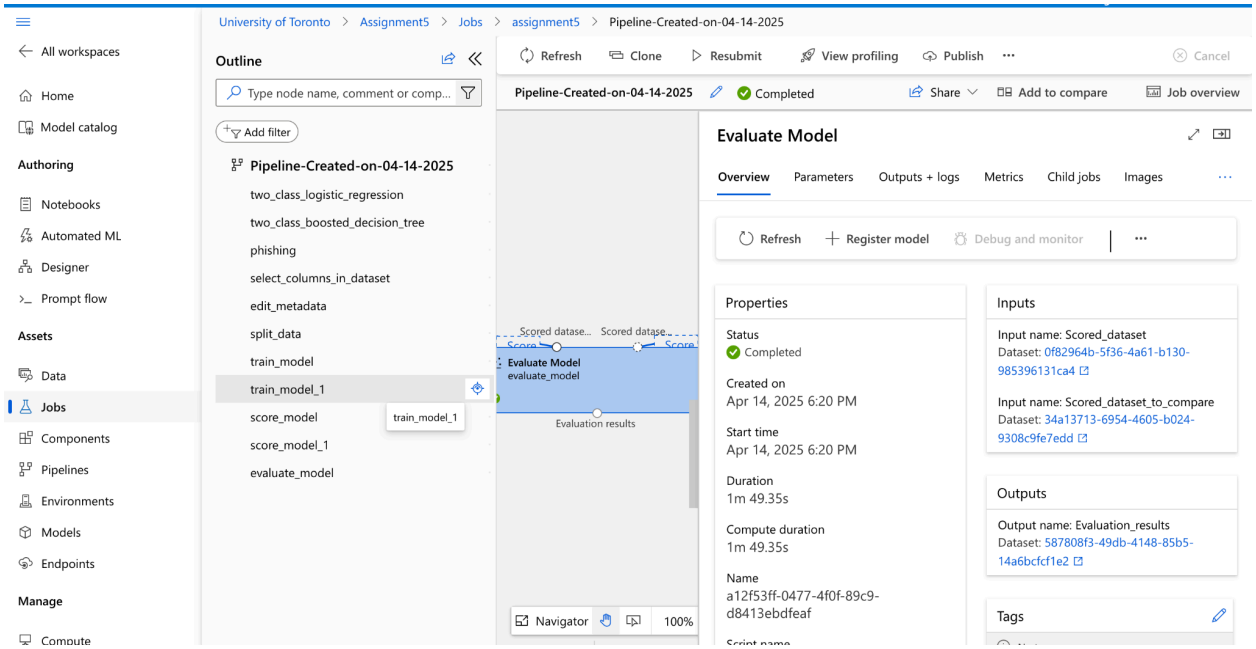
9. We can click the job to view the status of each component.



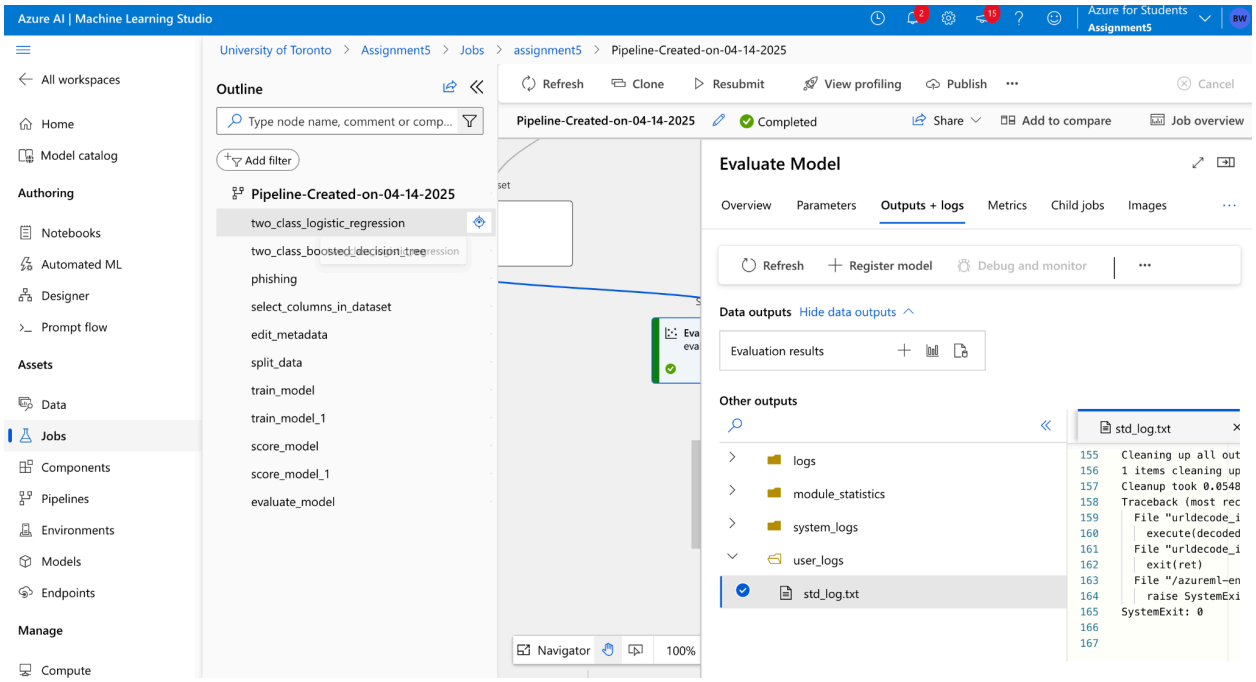
10. We can also select a component and click “view in canvas” for more details.



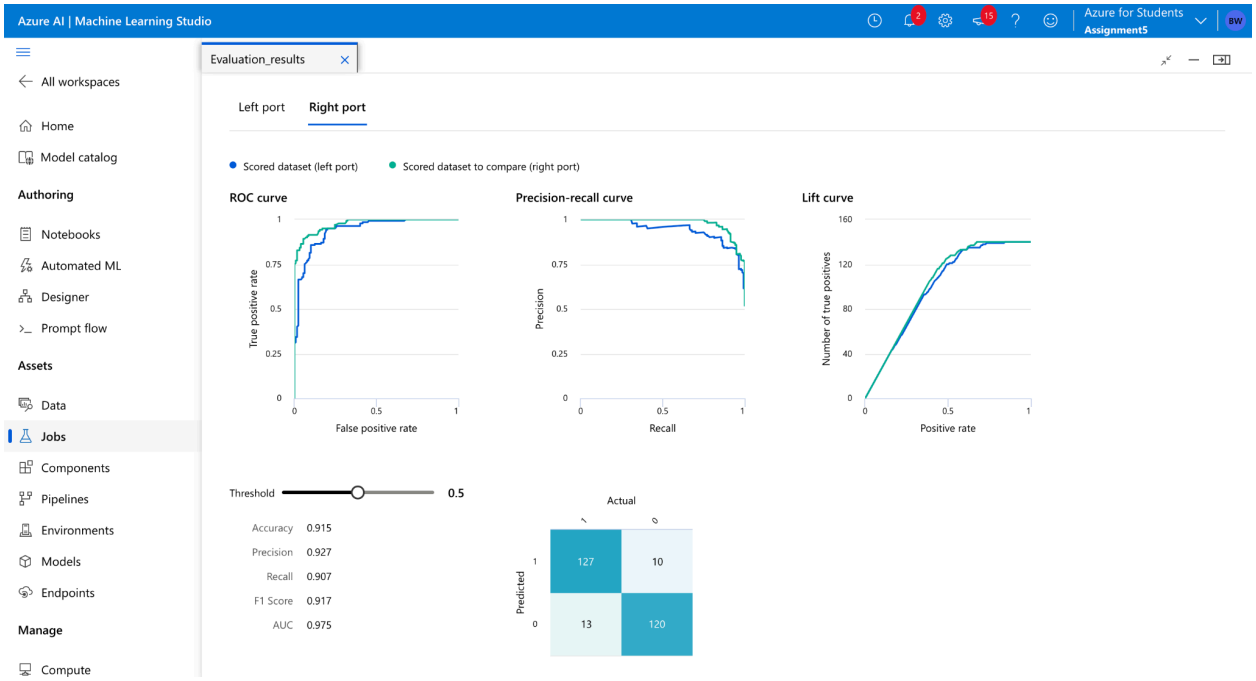
11. Evaluate Model details

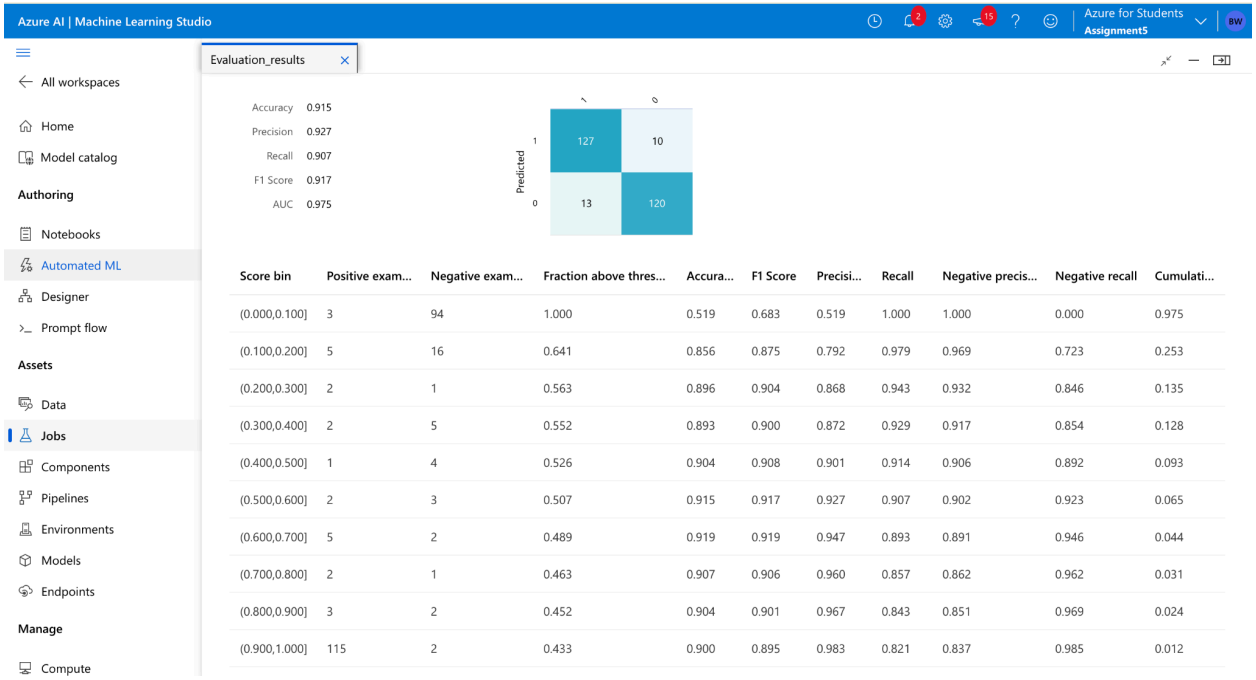


12. In the canvas, we can go to the “Outputs + logs” tab, click “Hide data outputs”, and click “Preview data” to view the result



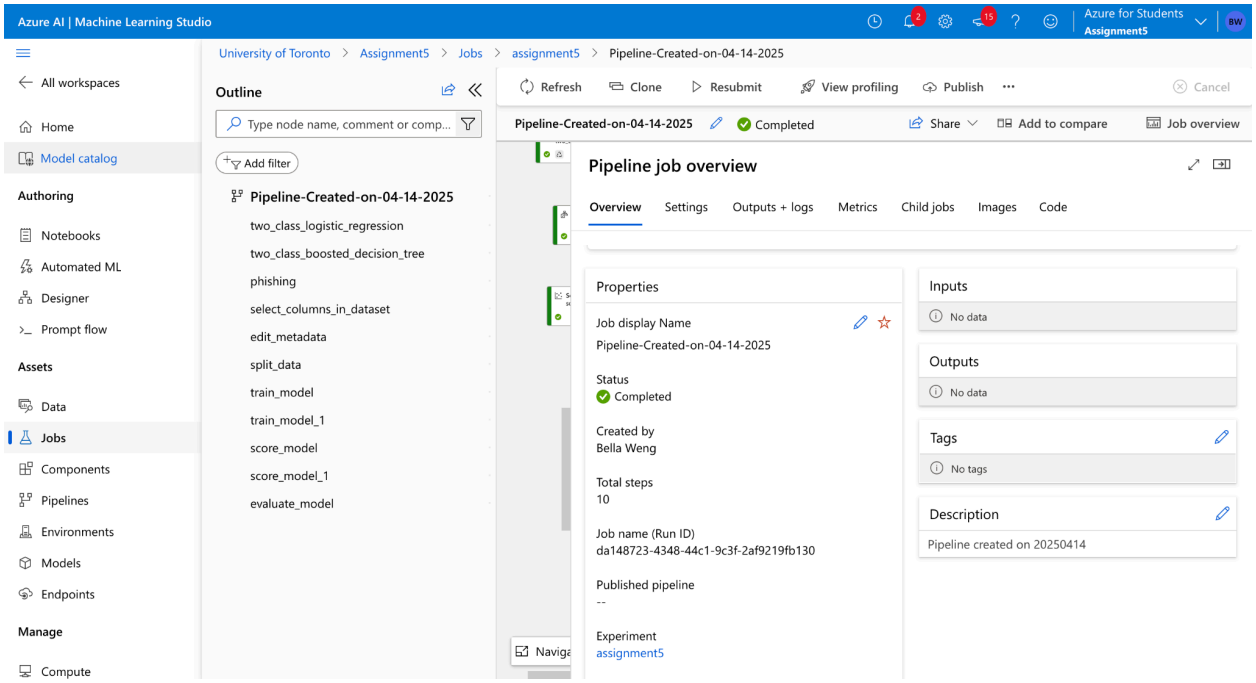
13. Here's the output

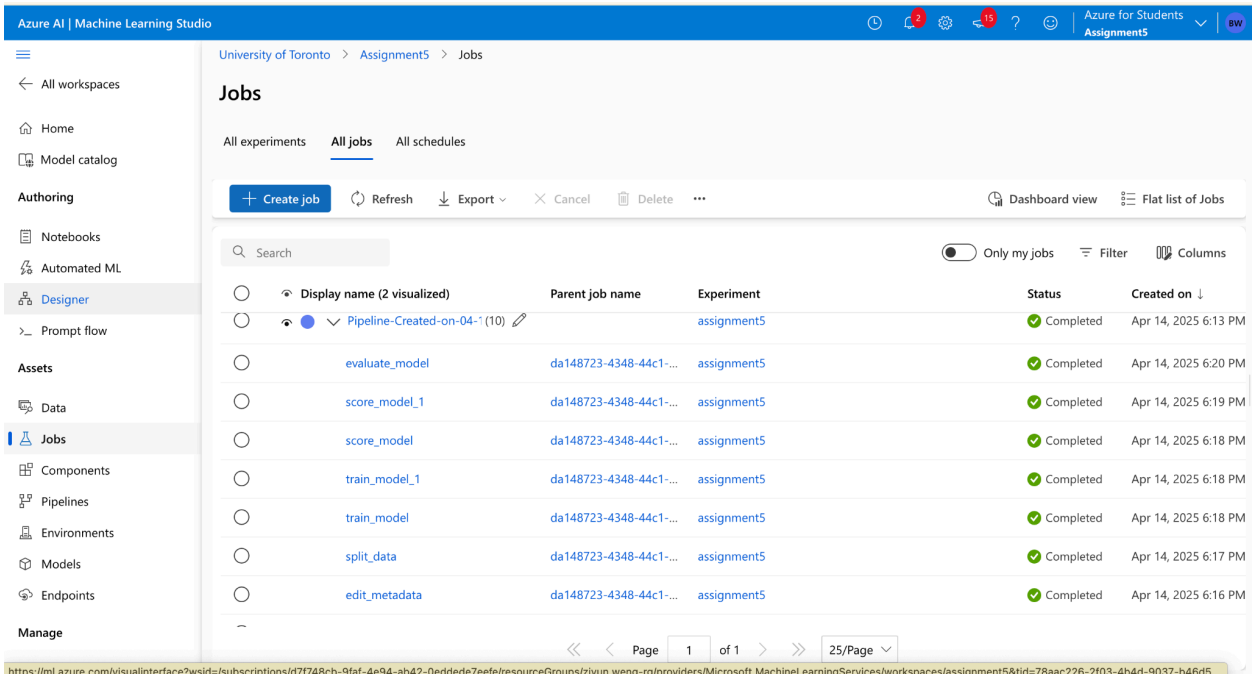




Score bin	Positive exam...	Negative exam...	Fraction above thres...	Accura...	F1 Score	Precisi...	Recall	Negative precis...	Negative recall	Cumulati...
(0.000,0.100]	3	94	1.000	0.519	0.683	0.519	1.000	1.000	0.000	0.975
(0.100,0.200]	5	16	0.641	0.856	0.875	0.792	0.979	0.969	0.723	0.253
(0.200,0.300]	2	1	0.563	0.896	0.904	0.868	0.943	0.932	0.846	0.135
(0.300,0.400]	2	5	0.552	0.893	0.900	0.872	0.929	0.917	0.854	0.128
(0.400,0.500]	1	4	0.526	0.904	0.908	0.901	0.914	0.906	0.892	0.093
(0.500,0.600]	2	3	0.507	0.915	0.917	0.927	0.907	0.902	0.923	0.065
(0.600,0.700]	5	2	0.489	0.919	0.919	0.947	0.893	0.891	0.946	0.044
(0.700,0.800]	2	1	0.463	0.907	0.906	0.960	0.857	0.862	0.962	0.031
(0.800,0.900]	3	2	0.452	0.904	0.901	0.967	0.843	0.851	0.969	0.024
(0.900,1.000]	115	2	0.433	0.900	0.895	0.983	0.821	0.837	0.985	0.012

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





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.