Data Analytics Software Project 3

Microsoft Cortana Intelligence Suite

Wakjira Ashenafi

Date 23.02.2018

Introduction

In this project we will cover some subtasks on Microsoft Cortana Intelligence Suite. On the first task we use sample data named Energy Efficiency Regression Data set from the Microsoft Azure Machine Learning studio and use regression analysis and the second task we import the iris data set and use two different classification methods and compare the results.

TASK 1. Regression Analysis

The Microsoft Azure Learning studio interface looks like the figure bellow. We can simply drag the entities that we want from the tabs to the working are. After we are done with the selection we save and run the analysis that we want to analyse.

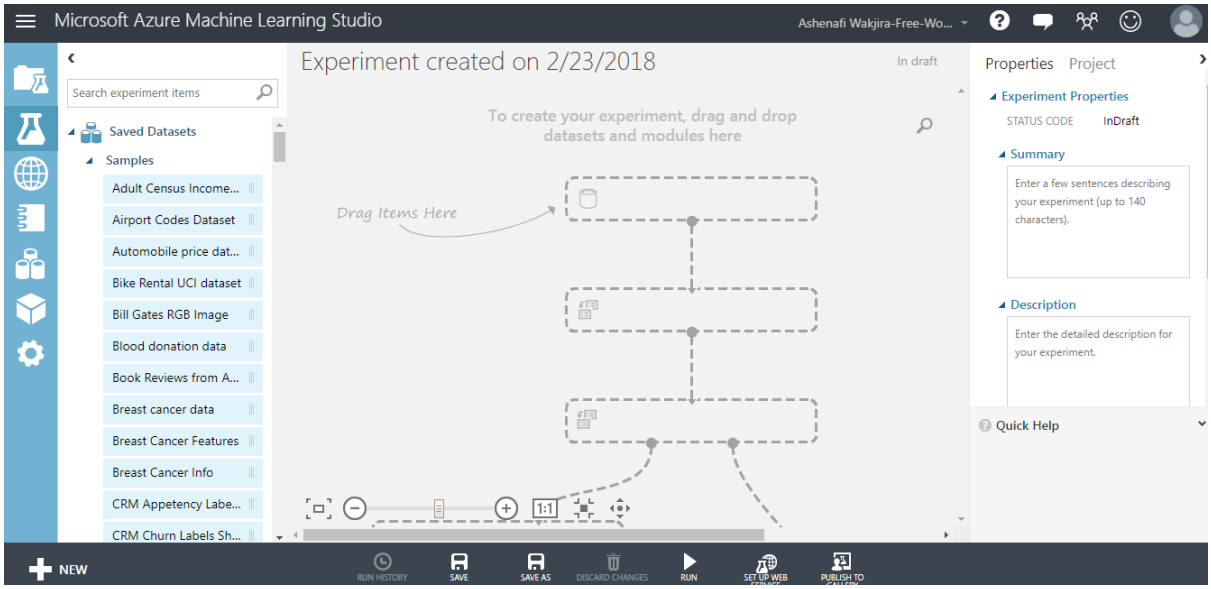


Figure 1. Microsoft Azure Machine Learning interface

First let us check out the descriptive summary of the data to see what the data looks like, the number of features and class, the outliers and so on. We can simply do that by right click on the data and see the visualization, it gives us the frequency distribution of each data with an option of bar chart as well as boxplot. It also states the number of column and rows, missing values, and all other statistical values with an option of comparison of two columns.

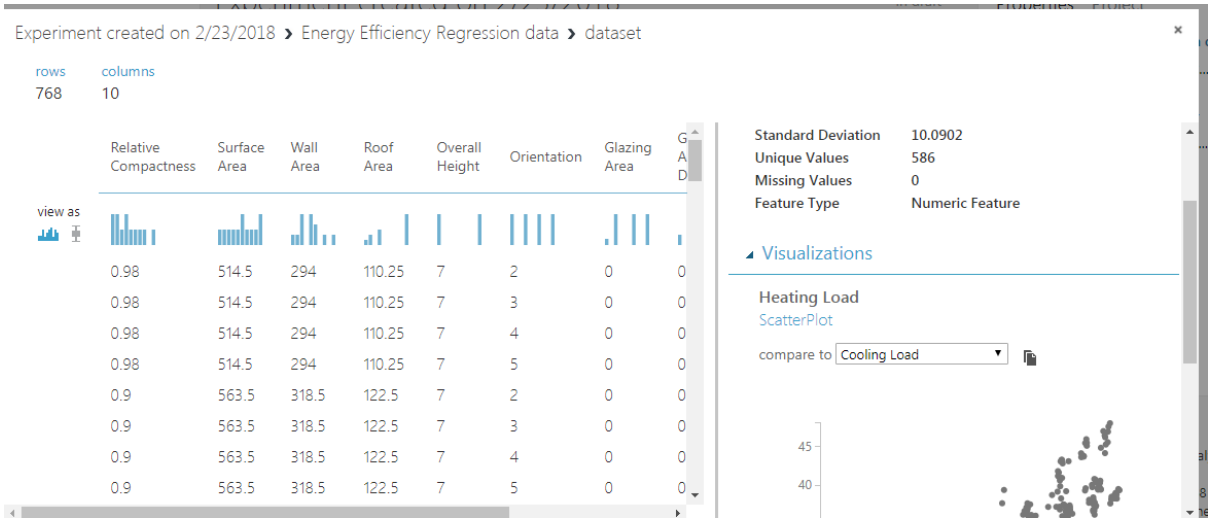


Figure2.

But if we want to explore more on the data, there is some other options like the Summarize data tab. It gives us around 22 different statistical description like skewness, p values and so on.

Figure 3. Summarize data out puts with more statistical options

While analysing this data it logical to use CRISP-DM steps. Let us see what the data is about. The dataset contains the energy analysis result from stimulating 12 different building shapes on Ecotect. As we observe the dataset has 768 samples, 8 features characterizing the buildings (Relative Compactness, Surface Area, Wall Area, Roof Area, Overall Height, Orientation, Glazing Area, Glazing Area Distribution) and two responses (Heating Load, Cooling Load).

As we can observe the dataset has tow response variables and we are explicitly required to predict the Heating Load. As a result, we need to have a data preparation phase. In this phase we exclude the second response variable cooling Load from our model.

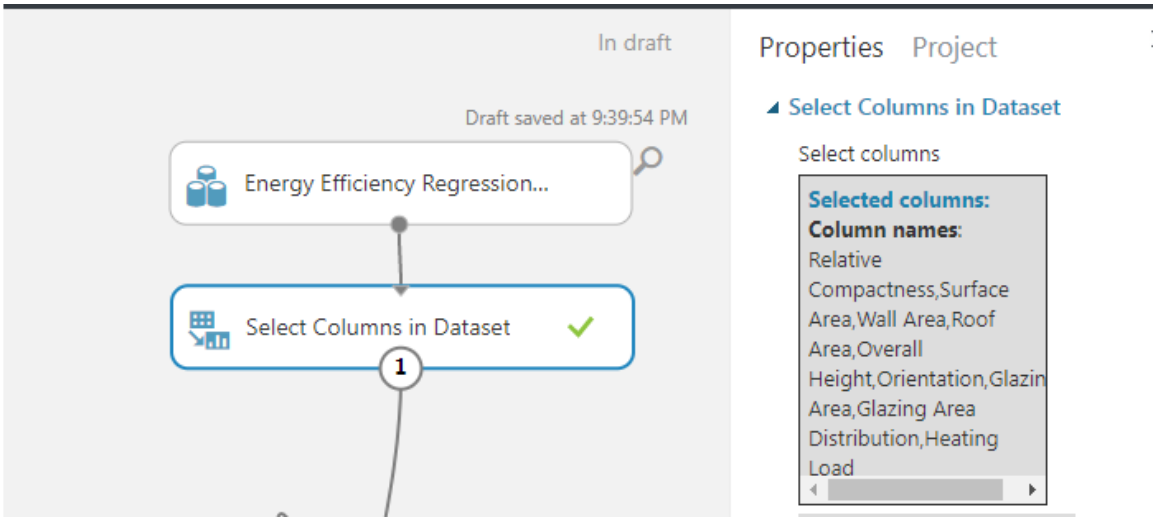


Figure 4. Data preparation, deselect the cooling Load from the model.

Next step will be to split the data to training and testing set. We can select the ratio to any desired standard ratio. I decided to randomly split it to 70% training and 30 for testing with random seed set to an arbitrary number.

In this task we explicitly told to apply linear regression model. After selecting Train model block, Linear regression block, we connect the linear regression model block Train model block, the split data to Train model. And this point we need to choose the response variable too.

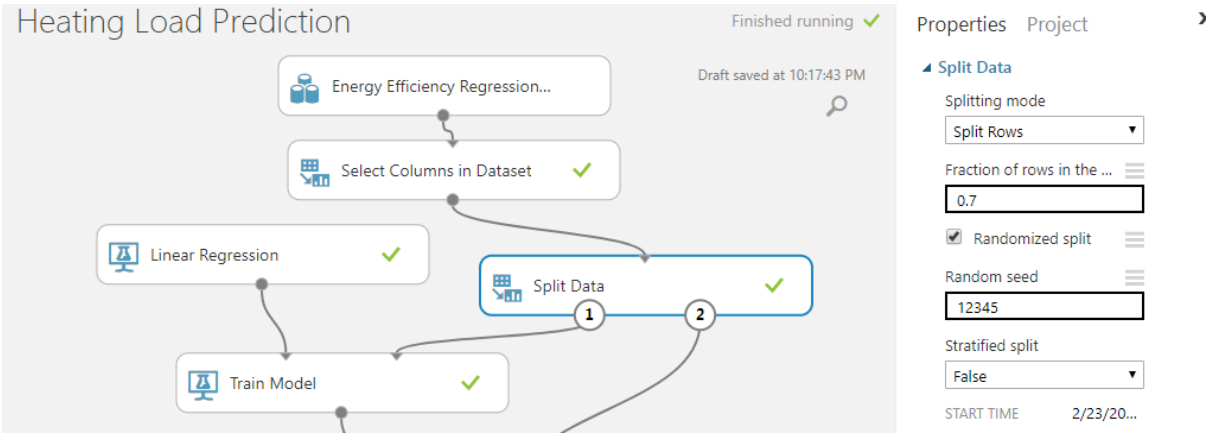


Figure 5. Split the data and connecting to the Linear regression model

Next step is to add the score model block, which takes the trained data from the regression model and the testing data. And finally, we use the Evaluate Model block to evaluate the regression model prediction with the new testing set data. Figure 5 shows the final model after we run the model.

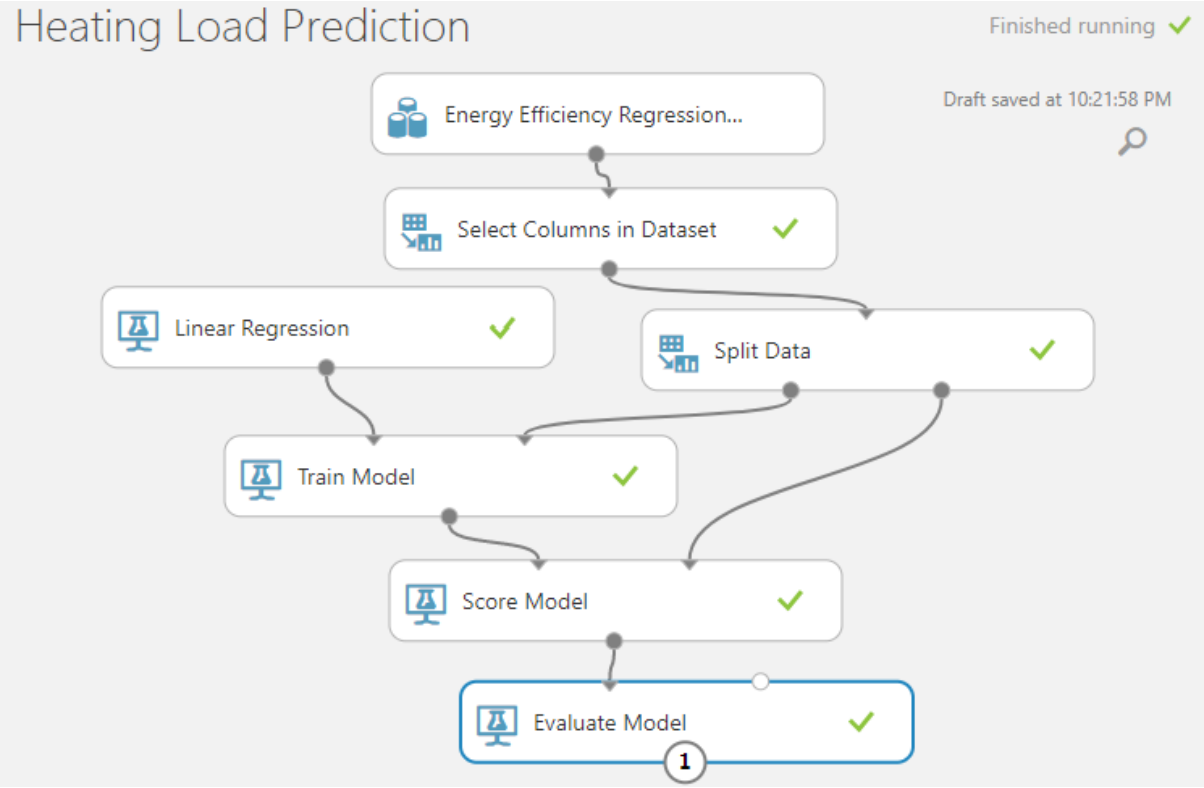


Figure 6. Final model for Linear regression.

Output of each steps

The output of the row data is shown on figure two while showing the descriptive analysis of the data, the select column block is the same data without the Cooling Load, the split data has two data sets that we can visualize.

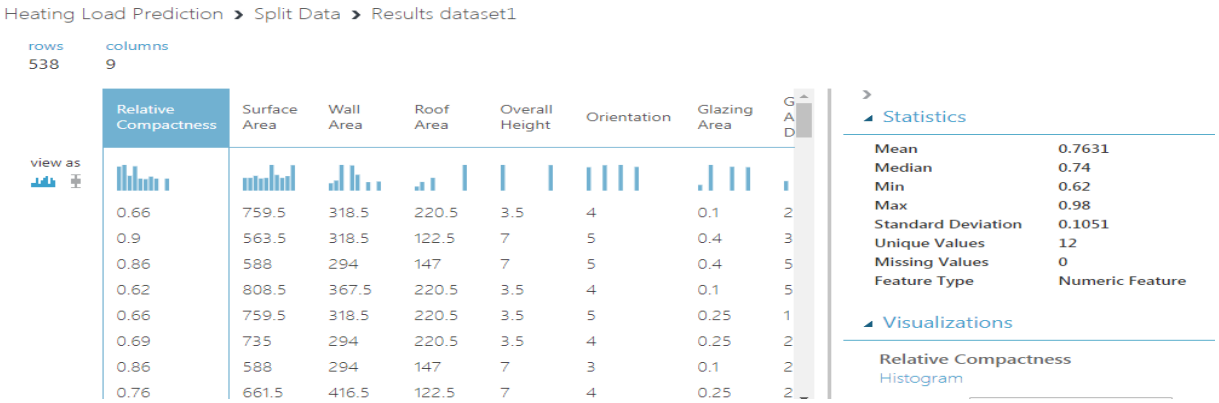


Figure 7. Training data set

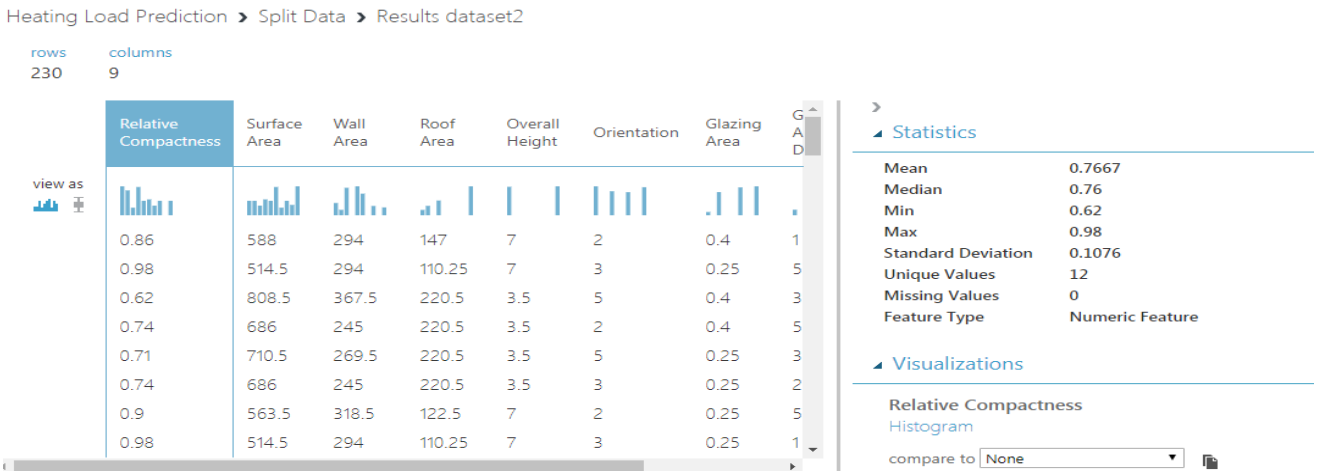


Figure 8. Testing dataset

The training data block output shows the linear coefficients of each features and the bias(the y intercept) of the linear regression model.

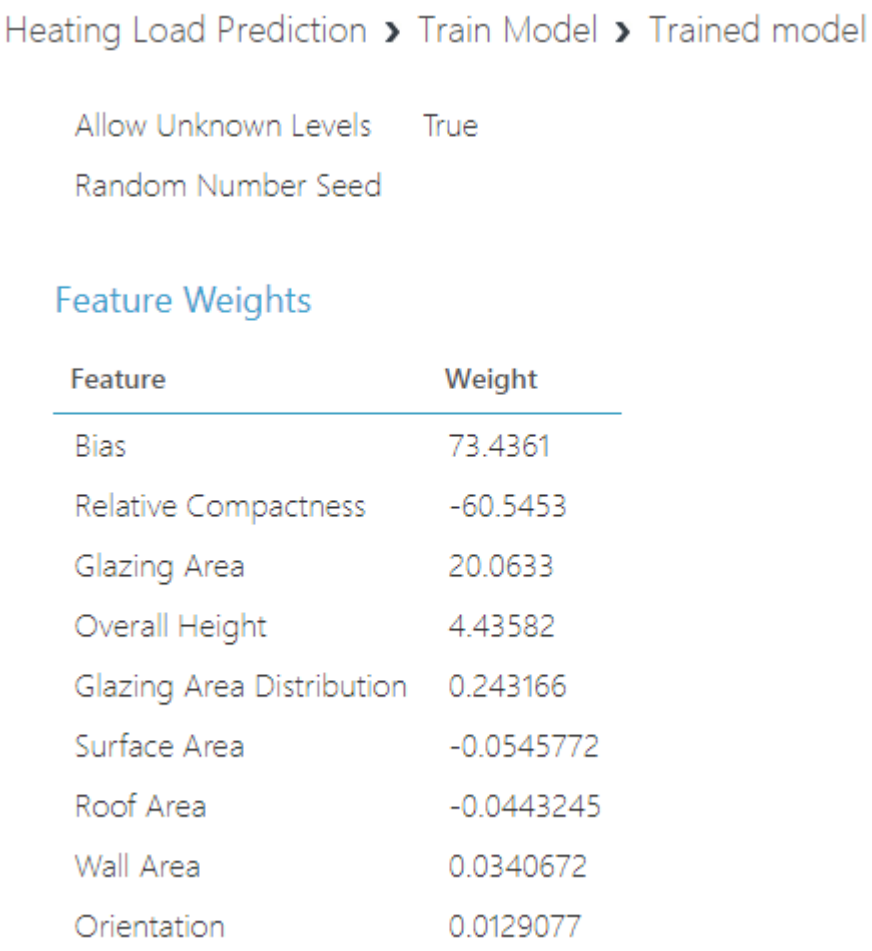


Figure 9. The weights of each features for the linear regression model

The Score Model shows the predicted values for the testing data set with. The figure bellow shows the scored values and the actual Heating Load side by side for each observation.

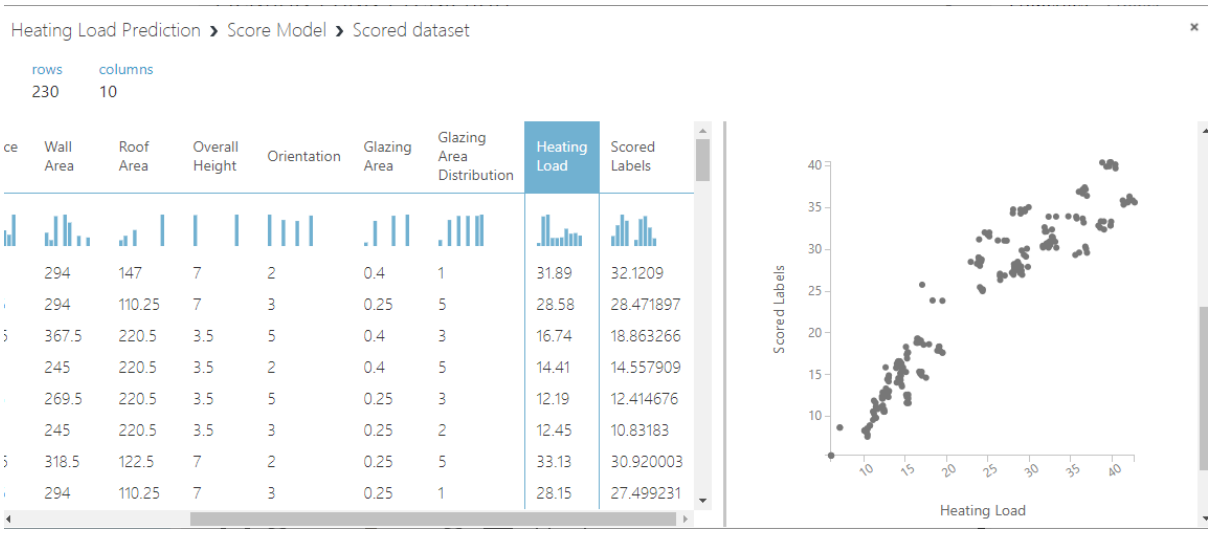


Figure 10. The Score Model comparing the predicted values and the actual value.

Evaluation model output Metrix differs from model to model. In case of linear regression our evaluation metrices are shown on the figure below. As we can see the errors are quite small which means that predicted values actual values are close to each other (the deviation is very small). And coefficient of deviation close to one is also a good prediction.

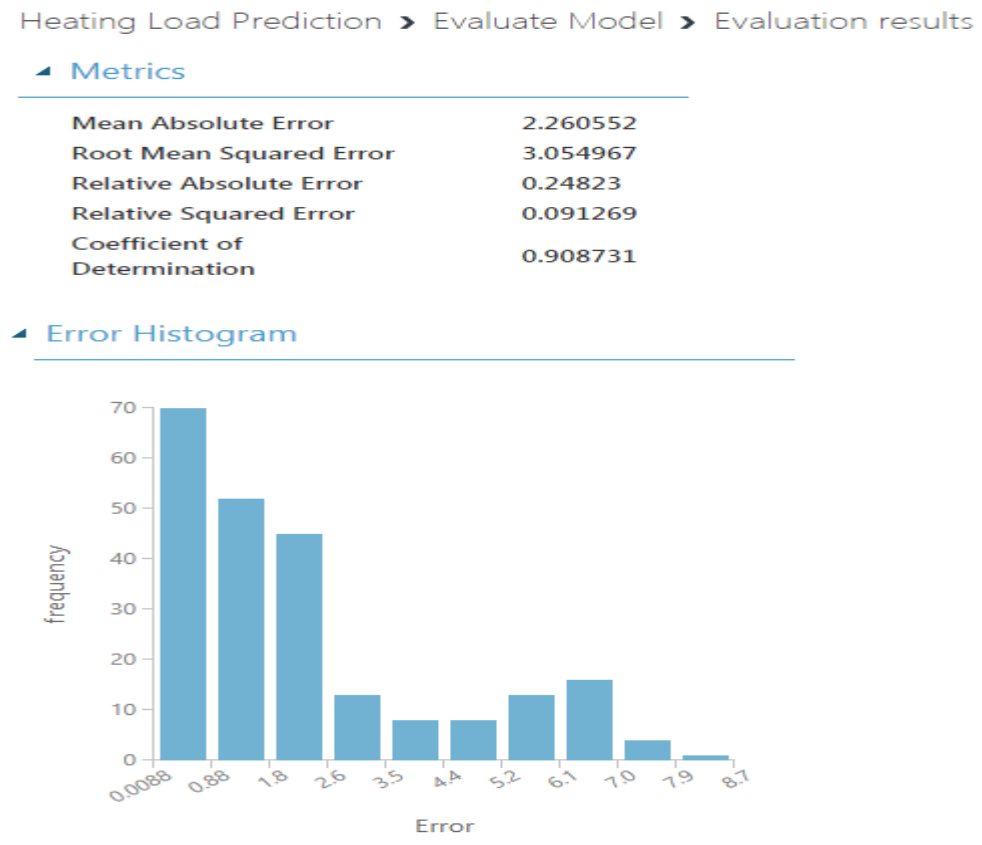
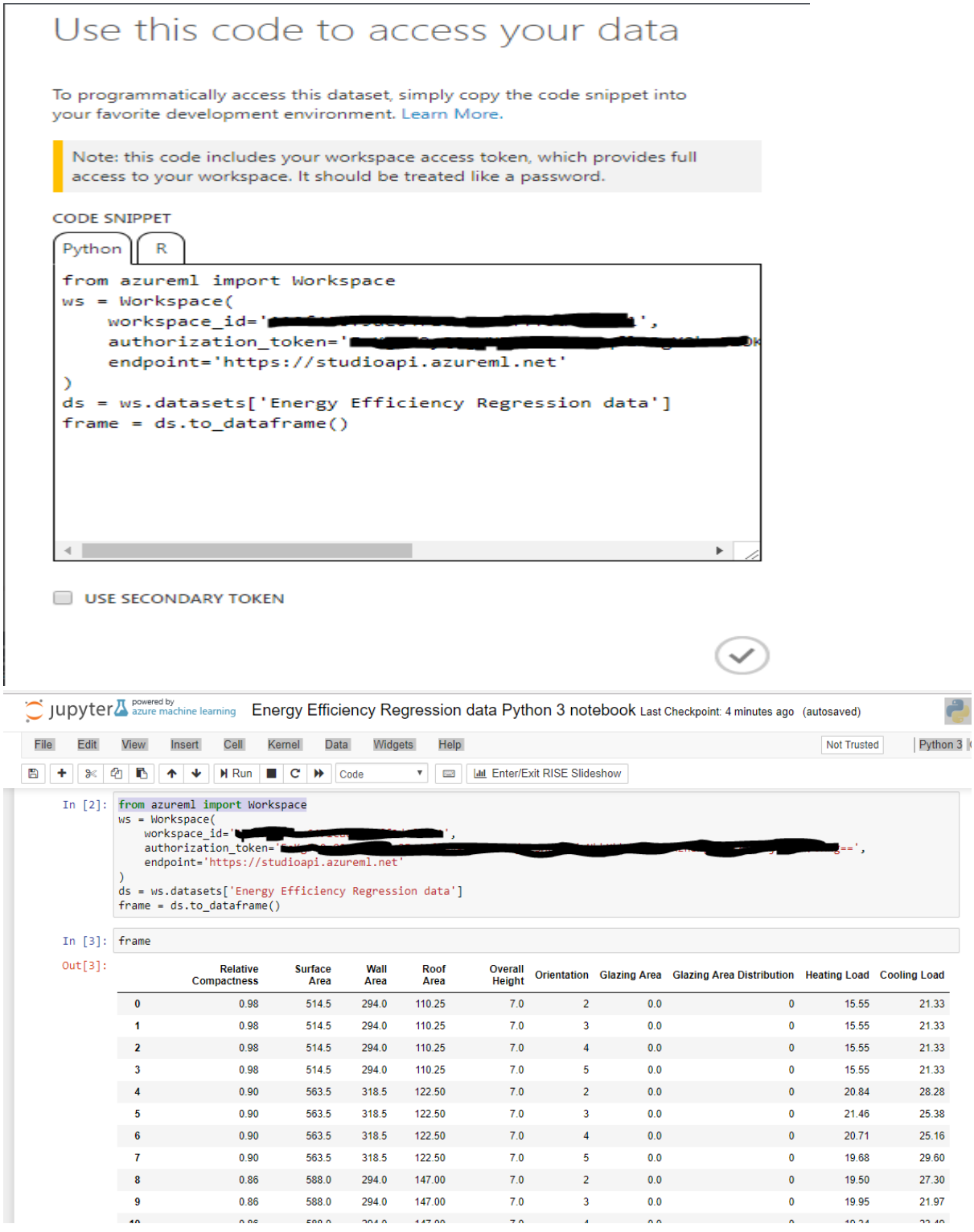


Figure 11. Evaluation of the regression model

Produce an API code for your project

To produce an API code for the project right click on the data generate data access code. With option of python and R.



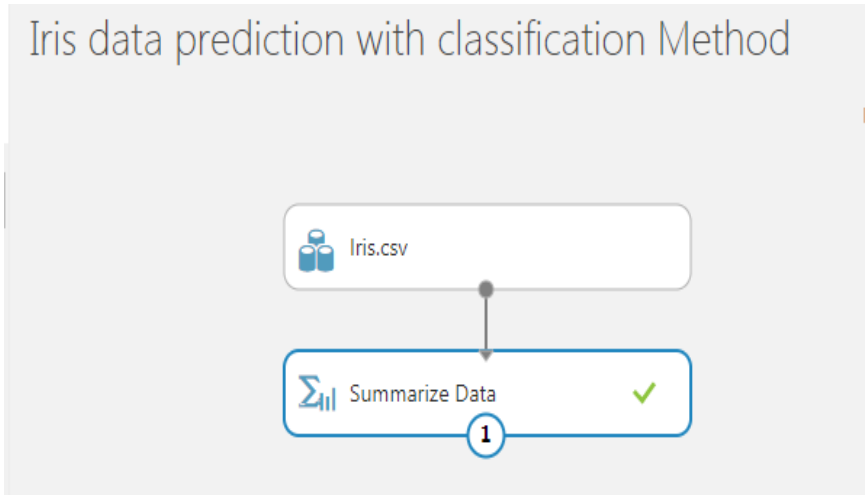
	Relative Compactness	Surface Area	Wall Area	Roof Area	Overall Height	Orientation	Glazing Area	Glazing Area Distribution	Heating Load	Cooling Load
0	0.98	514.5	294.0	110.25	7.0	2	0.0	0	15.55	21.33
1	0.98	514.5	294.0	110.25	7.0	3	0.0	0	15.55	21.33
2	0.98	514.5	294.0	110.25	7.0	4	0.0	0	15.55	21.33
3	0.98	514.5	294.0	110.25	7.0	5	0.0	0	15.55	21.33
4	0.90	563.5	318.5	122.50	7.0	2	0.0	0	20.84	28.28
5	0.90	563.5	318.5	122.50	7.0	3	0.0	0	21.46	25.38
6	0.90	563.5	318.5	122.50	7.0	4	0.0	0	20.71	25.16
7	0.90	563.5	318.5	122.50	7.0	5	0.0	0	19.68	29.60
8	0.86	588.0	294.0	147.00	7.0	2	0.0	0	19.50	27.30
9	0.86	588.0	294.0	147.00	7.0	3	0.0	0	19.95	21.97

Figure 12. How to generate an API code for python demo

Task 2. Classification

2.1 Multiclass Decision forest

In this task we import the iris data set and chose two different classification methods and make predictions. Then we compare the predictive strength of the two models. First let us import the data to Microsoft Azure ML studio and visualize the data. To investigate the data, let us use the Summarize Data block. This block gives us the descriptive statistics of the data.



Iris data prediction with classification Method › Summarize Data › Results dataset

rows 5 columns 23

Feature	Count	Unique Value Count	Missing Value Count	Min	Max	Mean	Mean Deviation
sepal length	150	35	0	4.3	7.9	5.843333	0.687556
sepal width	150	23	0	2	4.4	3.054	0.333093
petal length	150	43	0	1	6.9	3.758667	1.56192
petal width	150	22	0	0.1	2.5	1.198667	0.658933
Class	150	3	0				

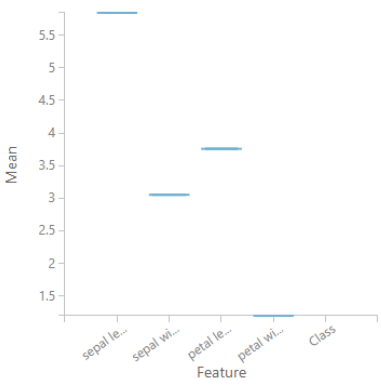


Figure 13. Descriptive Summary of iris dataset

As we can see from figure 13, the data contains 4 features, 1 class and 150 observations. No missing value on the dataset and we have 22 different statistical information with this option. For quick observation of the data we can simply right click the data and visualize.

Unlike the above data we don't need to select column out. In this data set we only have one class and 4 features. Next step would be to split the data. Let us divide it with the same proportion as before. 70% for training and the rest for testing. As we can see from the figure bellow, the data is split in two. When we try to visualize the Split Data block, we have two data sets.

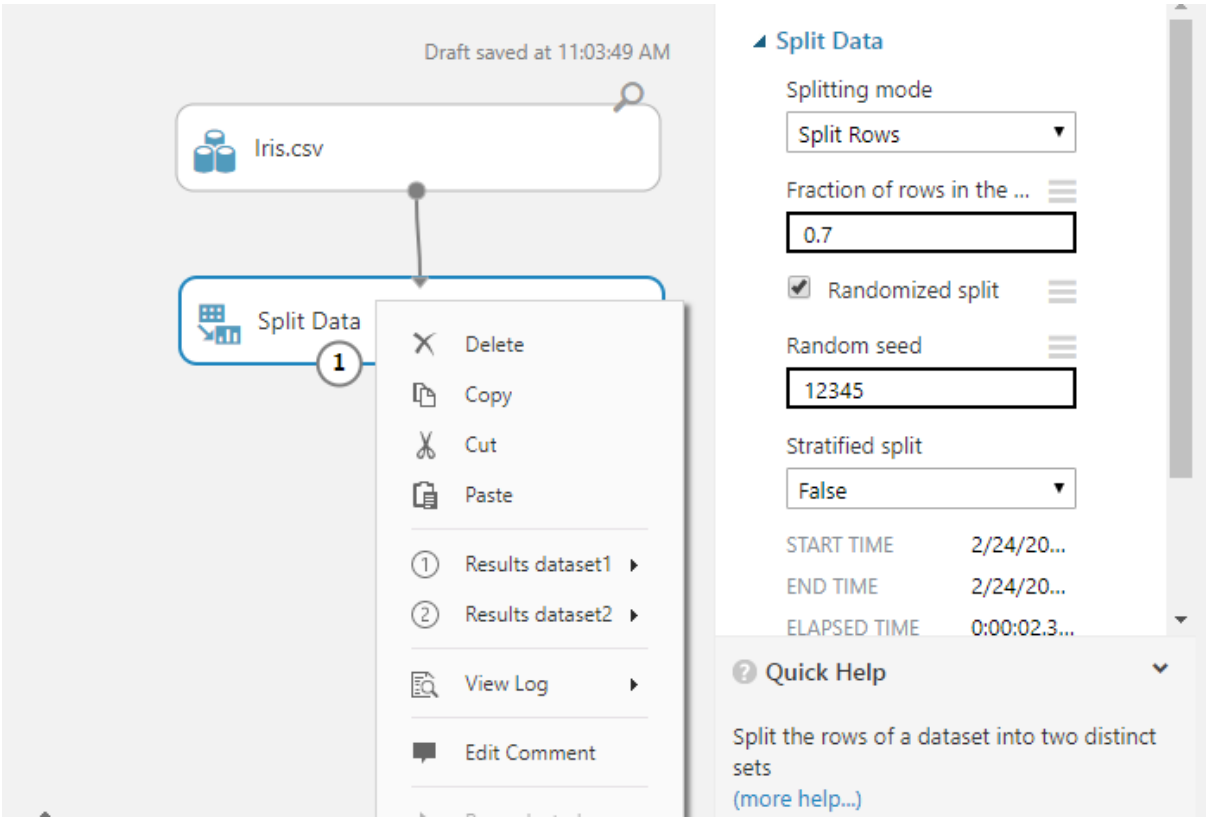


Figure 14. Split Data dividing the data in to two sets with the given proportion.

Next step would to train the data with our model. For that we need to select the Train Model block and a classification model. If we observe the classification models on Microsoft Azure ML studio they are divided with multiclass, two-class and one-vs-all options. The iris dataset class has three options, class-setosa, iris-virginica and iris-versicolor. So, we have a multiclass data and we need to use the multiclassification option.

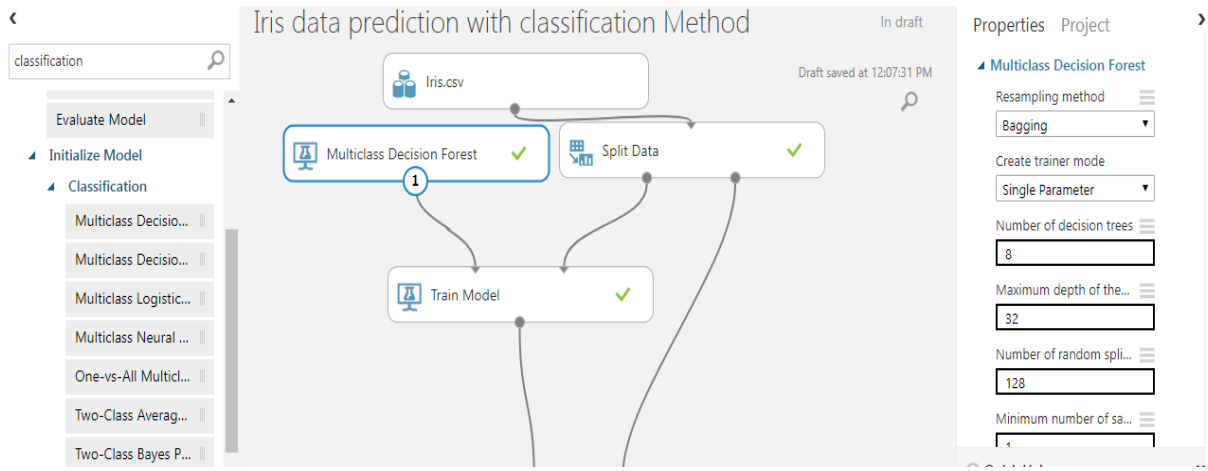


Figure 15. Selecting a classification model

As we can see from figure 15 we don't have the luxury of selection for multiclass classification option. We have four options and I picked Multiclass Decision forest as my first choice. We have an option to choose the number of the decision tree, the depth, the random split and so forth.

The next step would be to add the Score Model block and connect the trained data and the test data. This block tells us how the decision is made on the testing data and finally add the Evaluate Model block. The evaluate model tells us the accuracy (predictive strength) of our model. The final scheme is shown on figure 16.

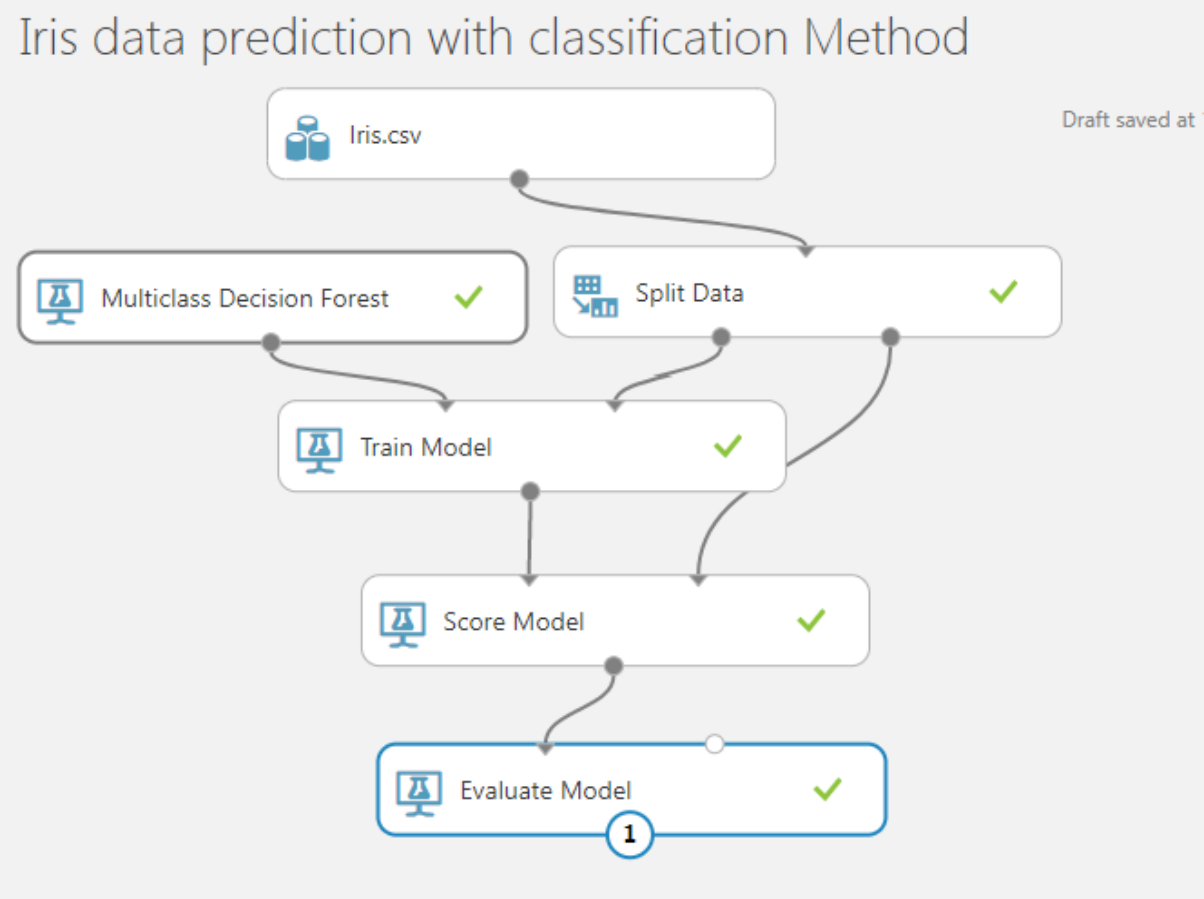


Figure 16. Final scheme of the model for Multiclass Decision Forest

Output of each steps

As we have already shown the output of the row data and the split data, I start to show the output of the data starting from Train Model. Since we set the Multiclass Decision Forest to have 8 different trees, we can see from figure 17 that, it has 8 different decision trees. The tree shown on the figure shows how the decision is made. On the first decision forest, it starts with the root node petal width and comparison criteria <=0.5 and goes on until it covers all the possible criteria for all classes.

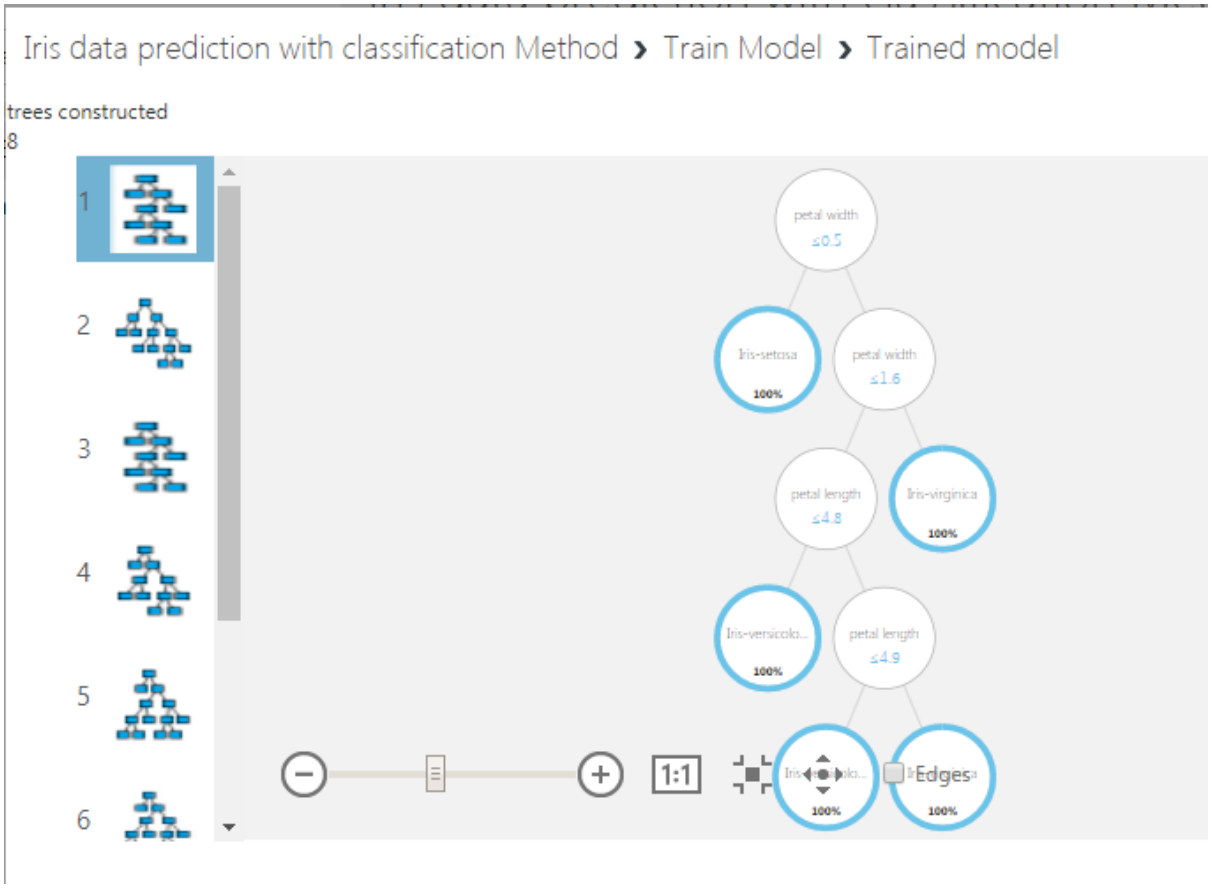


Figure 17. classification of a decision forest

Next let us see the out put of the Scored model. It shows how the scored labels are decided based the scored probabilities of each class for the testing set. Figure 18 shows that clearly with the frequency distribution of each classes for the testing set with the bar graph on the right.

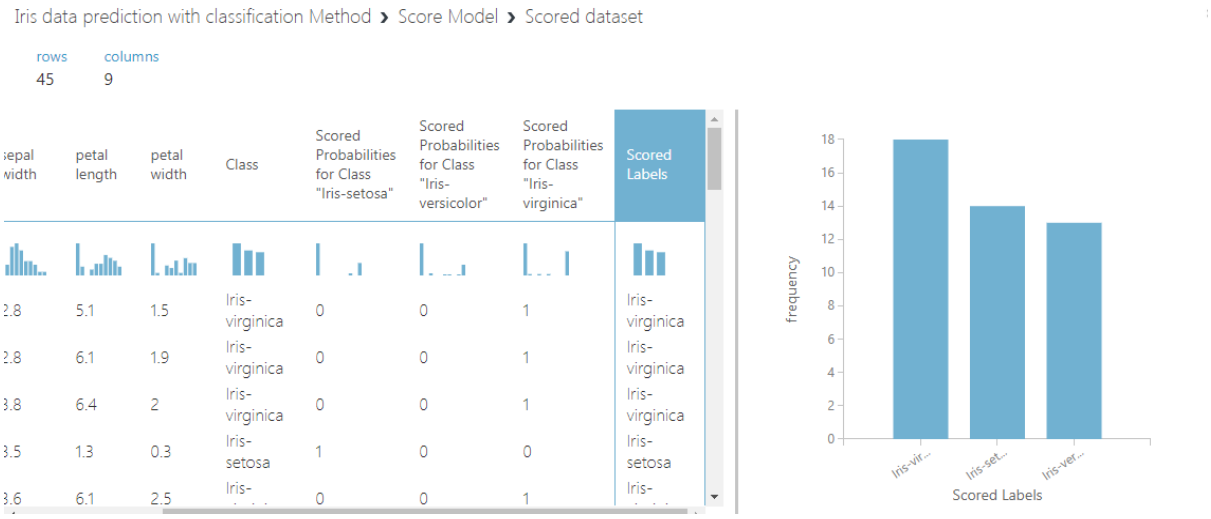


Figure 18. The scored model shows how many of the testing data are labelled to each class based on the scored probabilities.

And finally let us see the evaluation of the model. The evaluation model tells us how good the model predicted the new observation from the testing test. Here we have different evaluation matrix than the regression model above. Accuracy, precision, recall and confusion matrix are used in this case

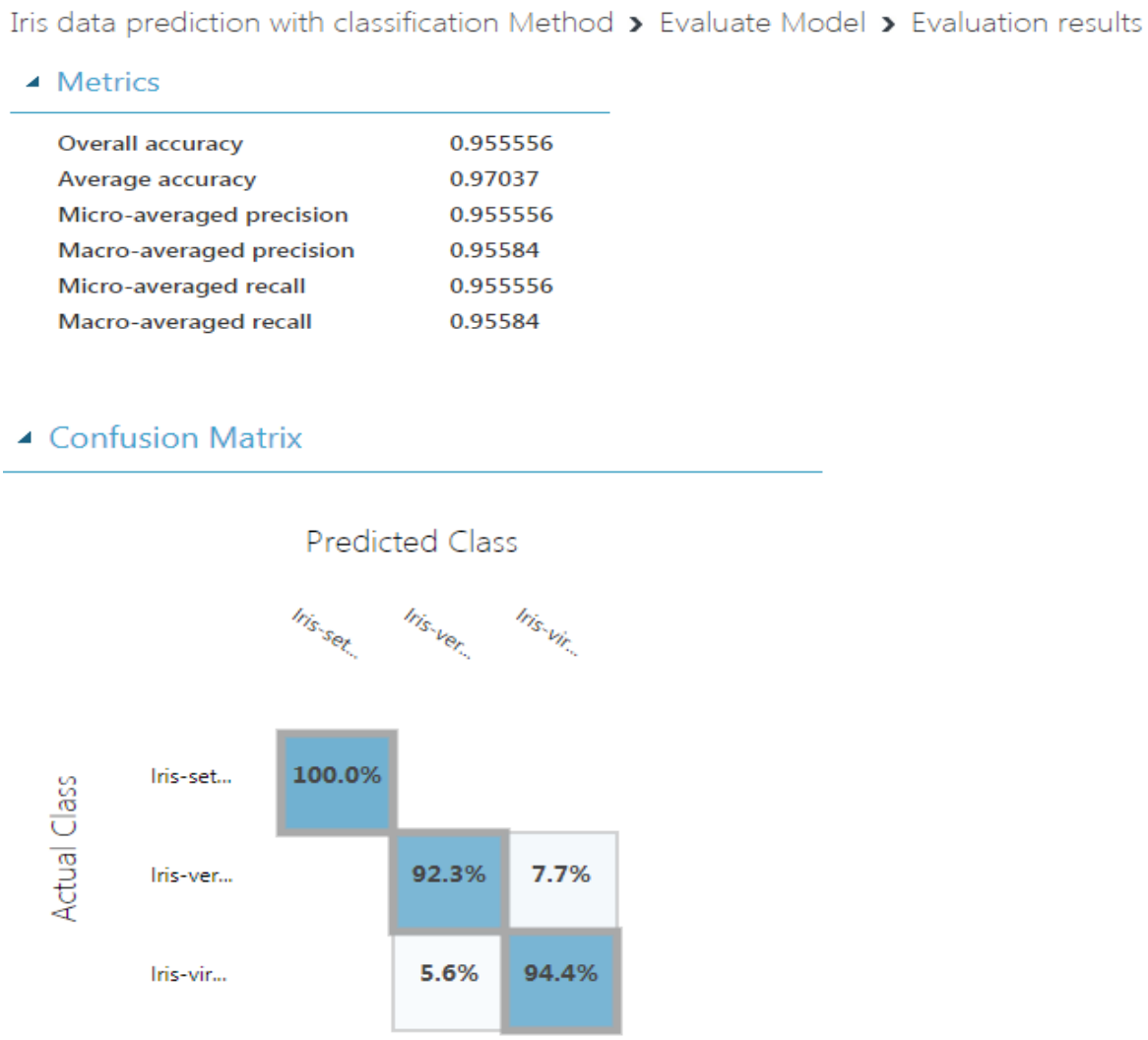


Figure 19. Evaluation matrix for decision forest.

As we can see from figure 19, the model predicted with overall accuracy of about 95.5 %. The most interesting evaluation is the confusion matrix. It even breaks each classes accuracy and we can see that iris-setosa has been classified with 100 %. The misclassification happens to be on the other two classes.

2.1 Multiclass Logistic Regression

We have the same steps as before while making the scheme. What we need to change is the model to logistic regression. The following figure shows the scheme of the logistic regression.

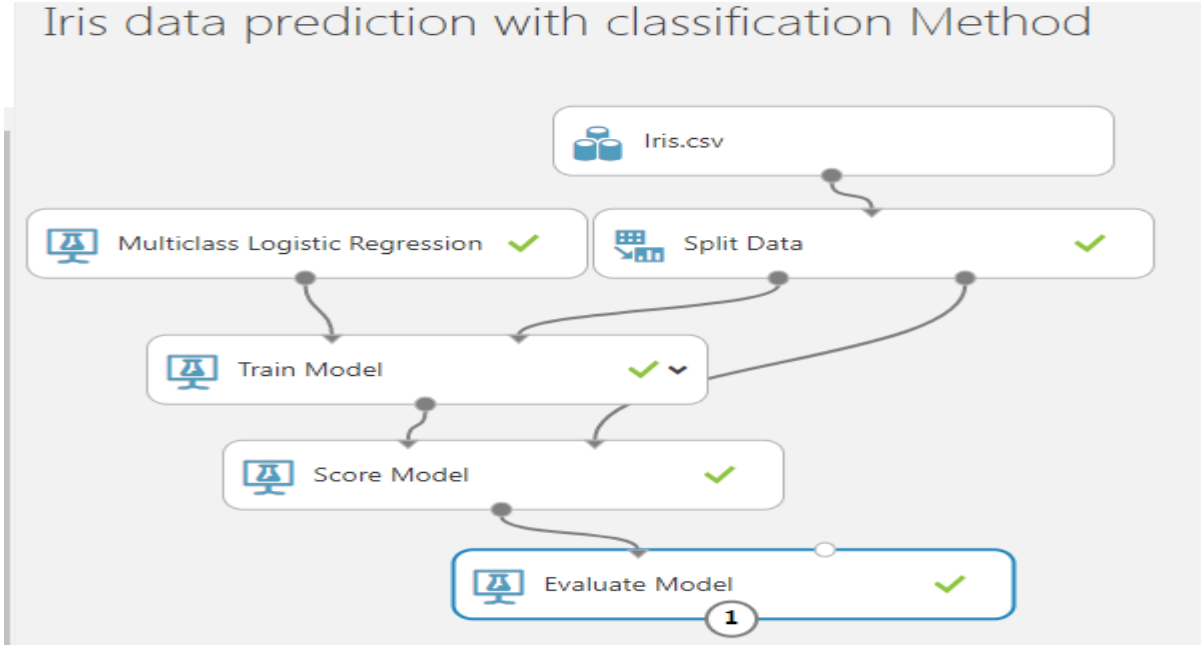


Figure 20. Scheme of Logistic regression

Next let us see the output results that are somehow different than the random forest classification method. In the training model output we can see the contribution of each attributes, which is the weight of each attributes.

Feature Weights

Feature	Iris-setosa	Iris-versicolor	Iris-virginica
petal length	-2.47124	0	1.5809
Bias	1.77043	0.560902	-2.33133
petal width	-2.24061	0	2.18588
sepal length	-0.777635	0	0.27363
sepal width	0.776467	-0.663428	0

Figure 21. The weight of each attributes

The Score Model block out put shows us how the score decision is made based on the probability scores. The visualization option on the right even gives a better insight. For the given testing set the model only misclassifies one observation. Figure 22 below shows the detail.

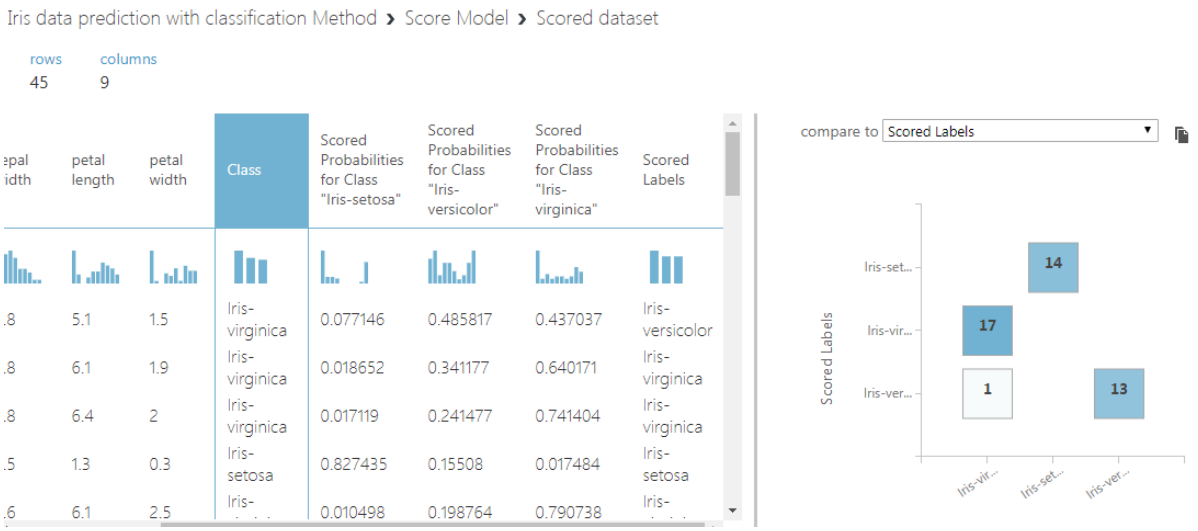


Figure 22. The score Model output shows how the classification is done based on the probability scores.

The Evaluation Model shows the accuracy, recall, precession and confusion matrix. We can see that this model classifies this data better than the random forest classification with an accuracy of 97.7%.

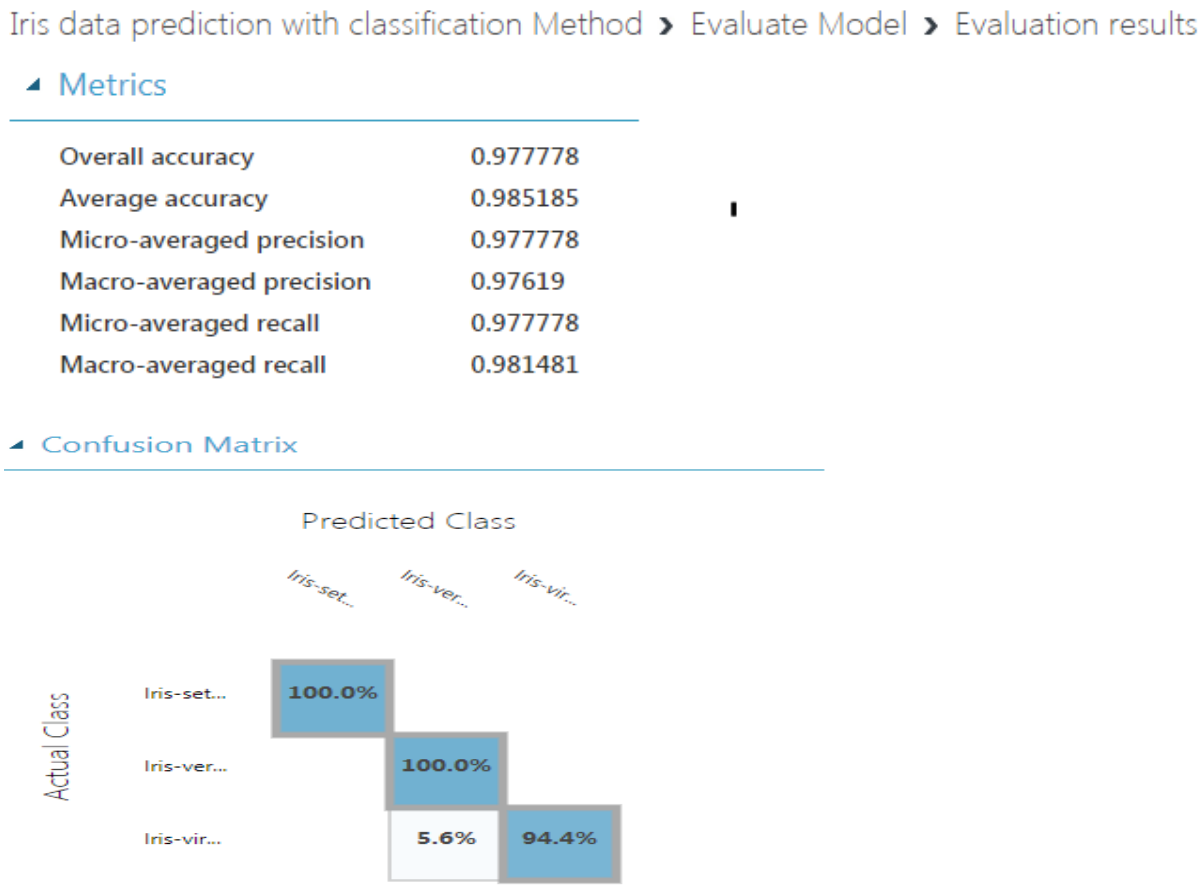


Figure 23. Evaluation matrices of Logistic regression model

Produce an API code

The API code can produce by right click on the data, generate data access code, with options with python and R.

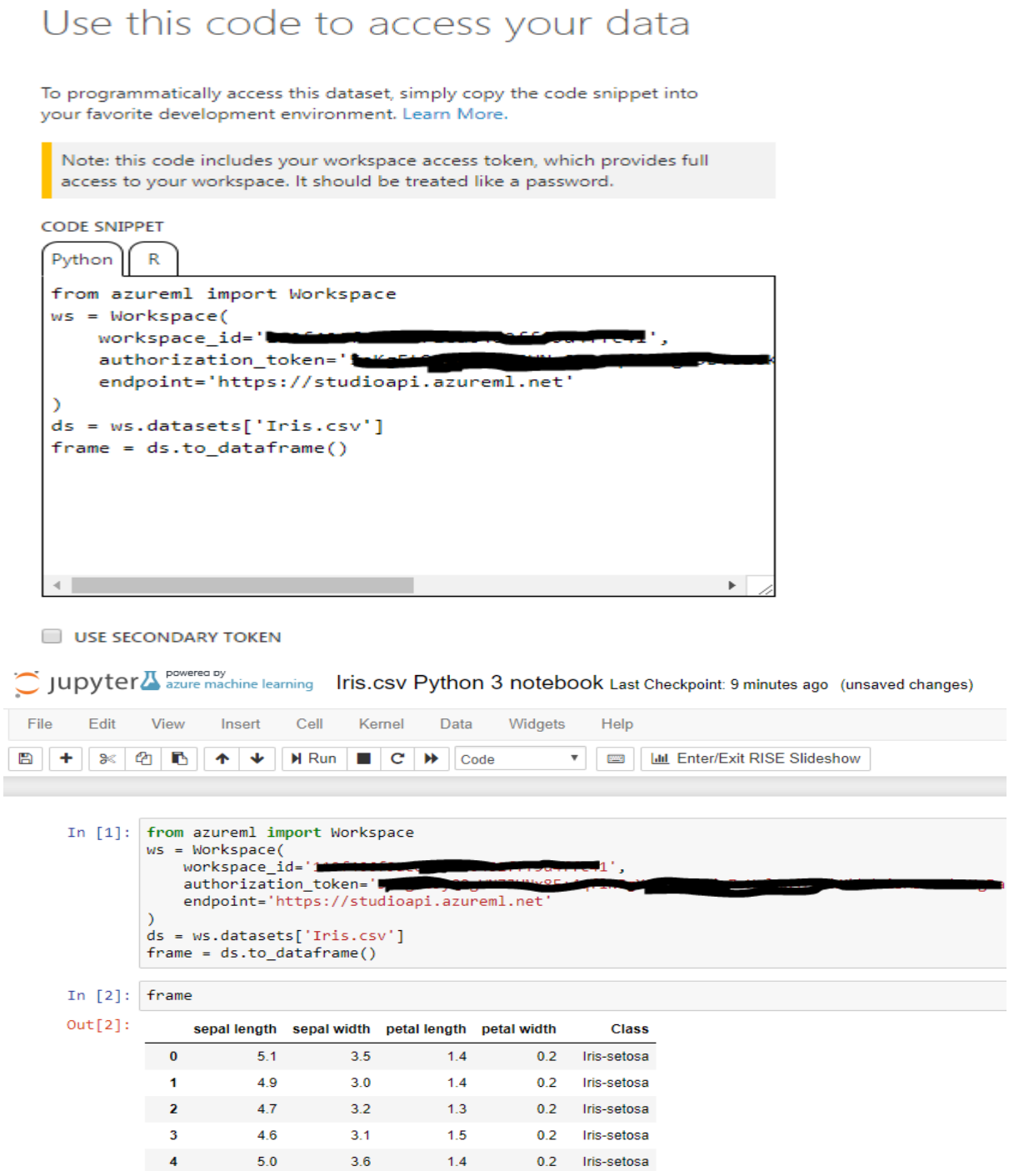


Figure 24. API code generation for python

Comparing two models

The cool thing about Microsoft Azure ML studio is the comparison option. If we observe the Evaluate Model block, it has two ports that allows a comparison of two models. Taking this advantage, we don't need to do all the steps that we did all over again. By sharing the same split data, we train them with two different models. As the second choice I took Multiclass Logistic Regression. The Scheme of the comparison is shown on figure 25.

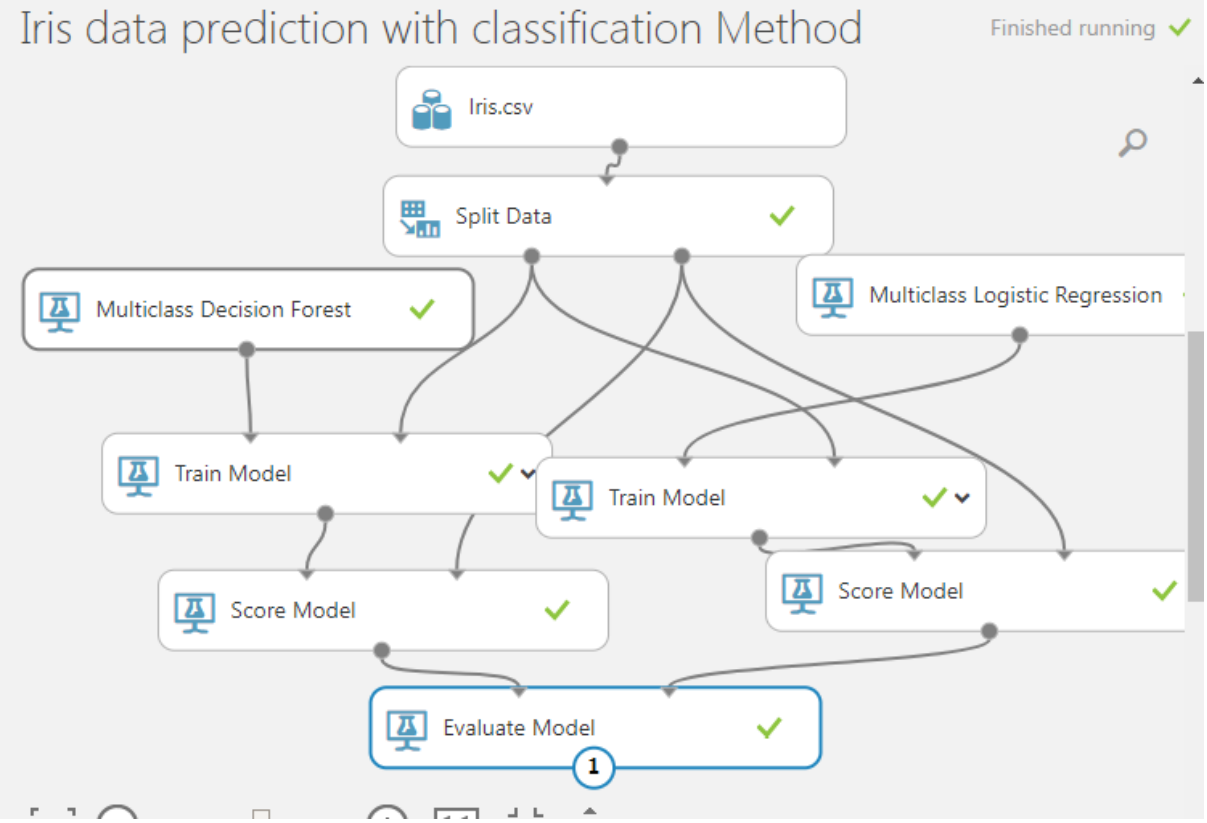


Figure 25. The scheme of the comparison of the two models

Output for the comparison

The output of each steps similar, the interesting part that we should observe is on the Evaluate model part. The following figure 26 illustrates the comparison of Multiclass decision forest and Multiclass Logistic regression model.

As we can see from the comparison result shown on figure 26, the logistic regression model even predicted with a better average accuracy of 97.7 %. From the confusion matrix we can see that iris-setosa and iris-versicolor are classified perfectly on the case of logistic regression. The misclassification occurs iris-virginica. From the result we can conclude that the logistic regression performs well on this dataset. Though, we must note that we only have 150 observations on this

dataset and it considerably small. Using this model might lead to overfitting. The reason behind achieving such high accuracy level might be a consequence of overfitting.

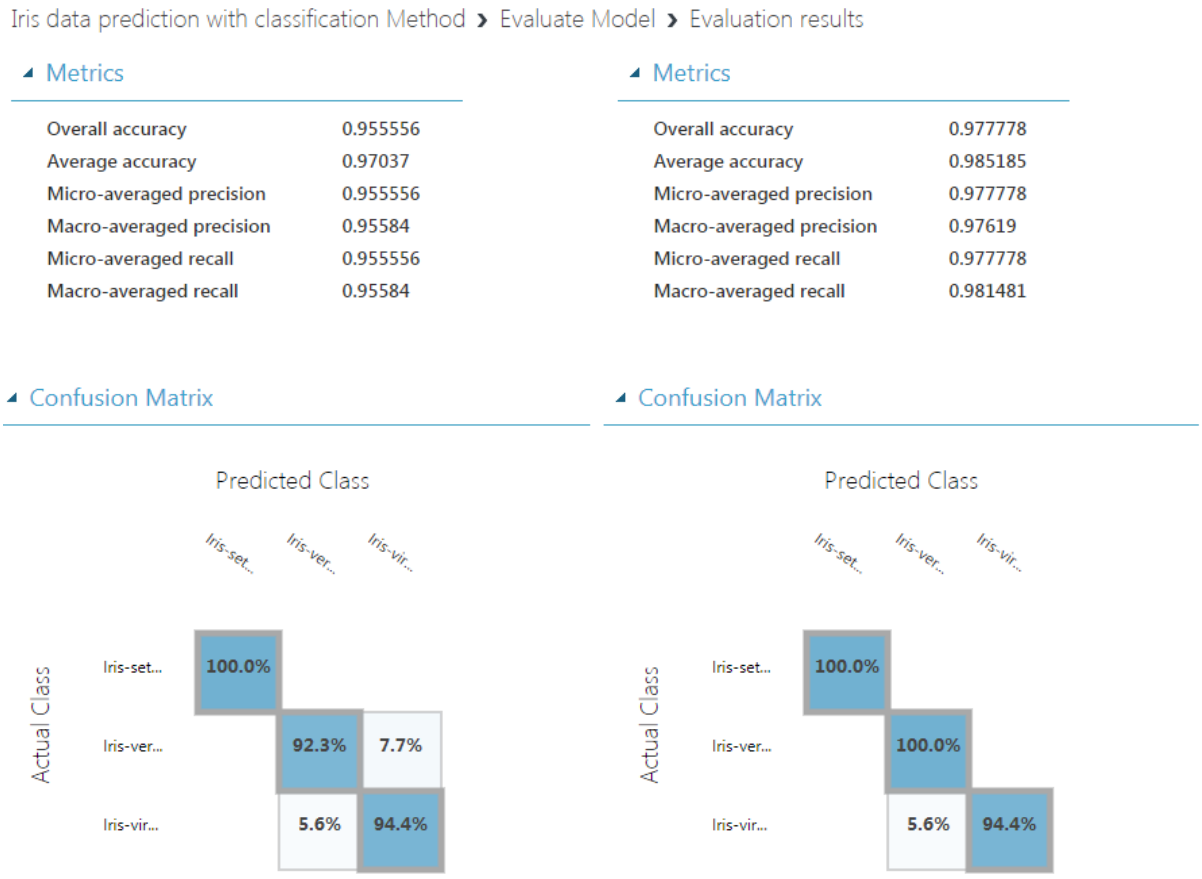


Figure 26. Comparison of decision forest and Logistic Regression Models

Figure 22. API code generation for this data

Task 3. Comparing the software used in the course.

In this course we used three different software. Two from IBM, SPSS and Watson Analytics and one form Microsoft, Microsoft Azure Machine Learning Studio.

SPSS- is a statistical software now acquired by IMB. It is widely used software in research, companies, education, marketing organization, data miners and so on. It is used for analysis of data with support of different machine learning algorithms.

The interface is very similar to Microsoft excel with some basic difference with data and variable windows are separated. Data analysis with SPSS tool needs some level of expertise. I found out that, interpreting the output is very difficult. Outputs are more of tabular form, and statistical background is needed to understand those terms.

Pros

It supports scripting languages R, Python, Sax Basic

It has both cloud and non-cloud support system

Data set has two-dimensional data and variables(simplifies defining data types)

Cons

It is not an open source software

The graphical outputs are relatively limited, default outputs are tabular and hard to interpret.

Constraints on internal structure of datatypes and data structure

Watson Analytics- is another IBM owned cloud-based analytics. It is created based on question/answer-based computing system.

Pros

Free 30 days trial subscription for anyone

Watson analytics selects the best algorithm for the data

No need of expertise on ML algorithm, any one can make the analysis

cons

It is not an open source service. Needs a cloud-based subscription

During analysis of a data, one has no control on choosing algorithm

Microsoft Azure ML Studio- It is one of Microsoft's cloud-based services.

Pros

It supports R and python languages and deployment as web service is possible.

Free for students

Cons

Need a cloud subscription, it is not open source

It supports different ML algorithms but limited numbers, for example for multiclass classification only four options are listed.

Conclusion

All the three software have their own targets users. If a company doesn't need a data analysis department, they can use Watson analysis service without needing an expertise on the field. For research-based analytics with some expertise one can use SPSS or Microsoft Azure ML studio or SPSS. If the company is not use cloud as service, SPSS might be their best choice for their data analysis task.

Personally, I like the Microsoft Azure ML studio. The simplicity of making the blocks with basic knowledge of ML algorithms, it's a cloud based capabilities, its support of python and R languages to be able to use your own algorithm in addition to the given build ones are the few points among many.