Anomaly Detection

Citizen Analytics – An Initiative by Data Science Team

START ▶

Learning Objectives

By the end of this module, you will be able to:

01 Understand the concept of outliers and familiarize with the available techniques in outlier detection.

02 Understand the concept of One-Class Support Vector Machine and perform it in Azure ML.

03 Understand the concept of PCA-Based Anomaly Detection and perform it in Azure ML.

Content

01. Outlier 04
a. Introduction to outlier detection
b. Multivariate anomaly detection techniques in Azure ML

02. One-Class Support Vector Machine (SVM) 07
a. One-Class Support Vector Machine
b. How to configure One-Class SVM in Azure ML
c. One-Class SVM in Azure ML

03. PCA-Based Anomaly Detection 23
a. PCA-Based Anomaly Detection
b. How to configure PCA-Based Anomaly Detection in Azure ML
c. PCA-Based Anomaly Detection in Azure ML

04. Summary 40

Outlier

When to use: Anomaly Detection

What is an anomaly?

- An observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism (Hawkins, 1980)

What type of Questions

- Is the Variable X abnormal? (Univariate Anomaly Detection)
- How different is this datapoint from rest of the datapoints? (Multivariate Anomaly Detection)

Assumptions

1. Anomalies only occur very rarely in the data
2. Their features differ from the normal instances significantly

What type of data

The input data should be continuous or a combination of both continuous and categorical

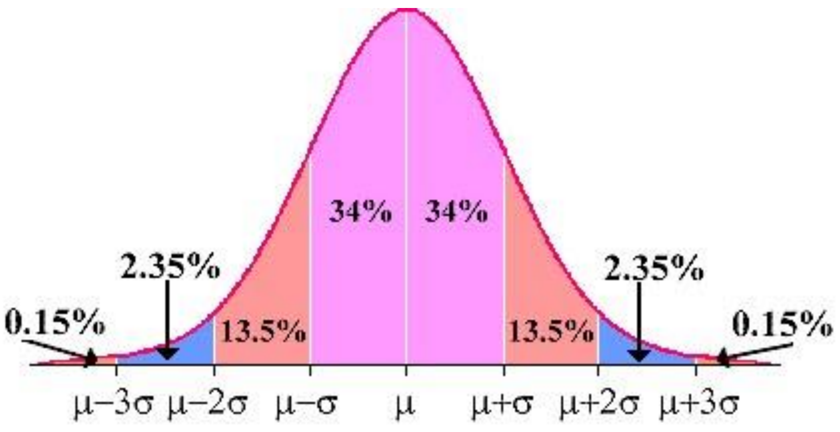
Use Cases

- **Engineering Domain** : Production plant applications – Is this gas pressure unusual?
- **Finance Domain:** Fraud Detection – Is this transaction normal?
- **Retail Domain:** E-commerce applications – Is this combination of purchases very different from what this customer has made previously?

PETRONAS

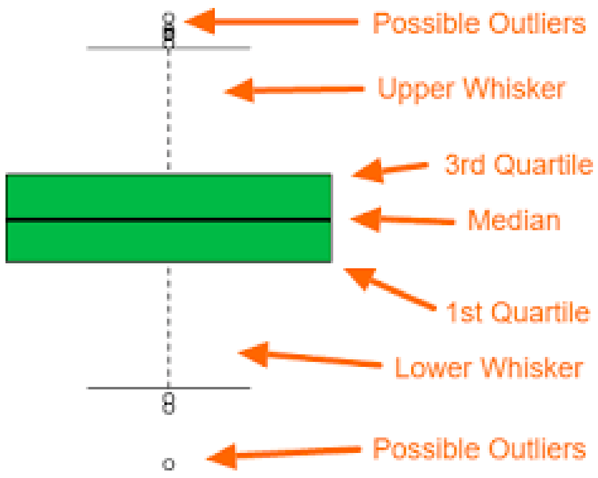
Univariate Anomaly Detection (Anomaly Detection on one variable at a time)

Z-Score Method



- Z-score can be calculated as Z = (x – μ) / δ, where μ is mean and δ is the standard deviation of the variable
- When computing the z-score for each sample on the data set a threshold must be specified.
- Some good 'thumb-rule' thresholds can be: 2.5, 3, 3.5 or more standard deviations.

Tukey's Method



- Inner "fences" are located at a distance of 1.5 IQR below Q1 and above Q3, and outer fences at a distance of 3 IQR below Q1 and above Q3.
- A value between the inner and outer fences is a possible outlier, whereas a value falling outside the outer fences is a probable outlier.

PETRONAS

Internal

Multivariate Anomaly Detection

Introduction

- Most of the analysis in anomalies that we end up doing are multivariate due to complexity of the world we are living in.
- In multivariate anomaly detection, outlier is a combined unusual score on at least two variables.

Intuition

- Multivariate approaches detect anomalies as complete incidents instead of individual variables.
- This approach also produces anomaly alerts.
- These are hard to interpret because all the metrics are inputs that generate a single output from the anomaly detection system.

Multivariate Anomaly detection techniques in Azure ML

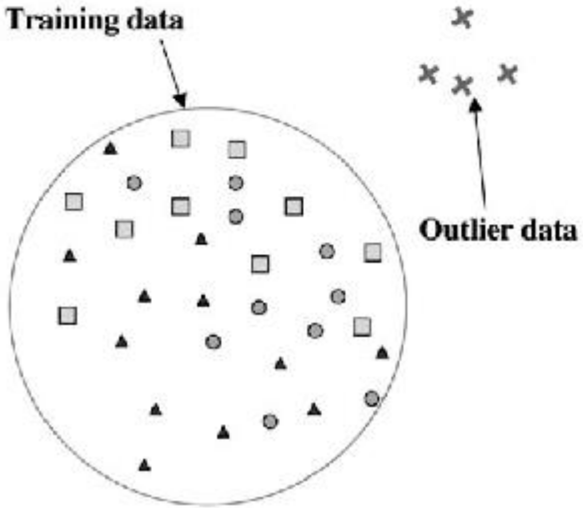
Multivariate Anomaly Detection Techniques

- One-Class Support Vector Machine
- PCA-Based Anomaly Detection

One-Class
Support Vector Machine

PETRONAS

One-Class Support Vector Machine



- Support vector machines (SVMs) are supervised learning models that analyze data and recognize patterns, and that can be used for both classification and regression tasks.
- An SVM model is based on dividing the training sample points into separate categories by as wide a gap as possible, while penalizing training samples that fall on the wrong side of the gap.
- Therefore, in one-class SVM, the support vector model is trained on the data that has only one class, which is the 'normal' class. It infers the properties of the normal cases and from these properties can predict which examples are unlike the normal examples.
- This is for anomaly detection because the scarcity of training examples is what defines anomalies; typically there are very few examples of network intrusion, fraud, or other anomalous behavior

How to configure One-Class SVM in Azure ML

1. Add the One-Class Support Vector Model

You can find the module under **Machine Learning - Initialize**, in the **Anomaly Detection** category.

2. Open the Properties Pane

Double-click the **One-Class Support Vector Model** module to open the **Properties** pane

3. For **Create trainer mode**, select an option that indicates how the model should be trained

Single Parameter: Use this option if you know how you want to configure the model and provide a specific set of values as arguments.
Parameter Range: Use this option if you are not sure of the best parameters and want to perform a parameter sweep to find the optimal configuration.

4. Select the value for η

This value represents the upper bound on the fraction of outliers. The nu-property, η lets you control the trade-off between outliers and normal cases.

5. Select the value for ε (epsilon)

This value is used as the stopping tolerance. The stopping tolerance, affects the number of iterations used when optimizing the model, and depends on the stopping criterion value. When the value is exceeded, the trainer stops iterating on a solution.

6. Connect a training dataset and set the value for one of the training modules

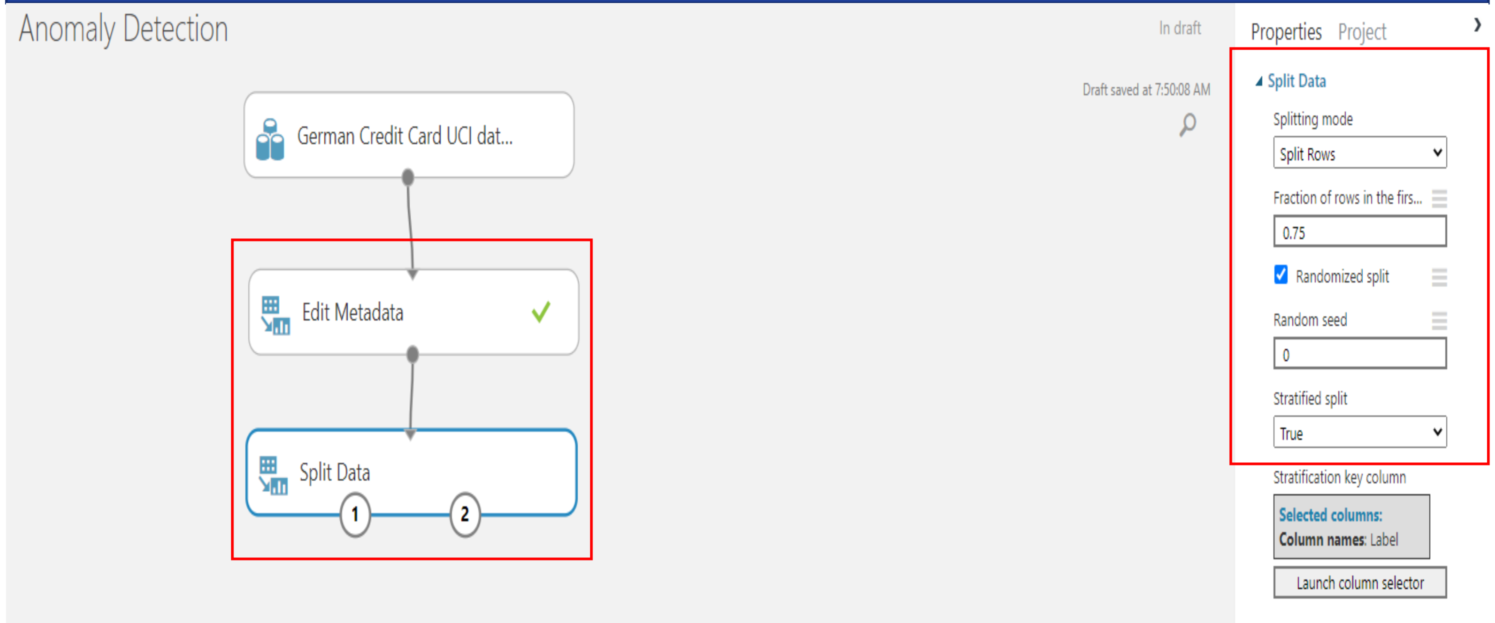
Connect a training dataset, and one of the training modules:
1. If you set Create trainer mode to Single Parameter, use the Train Anomaly Detection Model module.
2. If you set Create trainer mode to Parameter Range, use the Tune Model Hyperparameters module.

7. Run the experiment

One-Class SVM in Azure ML

1. Split the data into 75:25 ratio by adding the 'Split Data' module
2. Connect the 'Split Data' module to 'Edit Metadata' module

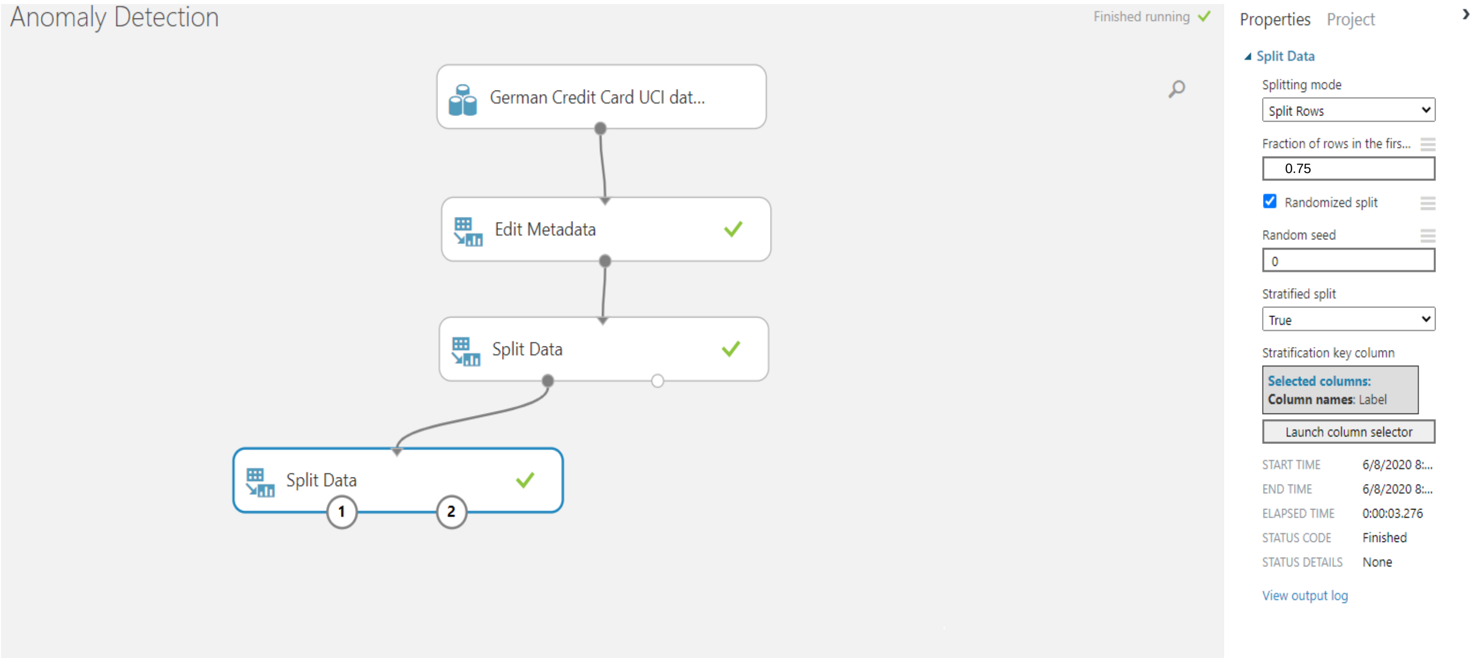
Splitting the data



One-Class SVM in Azure ML

3. Add the ‘Split Data’ module to the existing ‘Split Data’ module
4. Split the data into 75:25 ratio

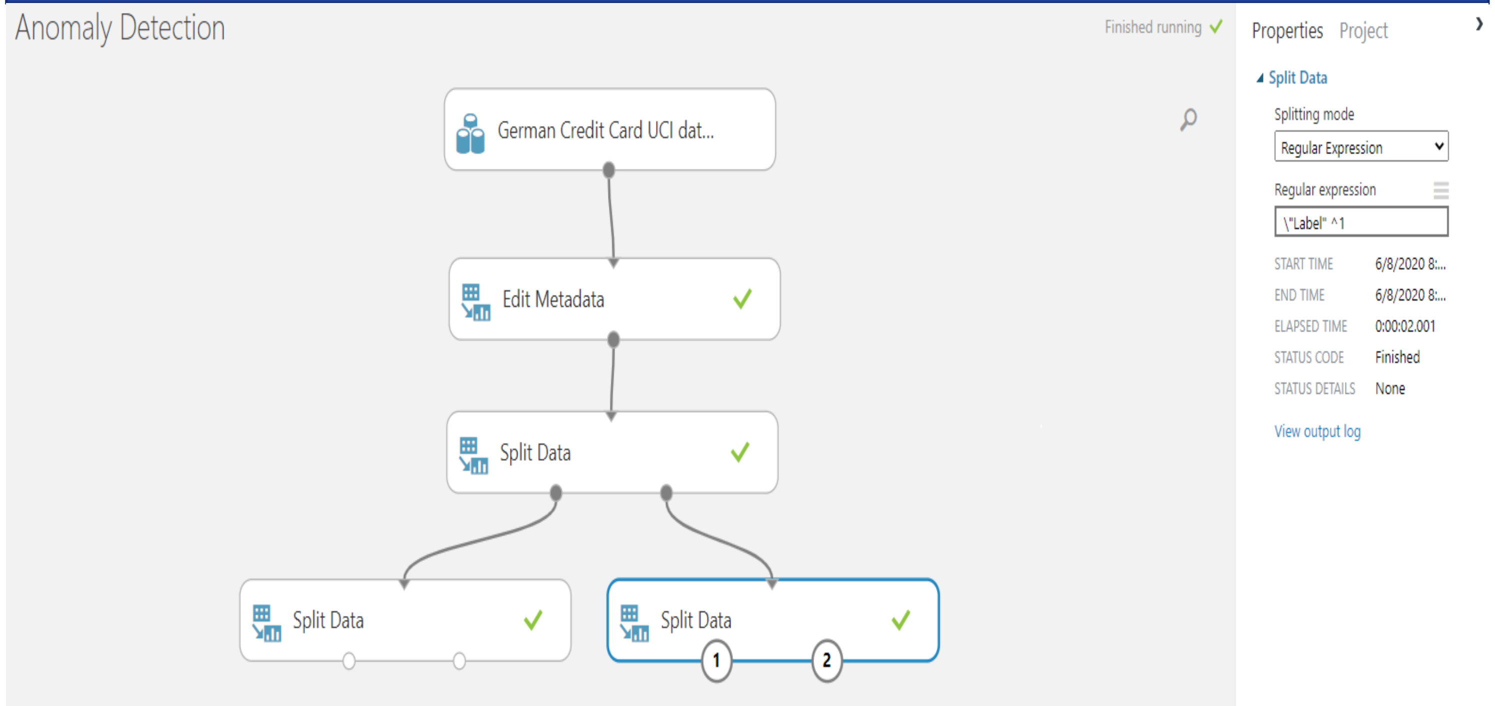
Splitting the data



One-Class SVM in Azure ML

5. Add the ‘Split Data’ module to the first ‘Split Data’ module
6. Split the data into normal observations where label is equal to 1

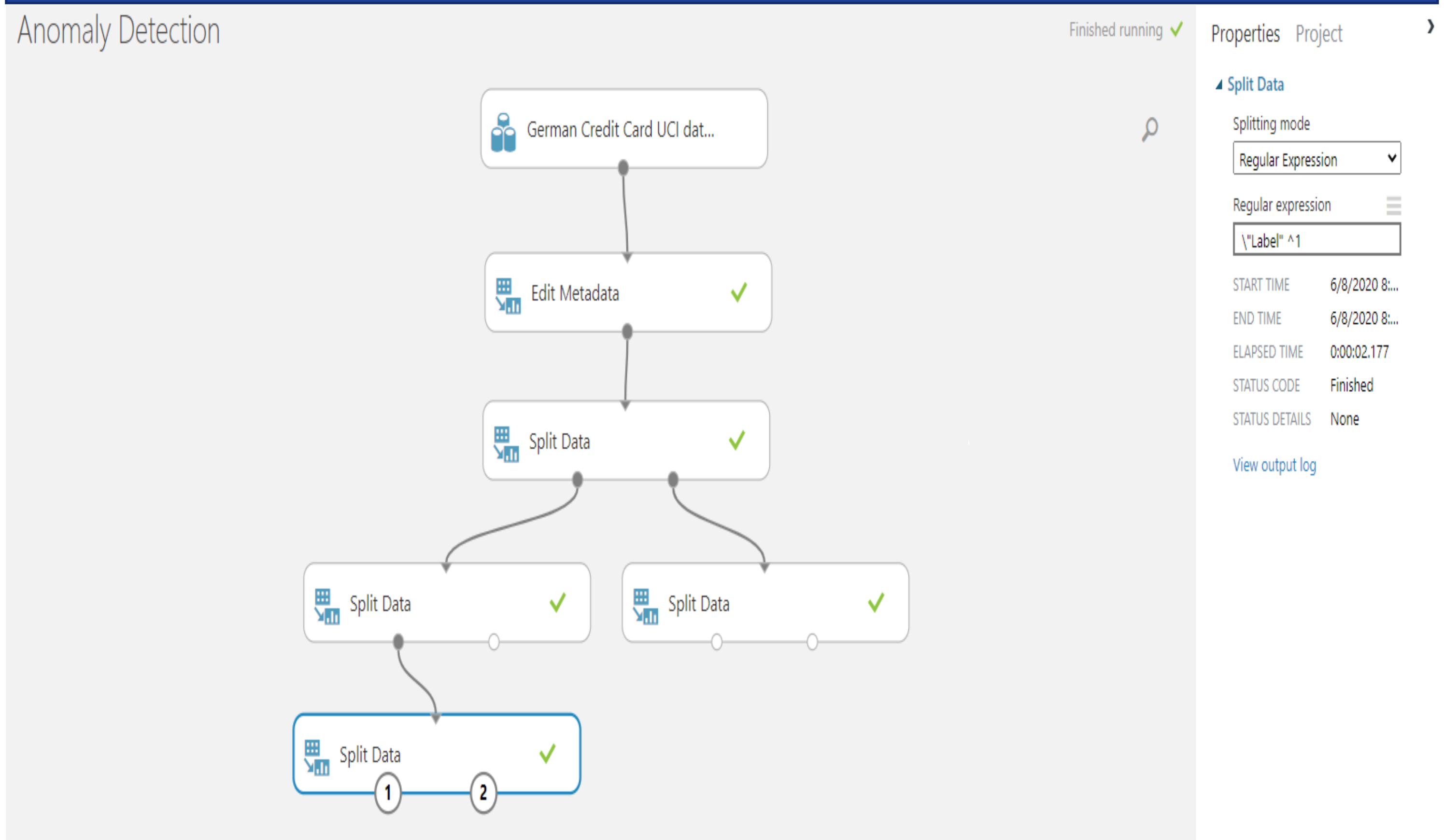
Splitting the data into normal observations



One-Class SVM in Azure ML

7. Add the ‘Split Data’ module to the second ‘Split Data’ module
8. Split the data into normal observations where label is equal to 1

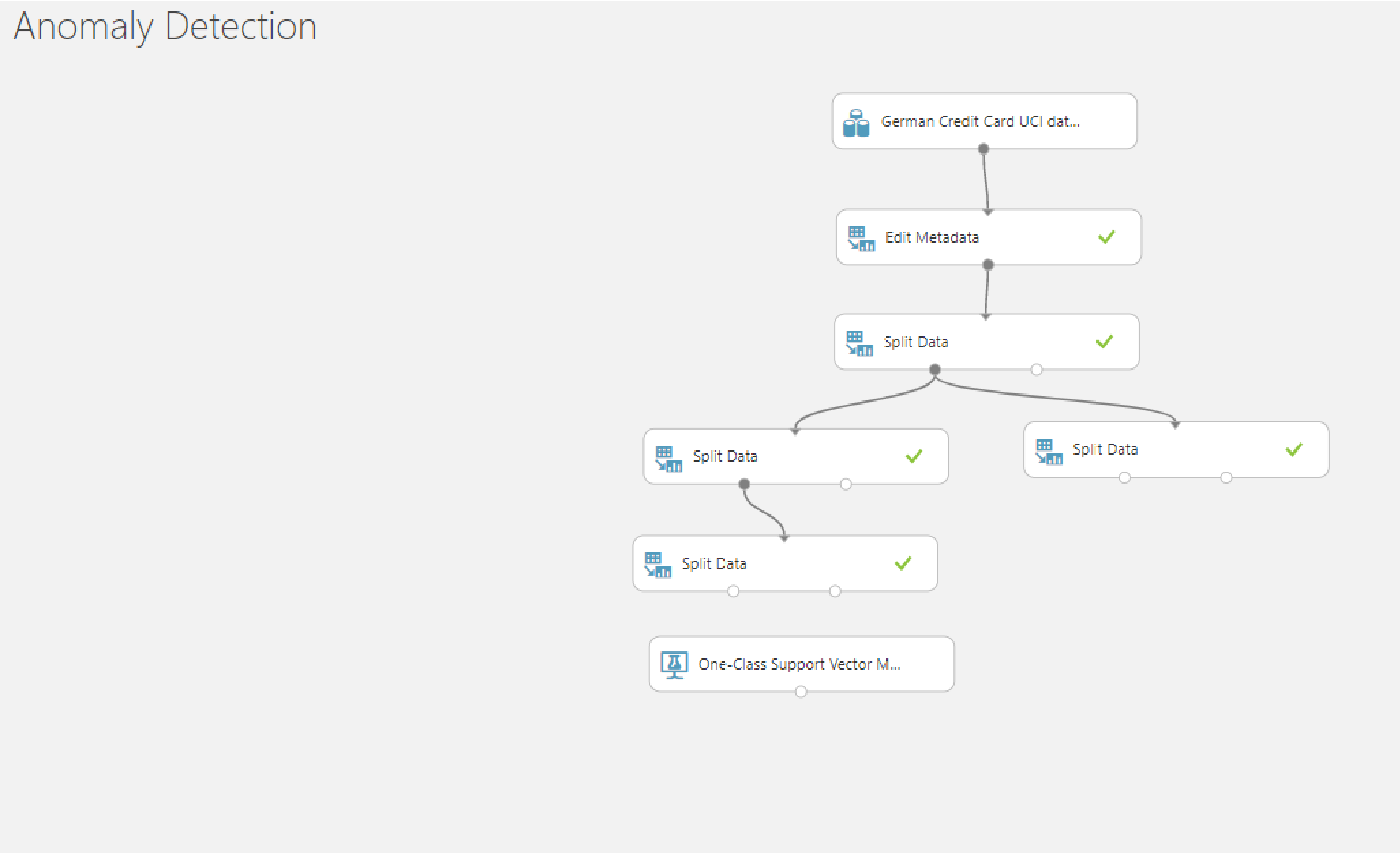
Splitting the data into normal observations



One-Class SVM in Azure ML

9. Add the ‘One Class Support Vector Machine’ module

Adding the One Class SVM module



PETRONAS

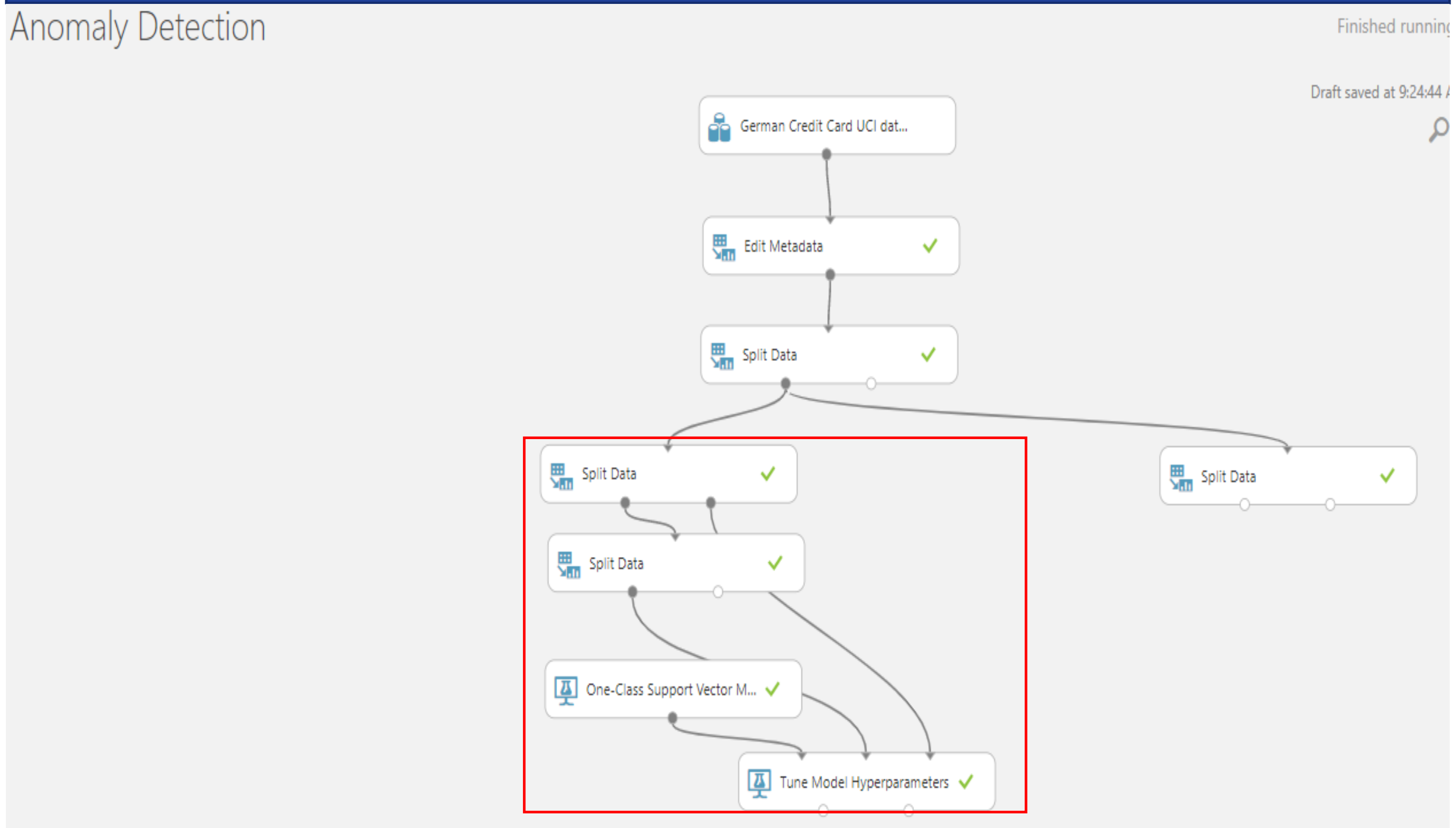
Internal

One-Class SVM in Azure ML

10. Add the ‘Tune Model Hyperparameters’ module

11. Connect it to the ‘One-Class Support Vector Machine’ module and ‘Split Data’ modules as highlighted

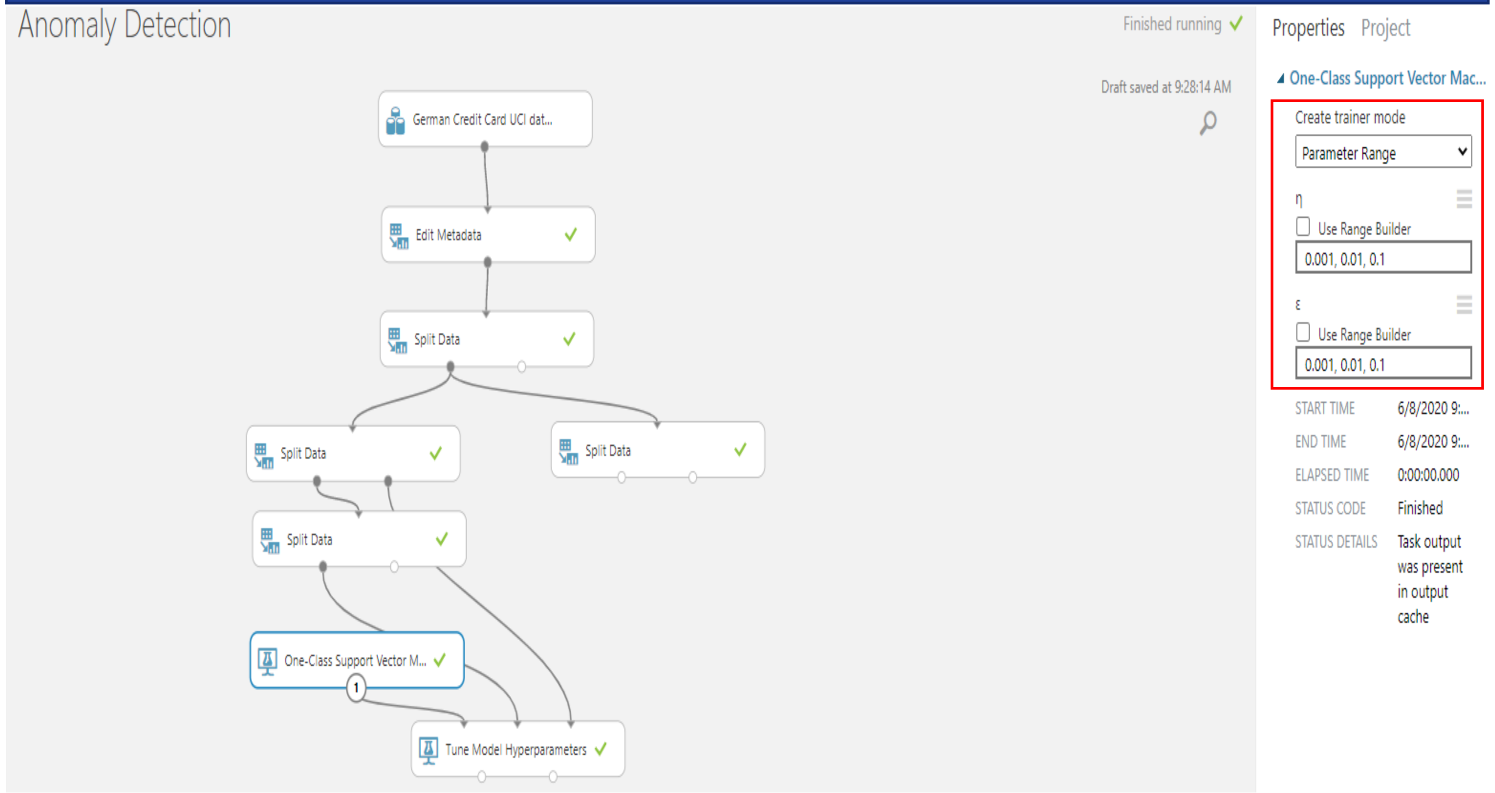
Adding the Tune Model Hyperparameters module



One-Class SVM in Azure ML

12. Select ‘One Class Support Vector Machine’ module and select the below parameters.
    - Training Mode – Parameter Range

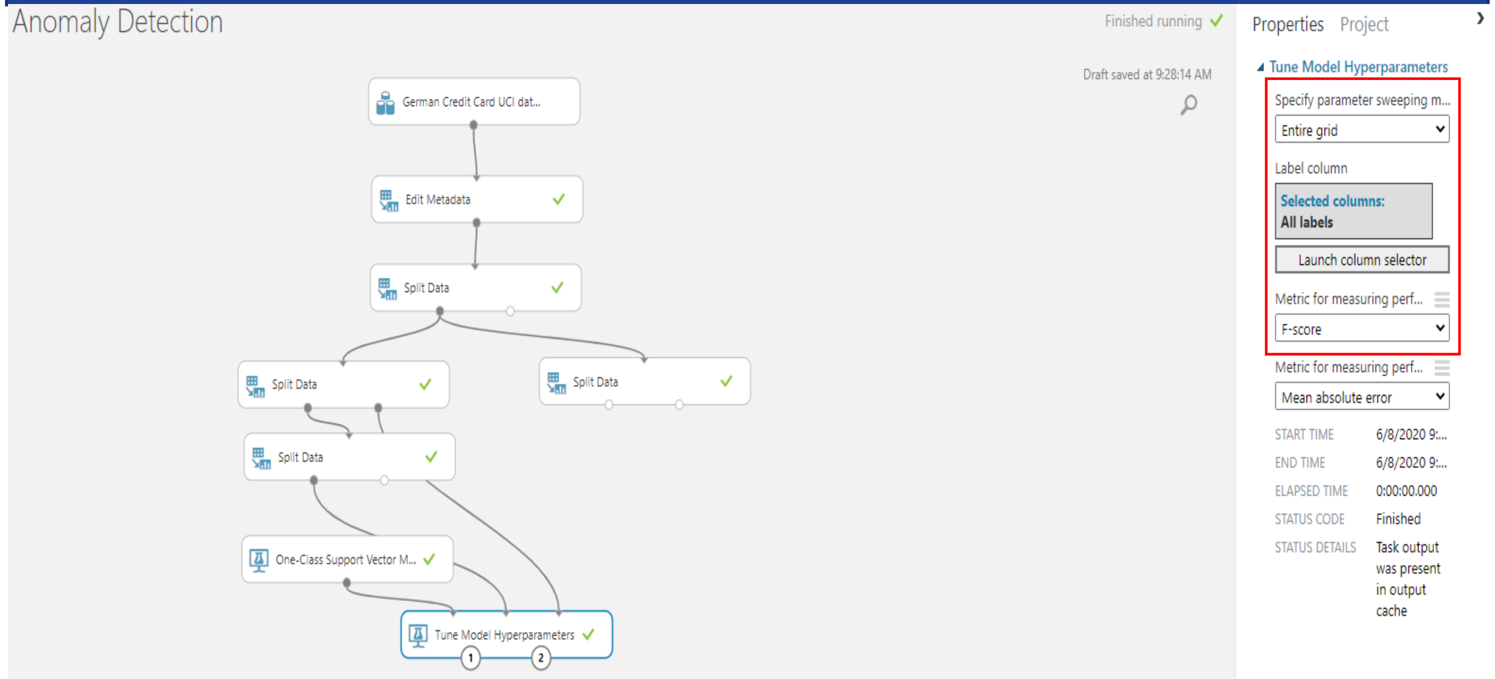
Selecting the Parameters for One Class Support Vector Machine Module



One-Class SVM in Azure ML

13. Select 'Tune Model Hyperparameters' module and select the below parameters.
    - Specify Parameter Sweeping Module – Entire Grid
    - Include all labels from 'Launch Column Selector'
    - Metric for Measuring Performance – 'F Score'

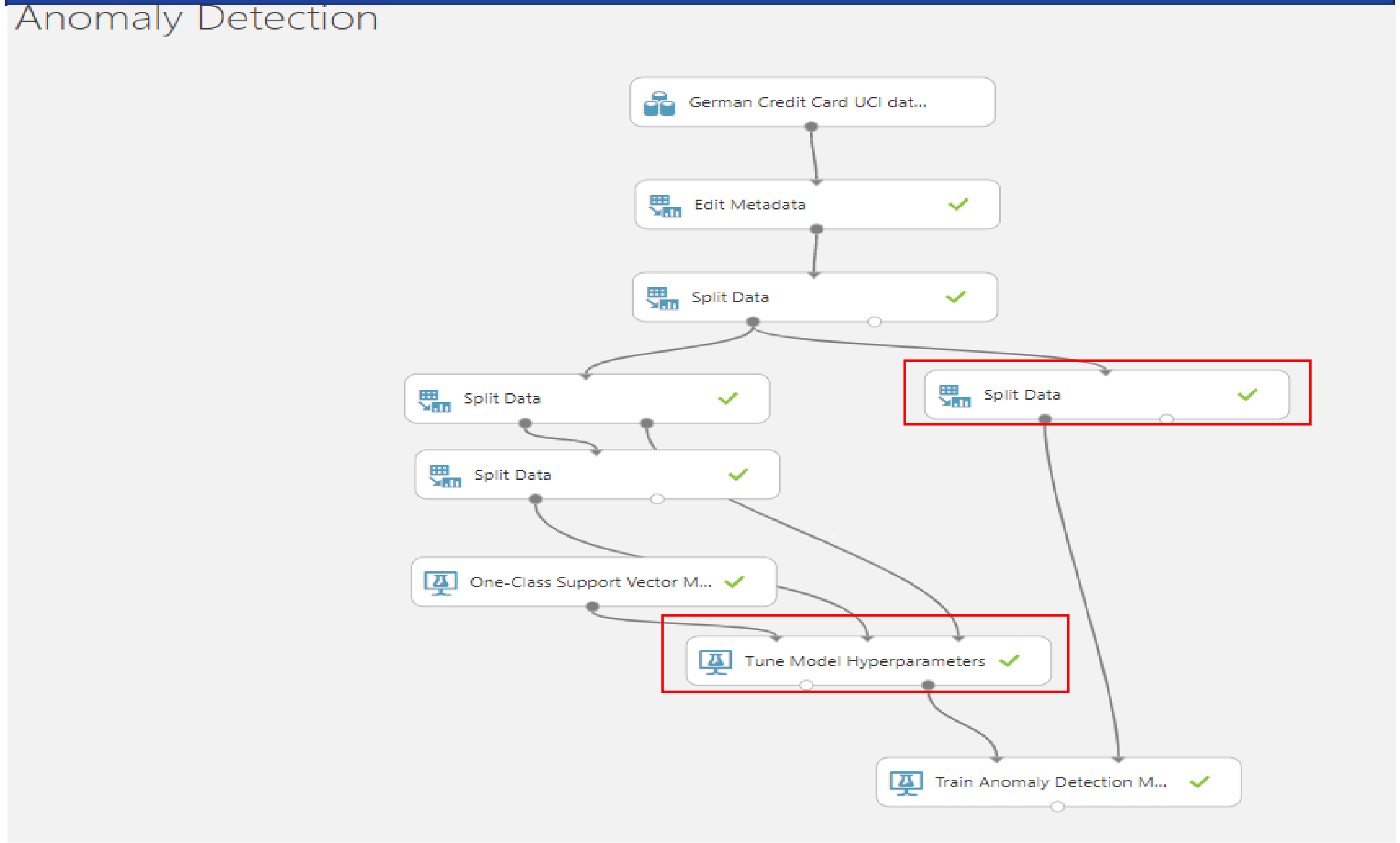
Selecting the Parameters for Tune Model Hyperparameters Module



One-Class SVM in Azure ML

14. Add the ‘Train Anomaly Detection Model’ module and connect it to the ‘Tune Model Hyperparameters’ and ‘Split Data’ modules as highlighted

Adding Train Anomaly Detection Model module



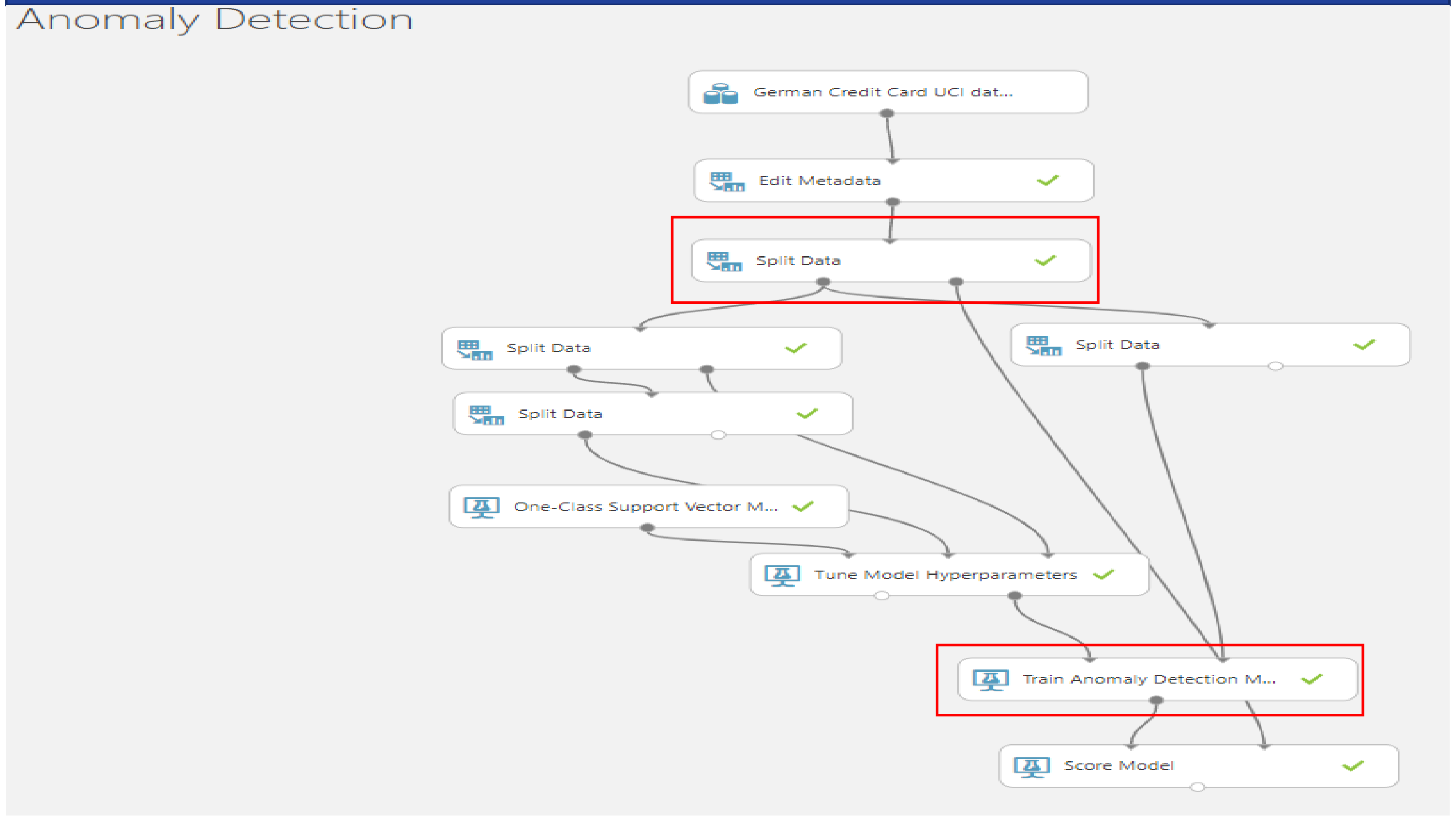
PETRONAS

Internal

One-Class SVM in Azure ML

15. Add the ‘Score Model’ module and connect it to the ‘Train Anomaly Detection Model’ and ‘Split Data’ modules as highlighted

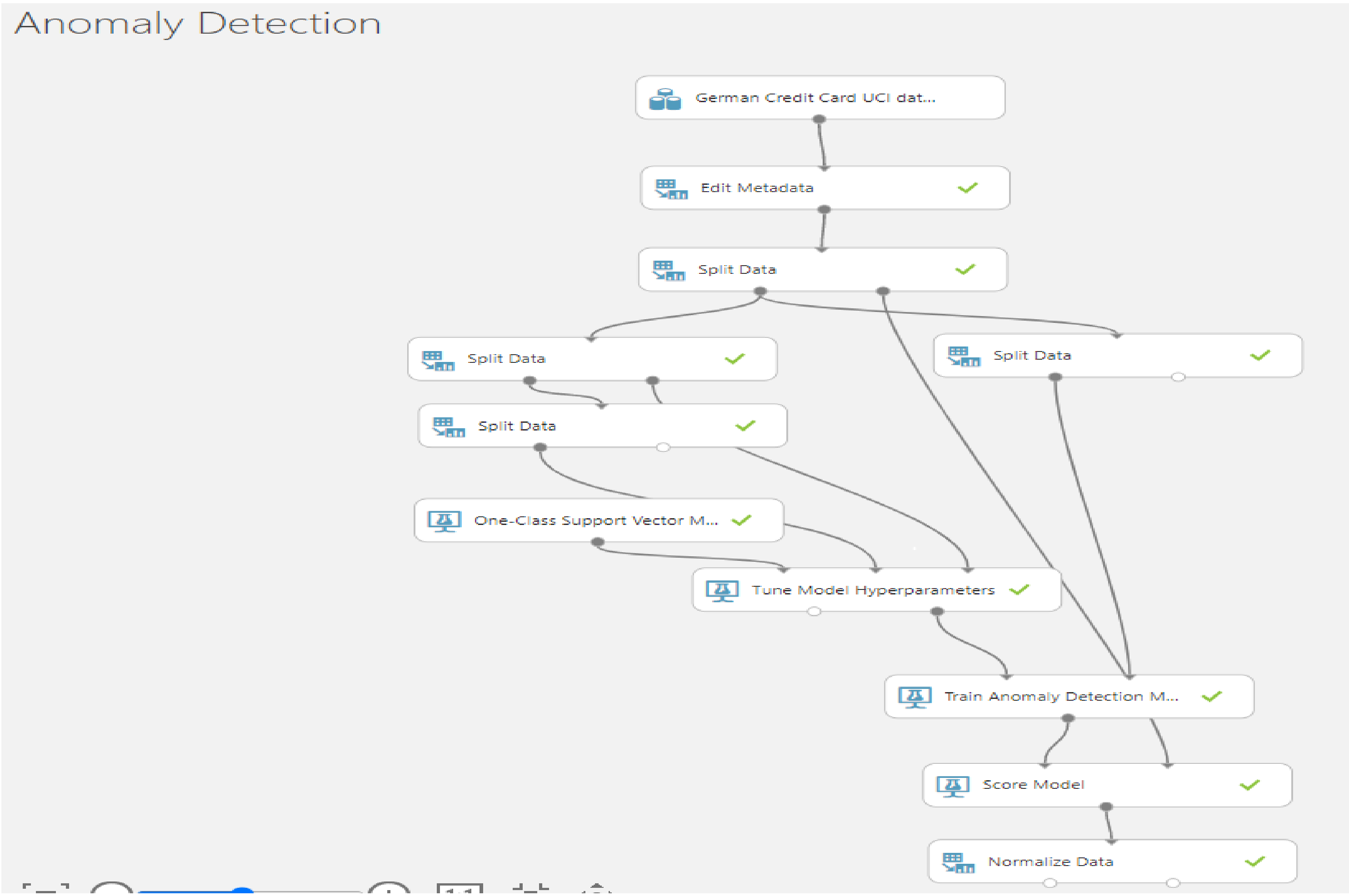
Adding Score Model module



One-Class SVM in Azure ML

16. Add the ‘Normalize Data’ module to the ‘Score Model’ module

Adding Normalize Data module

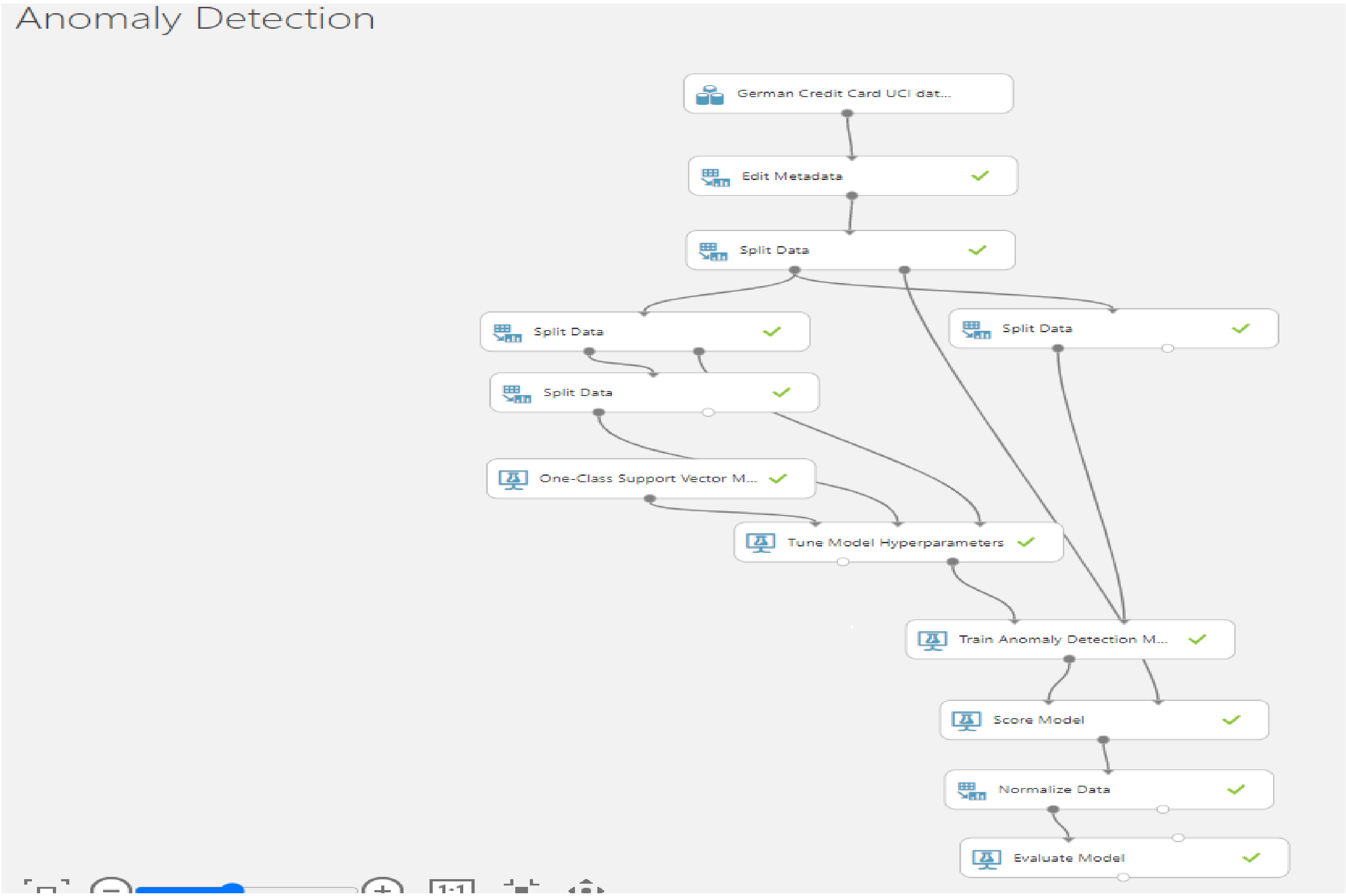


One-Class SVM in Azure ML

17. Add the ‘Evaluate Model’ module to the ‘Normalize Data’ module

18. RUN the experiment

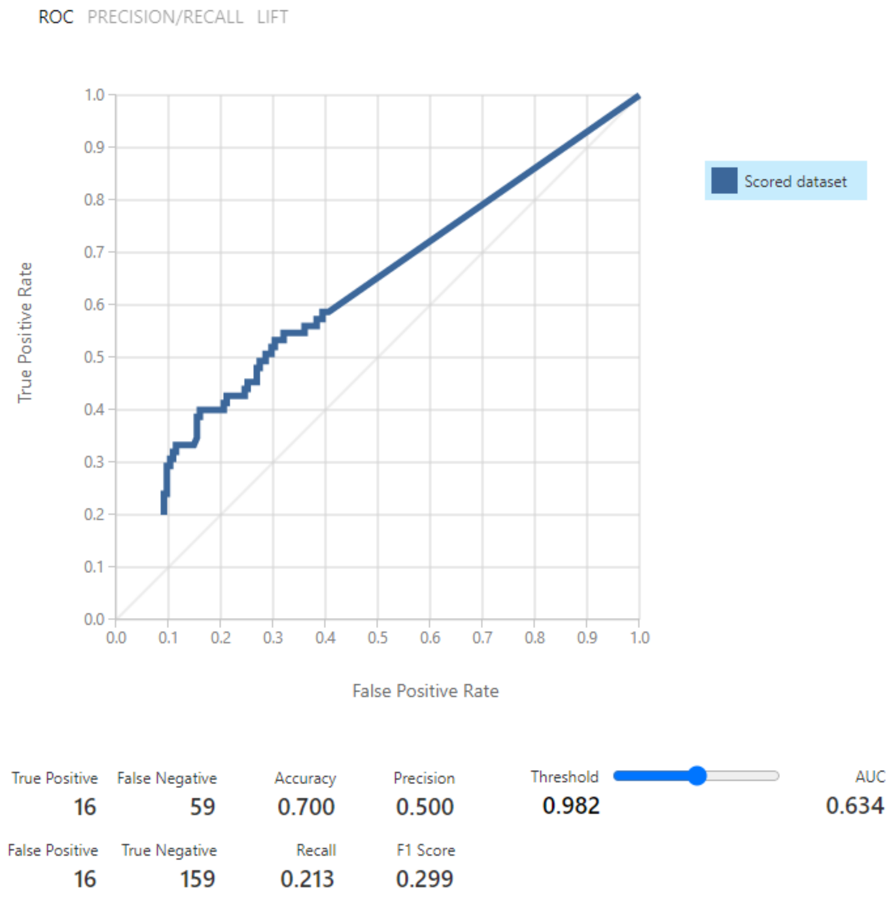
Adding Evaluate Model module

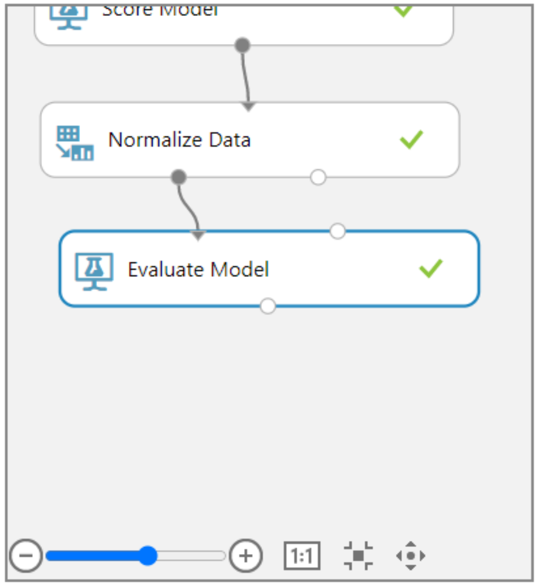


One-Class SVM in Azure ML

19. Right click on the ‘Evaluate Model’ module and click Visualize

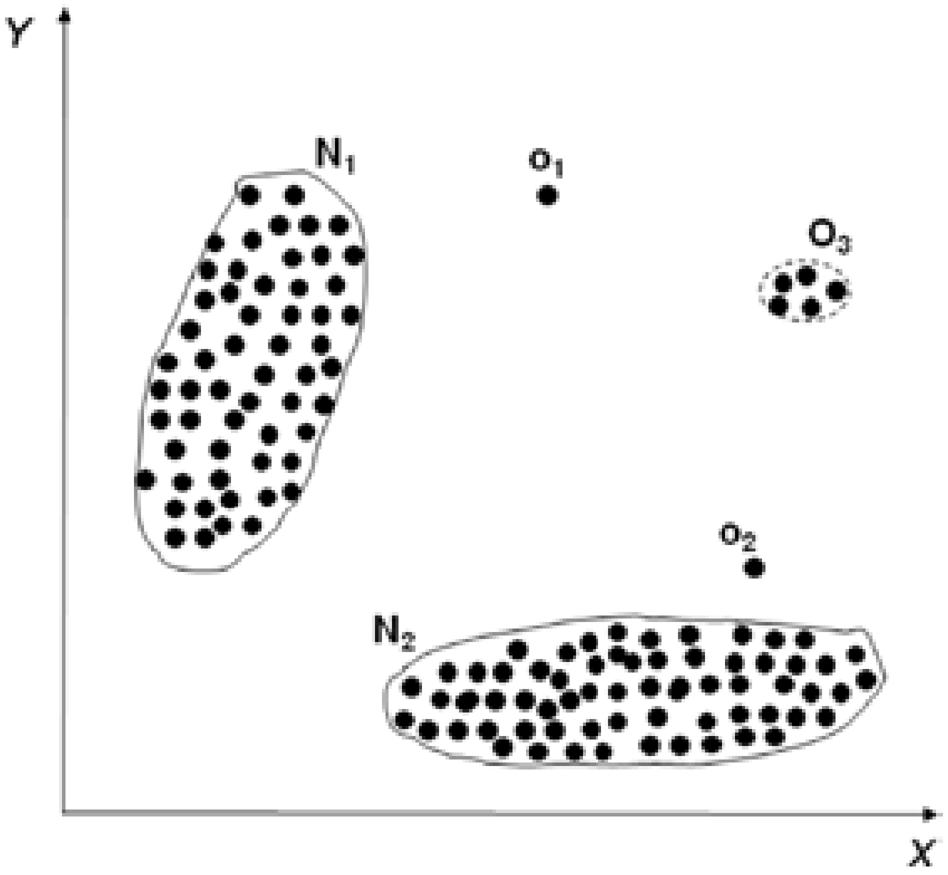
Visualize the model performance





PCA-Based Anomaly Detection

PCA-Based Anomaly Detection



- Principal Component Analysis, which is frequently abbreviated to PCA, is an established technique in machine learning. PCA is frequently used in exploratory data analysis because it reveals the inner structure of the data and explains the variance in the data

- PCA works by analyzing data that contains multiple variables. It looks for correlations among the variables and determines the combination of values that best captures differences in outcomes. These combined feature values are used to create a more compact feature space called the principal components

- For anomaly detection, each new input is analyzed, and the anomaly detection algorithm computes its projection on the eigenvectors, together with a normalized reconstruction error. The normalized error is used as the anomaly score. The higher the error, the more anomalous the instance is

PETRONAS

How to configure PCA-Based Anomaly Detection in Azure ML

1. Add the PCA Based Anomaly Detection module

You can find this module under Machine Learning, Initialize Model, in the Anomaly Detection category

2. Open the Properties Pane

Double-click the **PCA-Based Anomaly Detection** module to open the **Properties** pane

3. For **Create trainer mode**, select an option that indicates how the model should be trained

Single Parameter: Use this option if you know how you want to configure the model and provide a specific set of values as arguments.
Parameter Range: Use this option if you are not sure of the best parameters and want to perform a parameter sweep to find the optimal configuration.

4. Number of components to use in PCA, Range for number of PCA components

Specify the number of output features, or components, that you want to output. If you are unsure of what the optimum value might be, we recommend that you train the anomaly detection model using the **Parameter Range** option.

5. Specify the amount of oversampling to perform during randomized PCA training

If you specify 1, no oversampling is performed. If you specify any value higher than 1, additional samples are generated to use in training the model.

6. Enable input feature normalization

Select this option to normalize all input features to a mean of zero. Normalization or scaling to zero is generally recommended for PCA, because the goal of PCA is to maximize variance among variables

7. Connect a tagged training dataset, and one of the training module

1. If you set the Create trainer mode option to **Single Parameter**, use the Train Anomaly Detection Model module. 2. If you set the **Create trainer mode** option to Parameter Range, use the Tune Model Hyperparameters module.

8. Run the experiment

PCA-Based Anomaly Detection

- Principal Component Analysis, which is frequently abbreviated to PCA, is an established technique in machine learning. PCA is frequently used in exploratory data analysis because it reveals the inner structure of the data and explains the variance in the data

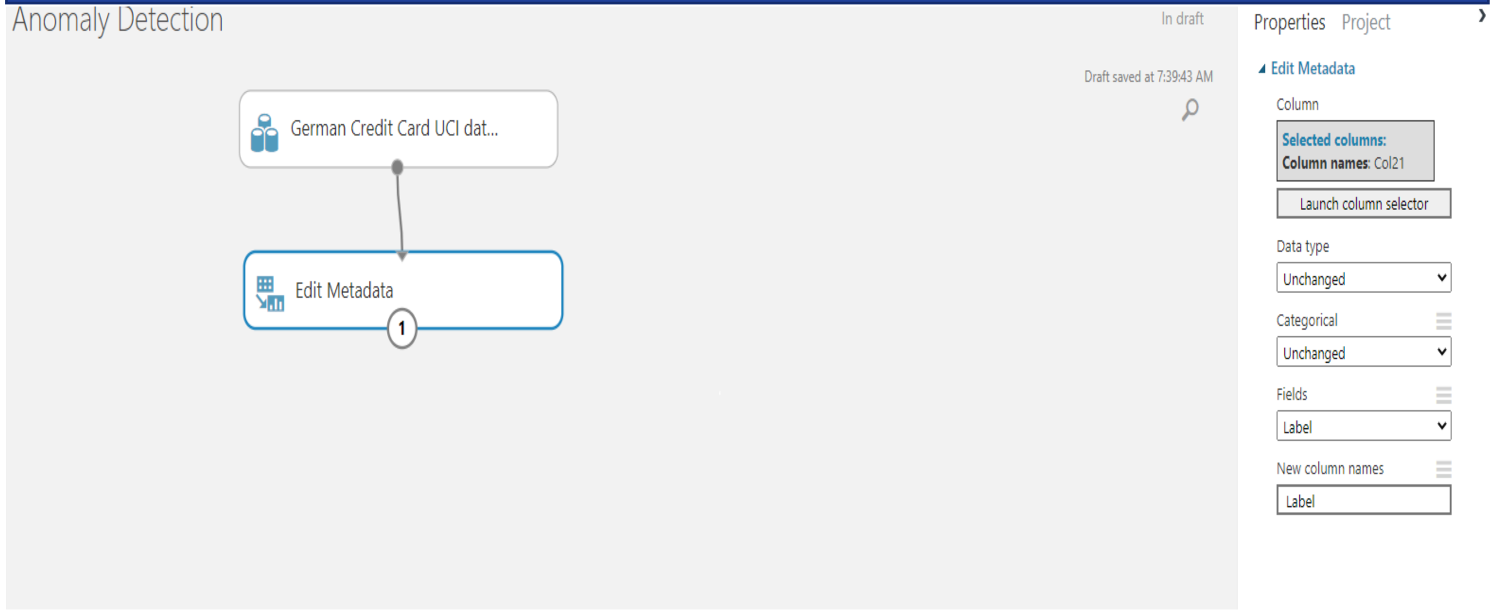
- PCA works by analyzing data that contains multiple variables. It looks for correlations among the variables and determines the combination of values that best captures differences in outcomes. These combined feature values are used to create a more compact feature space called the principal components

- For anomaly detection, each new input is analyzed, and the anomaly detection algorithm computes its projection on the eigenvectors, together with a normalized reconstruction error. The normalized error is used as the anomaly score. The higher the error, the more anomalous the instance is

PCA-Based Anomaly Detection in Azure ML

1. Load the data
2. Connect the ‘Edit Metadata’ module to the data
3. Rename the target column 'Col21' as 'Label' and also mark as type 'label' using the Metadata Editor module

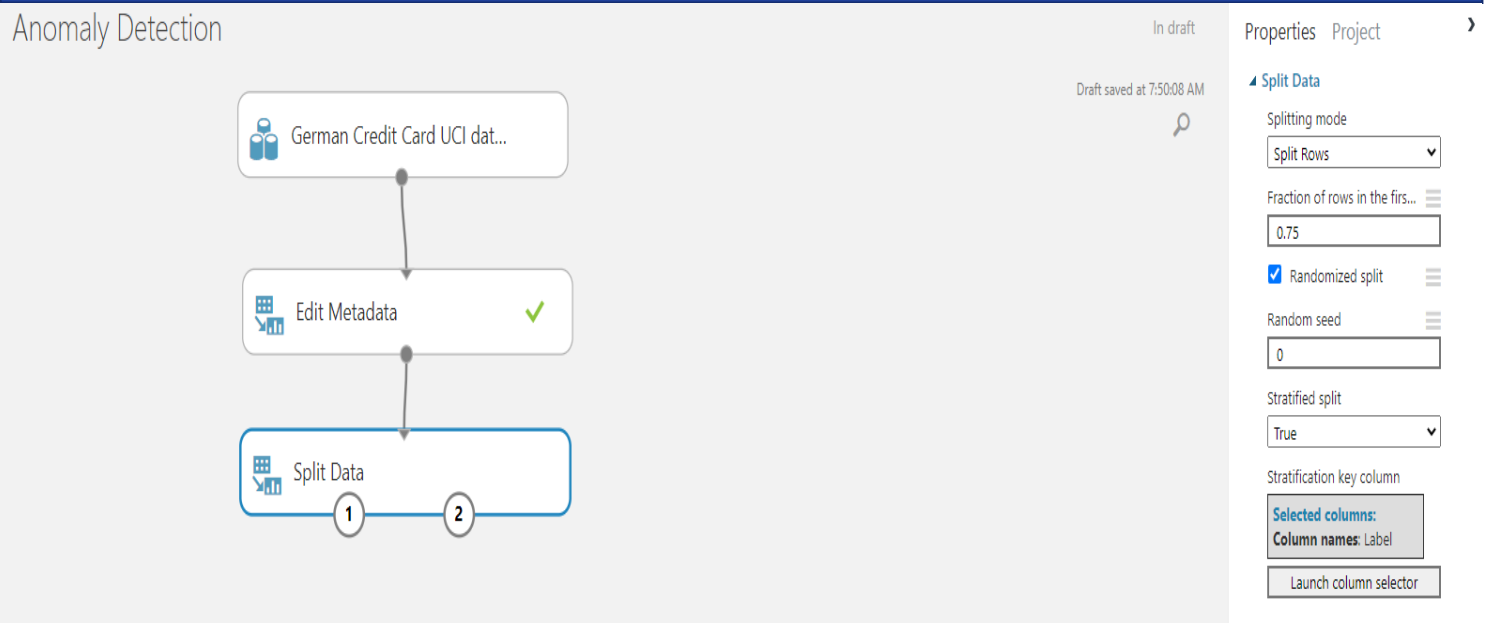
Loading the data



PCA-Based Anomaly Detection in Azure ML

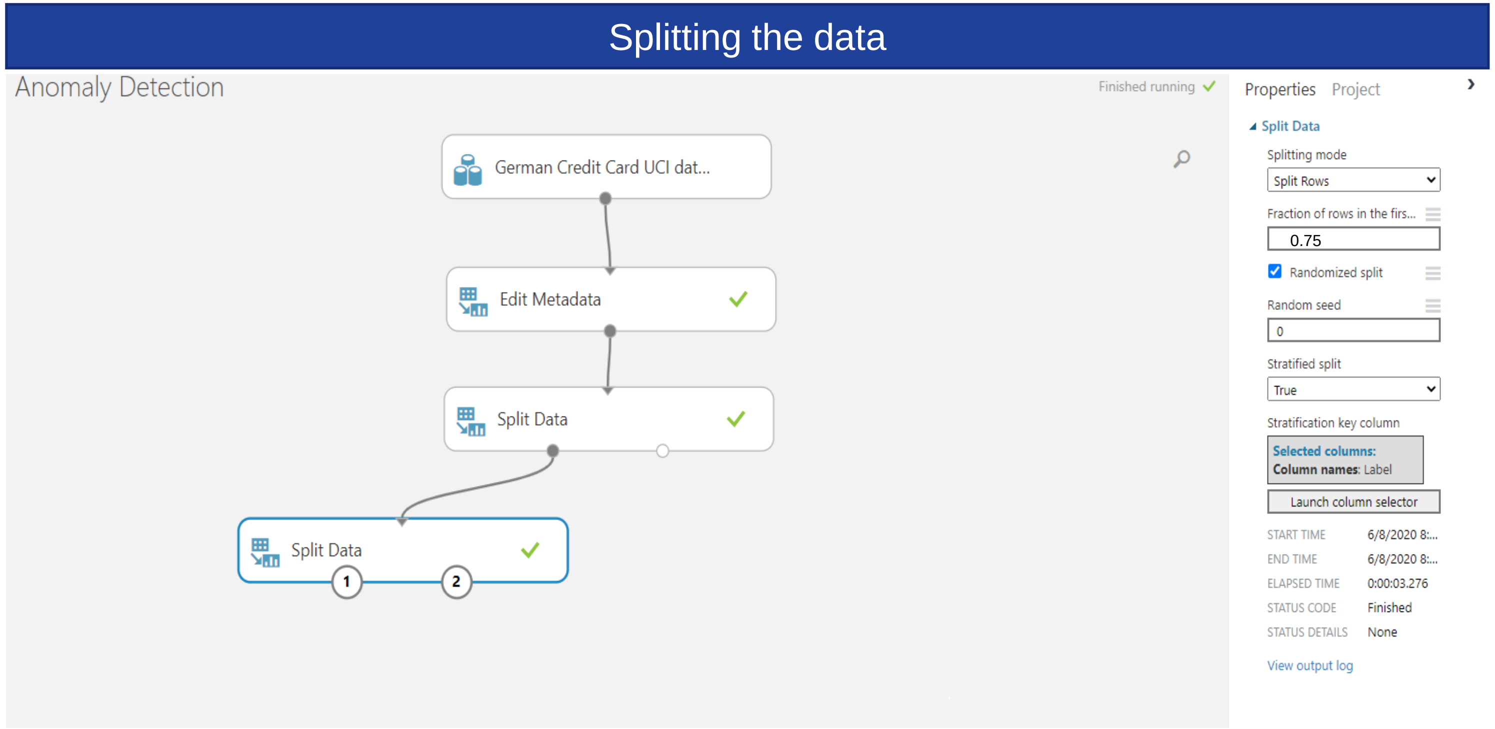
4. Split the data into 75:25 ratio by adding the ‘Split Data’ module
5. Connect the ‘Split Data’ module to ‘Edit Metadata’ module

Splitting the data



PCA-Based Anomaly Detection in Azure ML

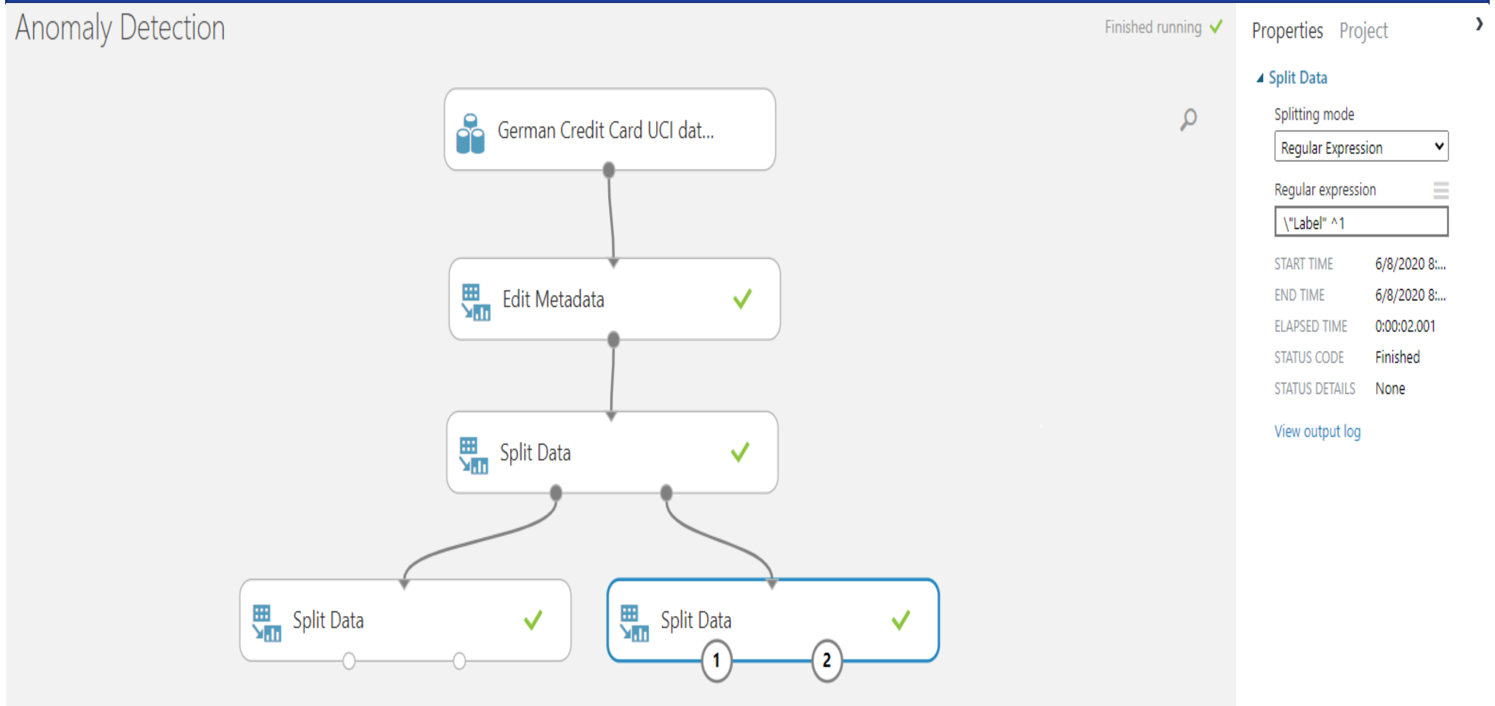
6. Add the 'Split Data' module to the existing 'Split Data' module
7. Split the data into 75:25 ratio



PCA-Based Anomaly Detection in Azure ML

8. Add the ‘Split Data’ module to the first ‘Split Data’ module
9. Split the data into normal observations where label is equal to 1

Splitting the data into normal observations

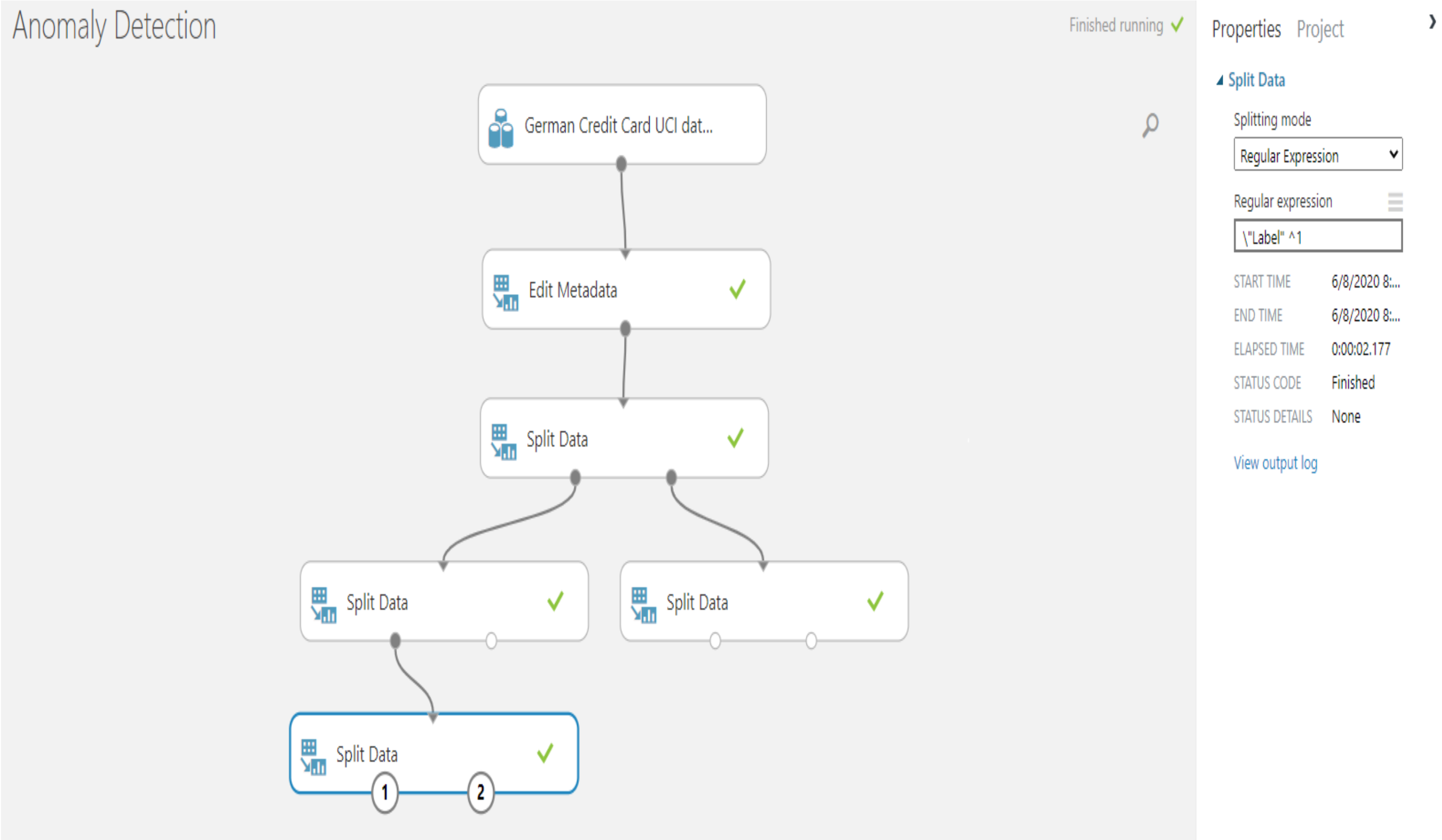


PETRONAS

PCA-Based Anomaly Detection in Azure ML

10. Add the ‘Split Data’ module to the second ‘Split Data’ module
11. Split the data into normal observations where label is equal to 1

Splitting the data into normal observations

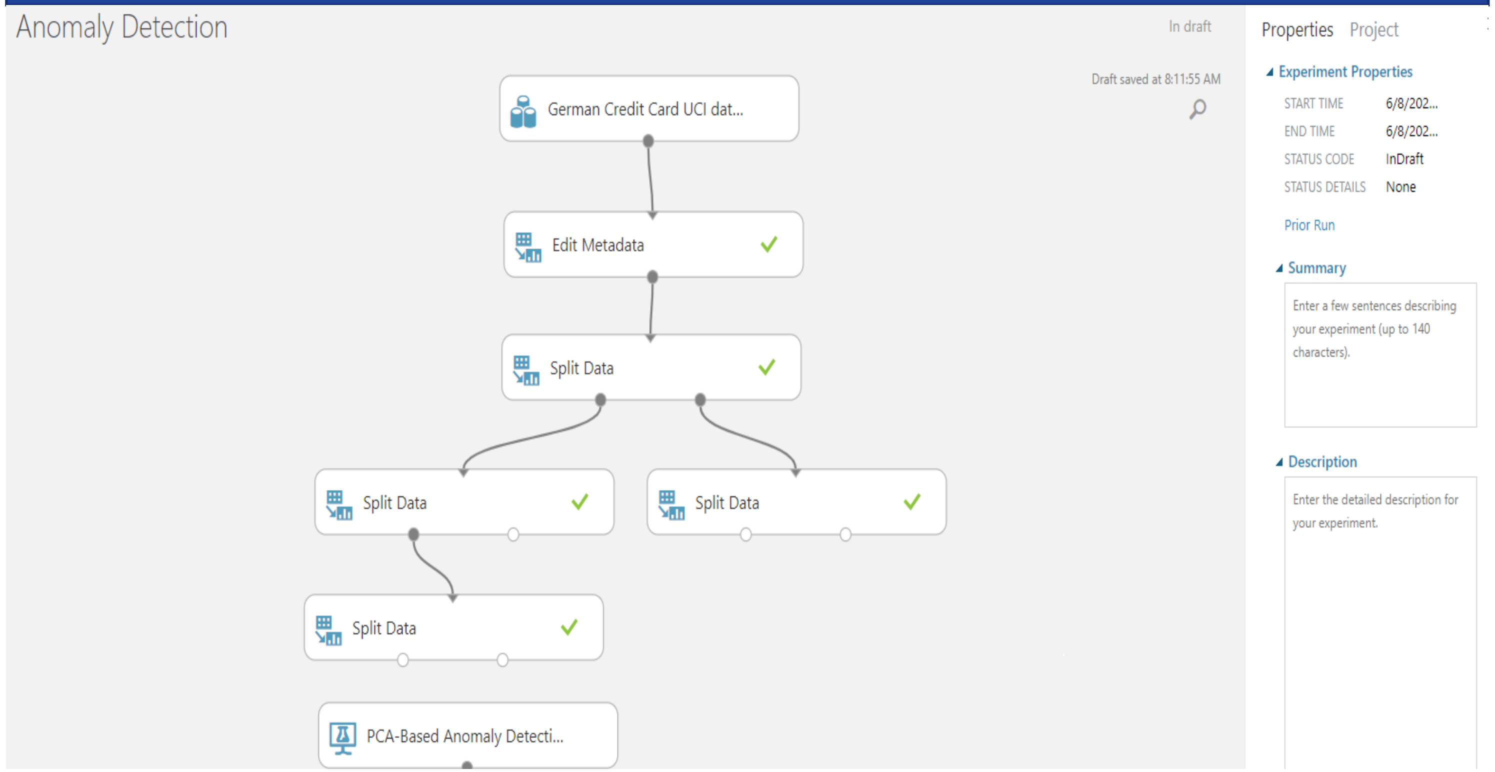


PETRONAS

PCA-Based Anomaly Detection in Azure ML

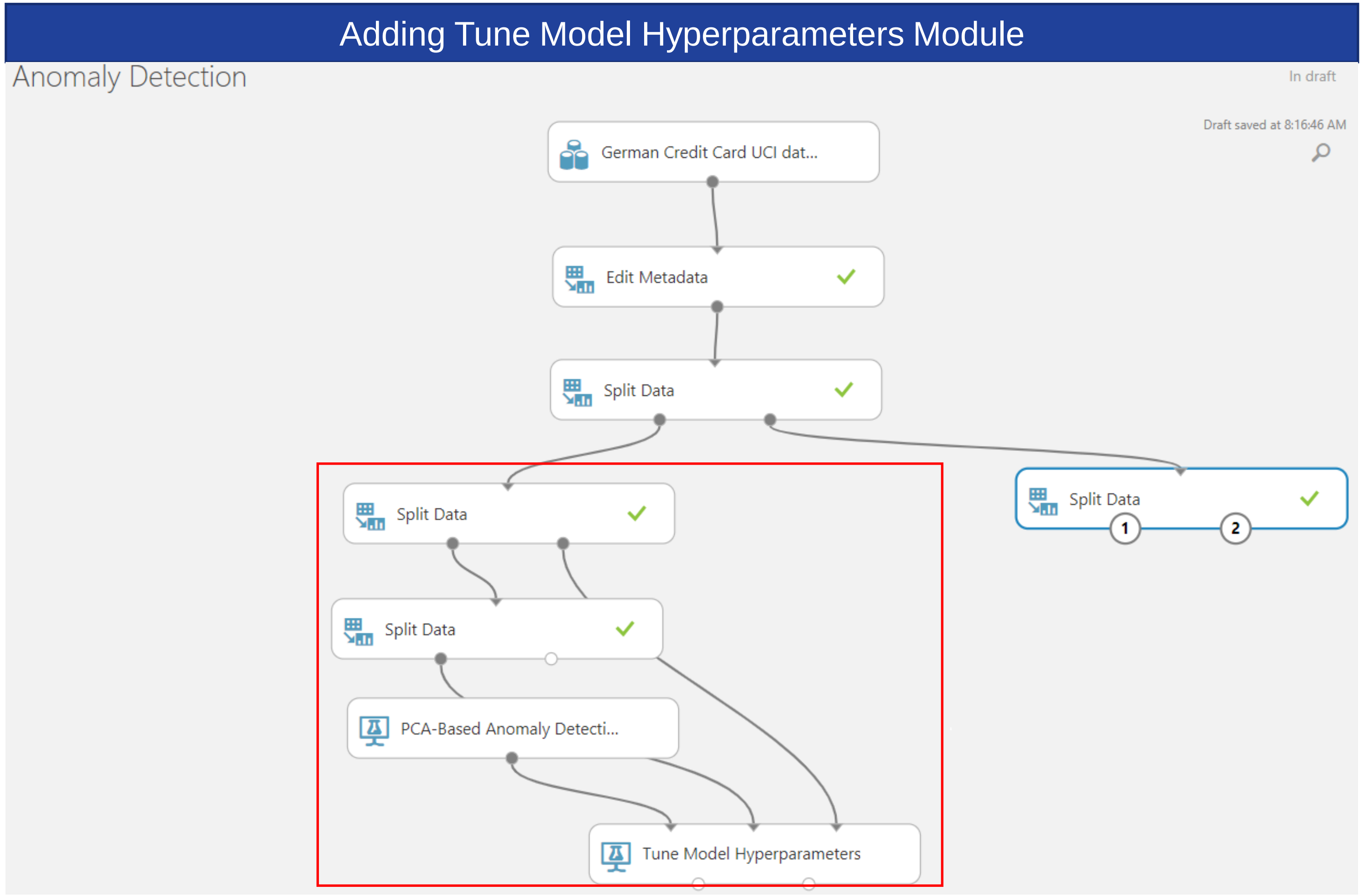
12. Add the 'PCA Based Anomaly Detection' module and connect to the last 'Split Data Module'

Adding PCA Based Anomaly Detection Module



PCA-Based Anomaly Detection in Azure ML

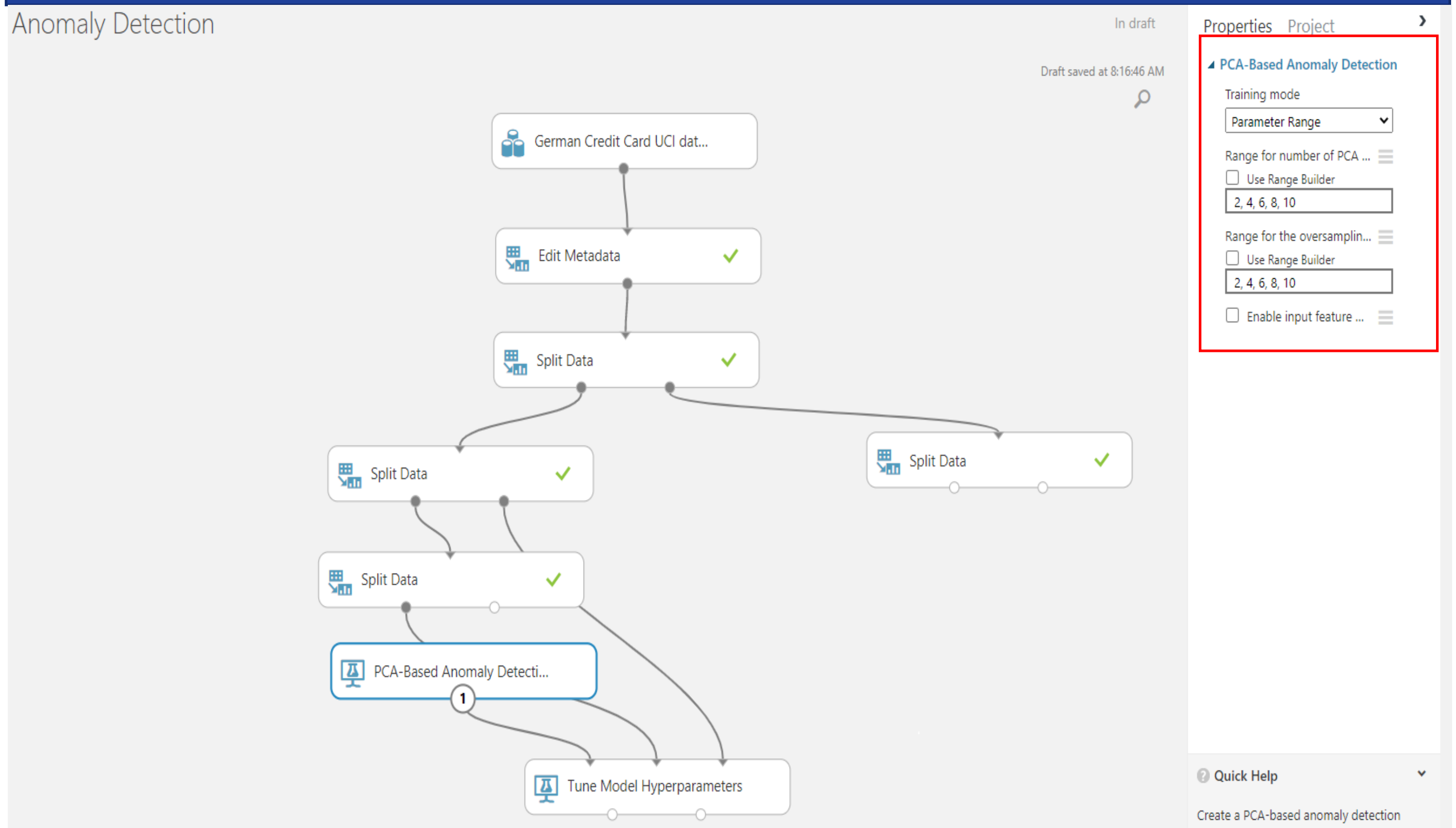
13. Add the 'Tune Model Hyperparameters' module and connect it to the 'PCA Based Anomaly Detection' and 'Split Data' modules as highlighted in the figure



PCA-Based Anomaly Detection in Azure ML

14. Select 'PCA Based Anomaly Detection' module and select the below parameters.
    - Training Mode – Parameter Range

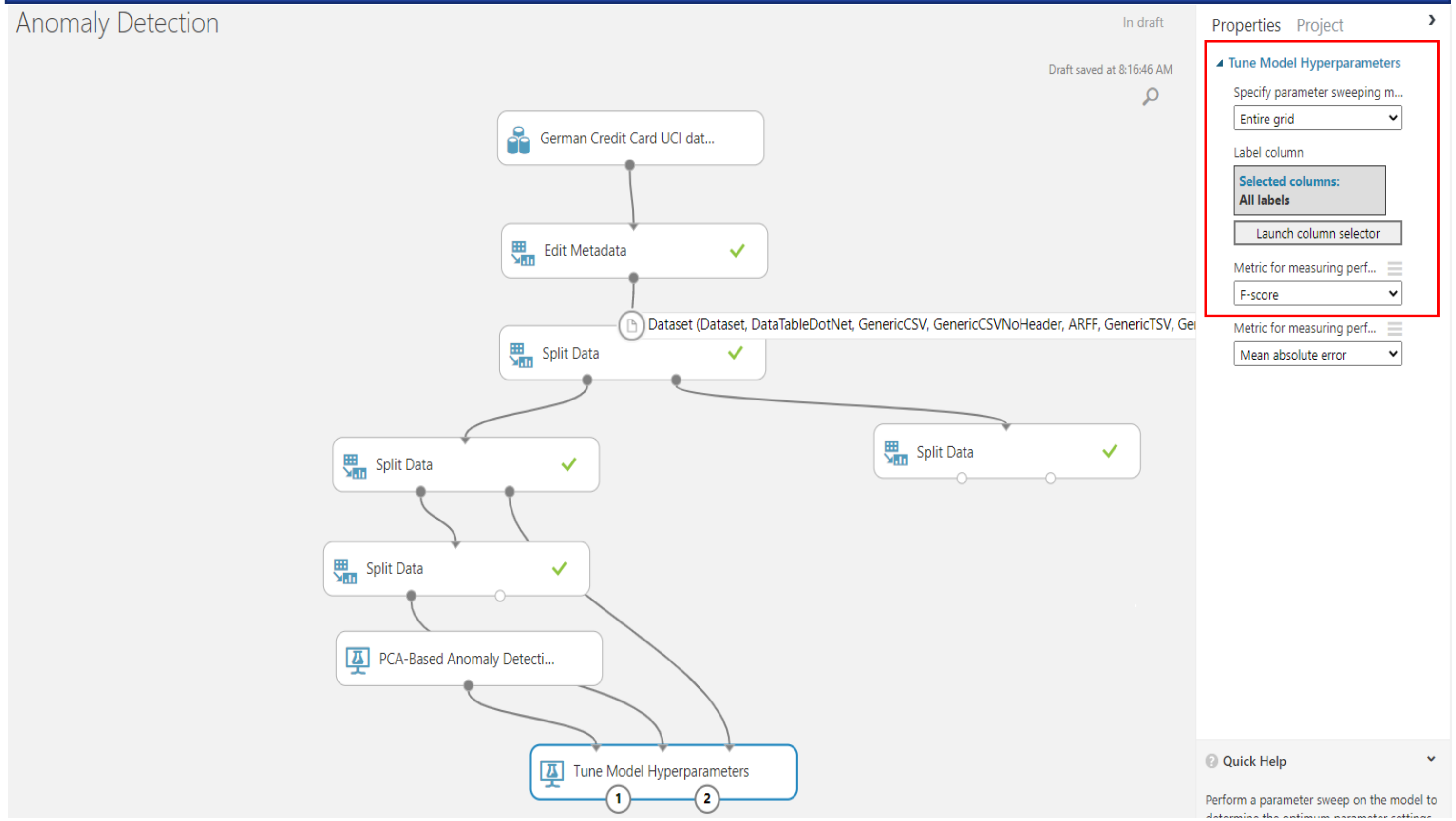
Selecting parameters for PCA Based Anomaly Detection module



PCA-Based Anomaly Detection in Azure ML

15. Select ‘Tune Model Hyperparameters’ module and select the below parameters.
    - Specify Parameter Sweeping Module – Entire Grid
    - Include all labels from ‘Launch Column Selector’
    - Metric for Measuring Performance – ‘F Score’

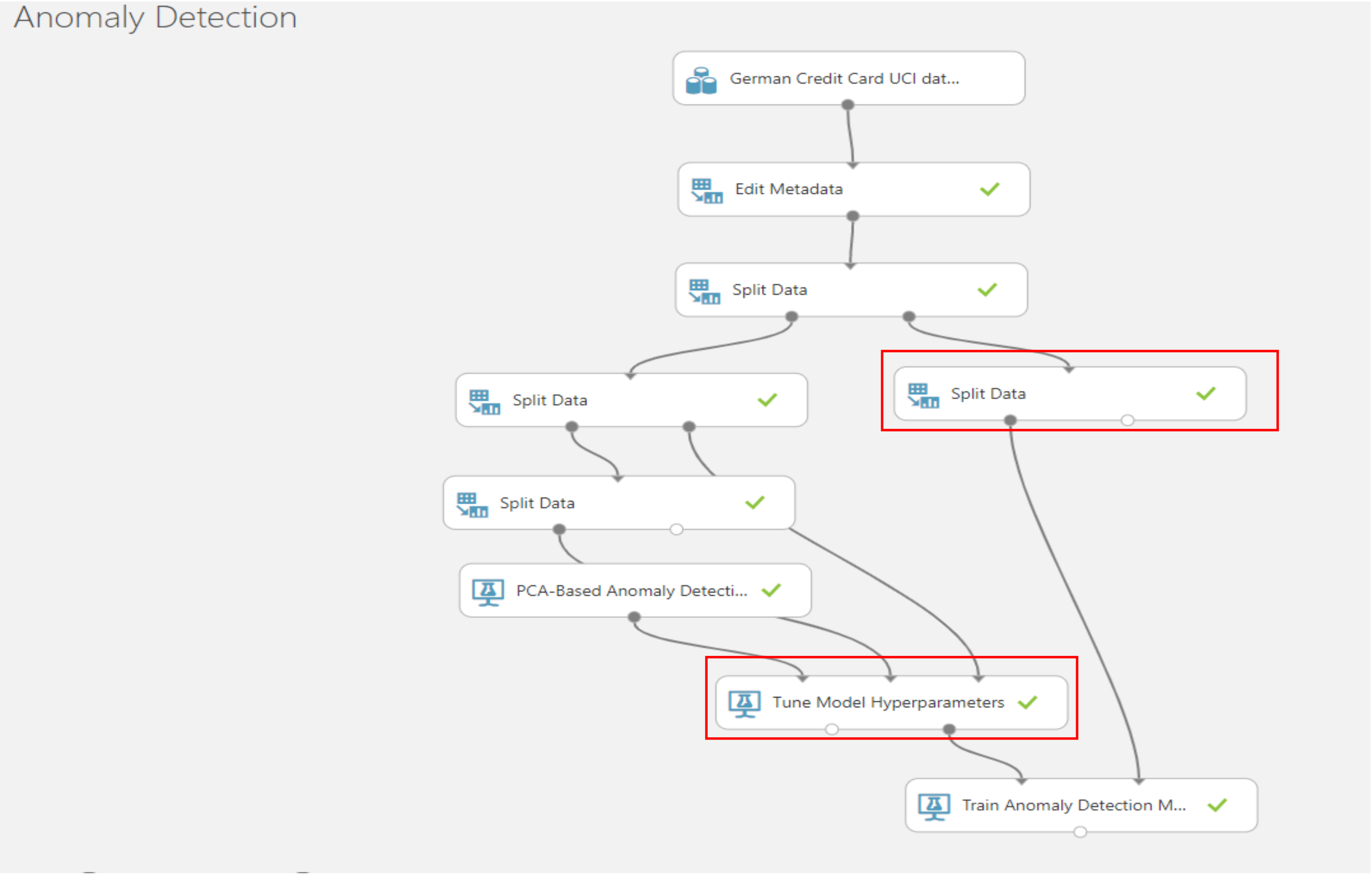
Selecting parameters for Tune Model Hyperparameters module



PCA-Based Anomaly Detection in Azure ML

16. Add the ‘Train Anomaly Detection Model’ module and connect it to the ‘Tune Model Hyperparameters’ and ‘Split Data’ modules as highlighted

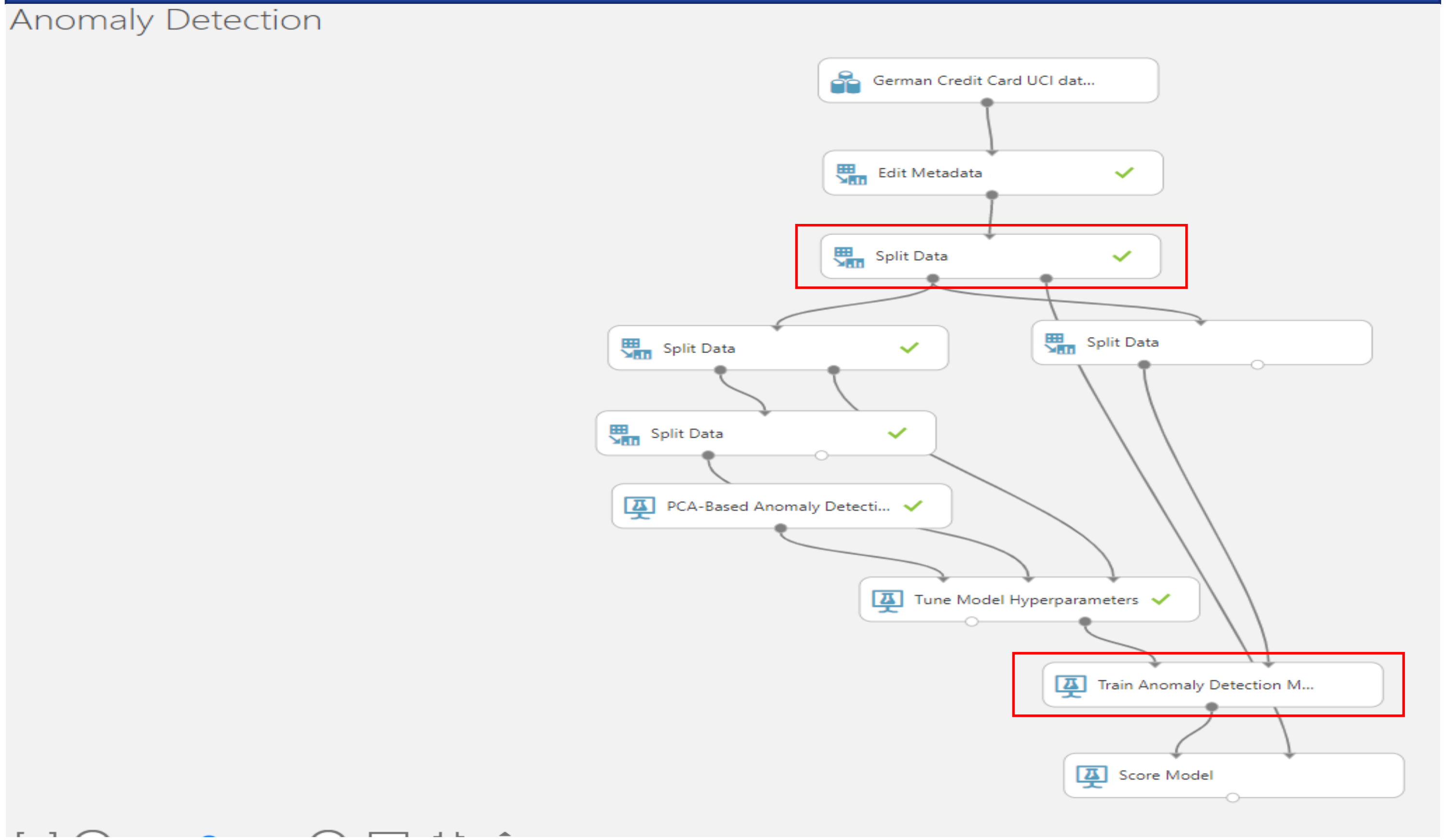
Adding Train Anomaly Detection module



PCA-Based Anomaly Detection in Azure ML

17. Add the ‘Score Model’ module and connect it to the ‘Train Anomaly Detection Model’ and ‘Split Data’ modules as highlighted

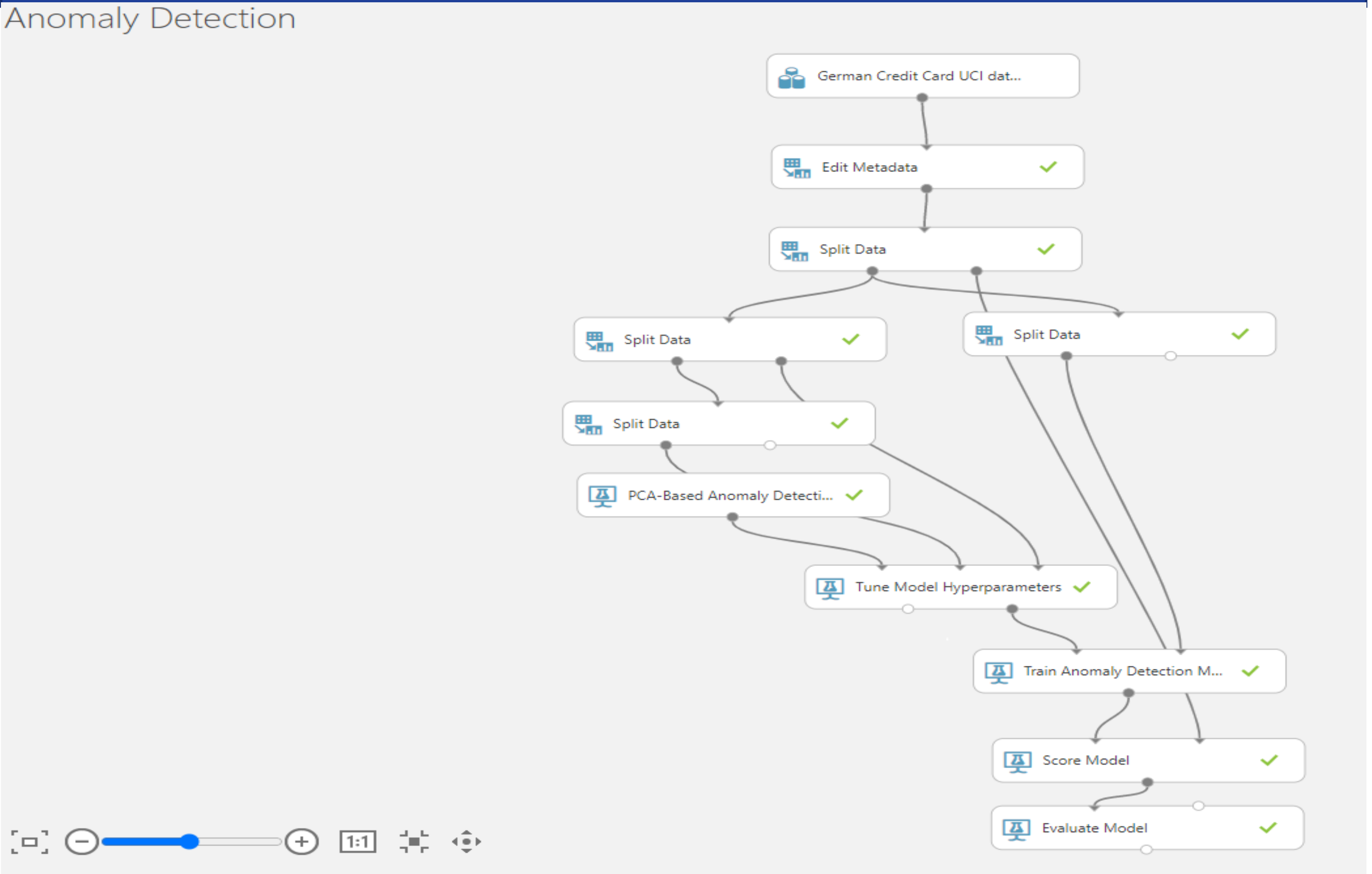
Adding Score Model module



PCA-Based Anomaly Detection in Azure ML

18. Add the ‘Evaluate Model’ module and add it to the ‘Score Model’ module
19. RUN the experiment

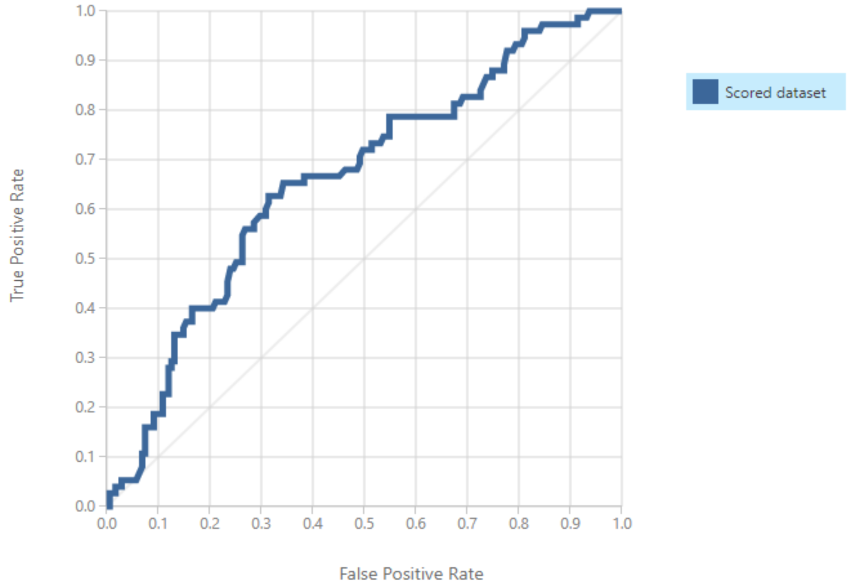
Adding Evaluate Model module

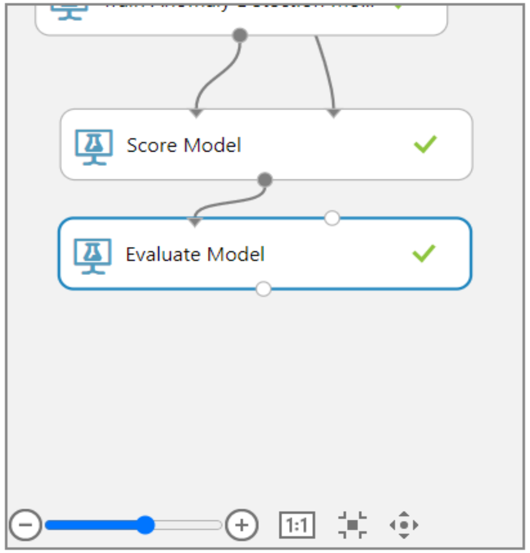


PCA-Based Anomaly Detection in Azure ML

20. Right click on the ‘Evaluate Model’ module and select ‘Visualize’

Visualizing the model performance





True Positive	False Negative	Accuracy	Precision	Threshold	AUC
72	3	0.400	0.329	0.5	0.661
False Positive	**True Negative**	**Recall**	**F1 Score**		
147	28	0.960	0.490		
Positive Label	**Negative Label**				
2	1				

PETRONAS

Internal

Summary

Summary

1

- An observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism.

2

- Anomaly detection techniques supported in Azure ML are One-Class Support Vector Machine and PCA-Based Anomaly Detection.

PETRONAS

Thank you for your passion!

PETRONAS