Implementing a Data Science Solution with Azure

By Sadique Mohammed – DFEDATA4

Part 1: Problem Domain Understanding

Question 1

Given the nature of the business you have been asked to describe the benefits of implementing the data science solution in the cloud compared to using on premise/hybrid data solutions.

The evaluation criteria should consider the following:

- security
- compliance
- scalability
- efficiency
- reliability
- fidelity
- flexibility
- portability

Please see table below for benefits of using a cloud data solution compared to on premise/hybrid data solution.

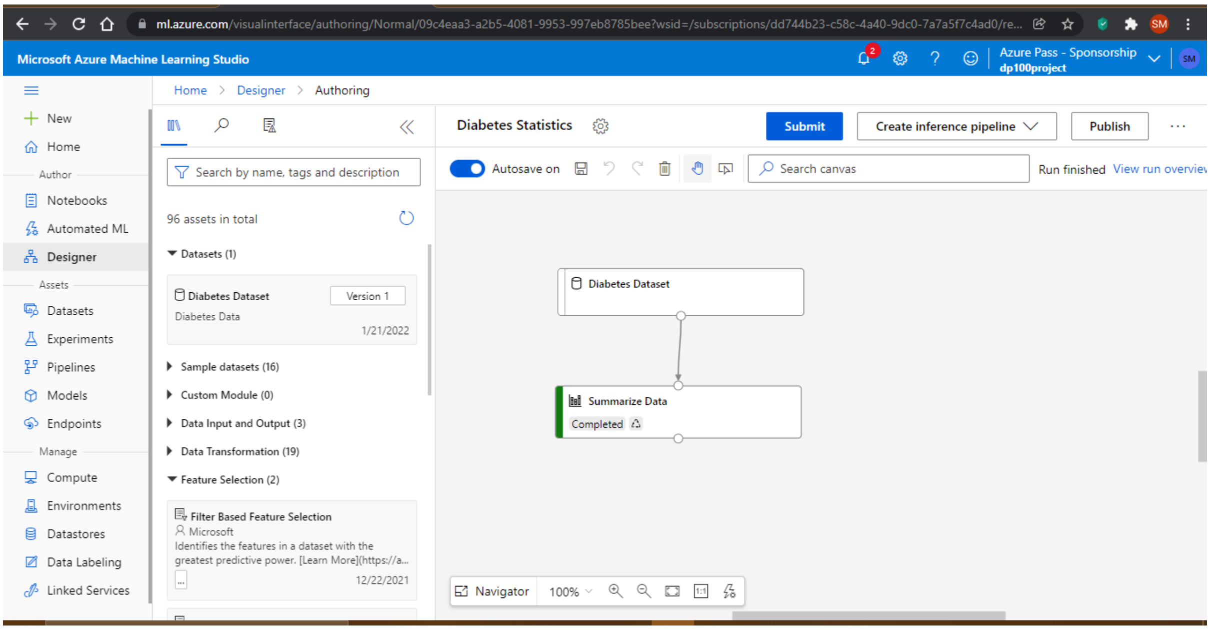
Cloud	**On Premise/Hybrid**
• On-demand scalability • Cost efficiency • Bundled capabilities such as IAM and analytics • Security • System uptime and availability	• Complete control over the tech stack • Local speed and performance • Governance and regulatory compliance

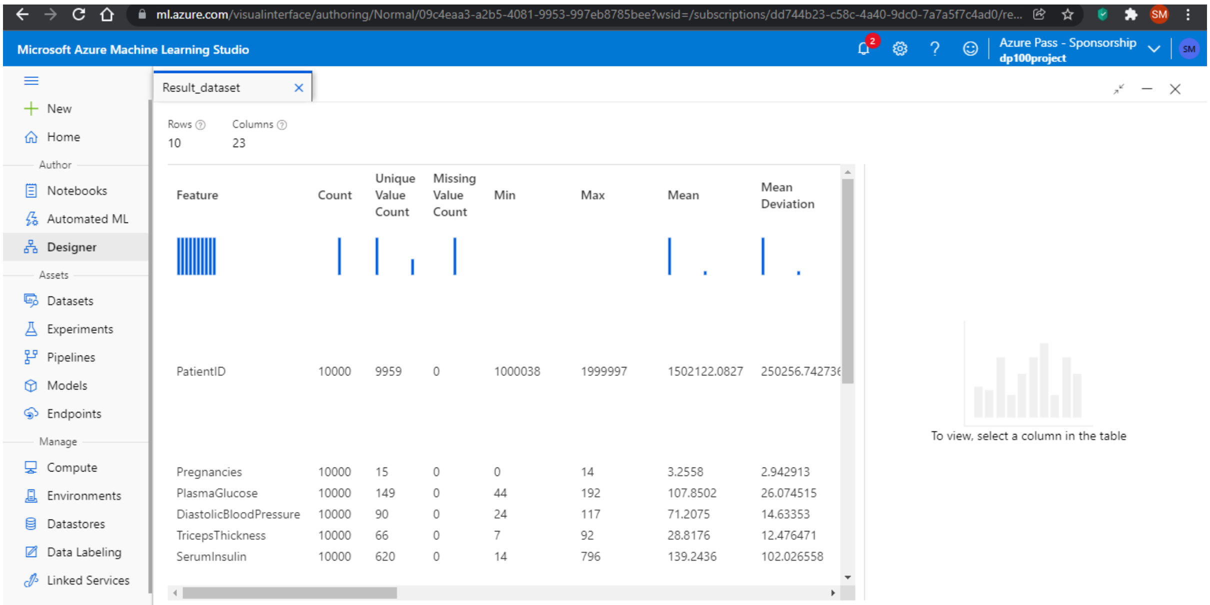
Part 2: Data Understanding

Question 1

Complete all the tasks below in Microsoft Azure Machine Learning Studio using the dataset identified for your project and discuss what the implications of the results for at least two of the tasks on the business questions that are candidates for your data science solution.

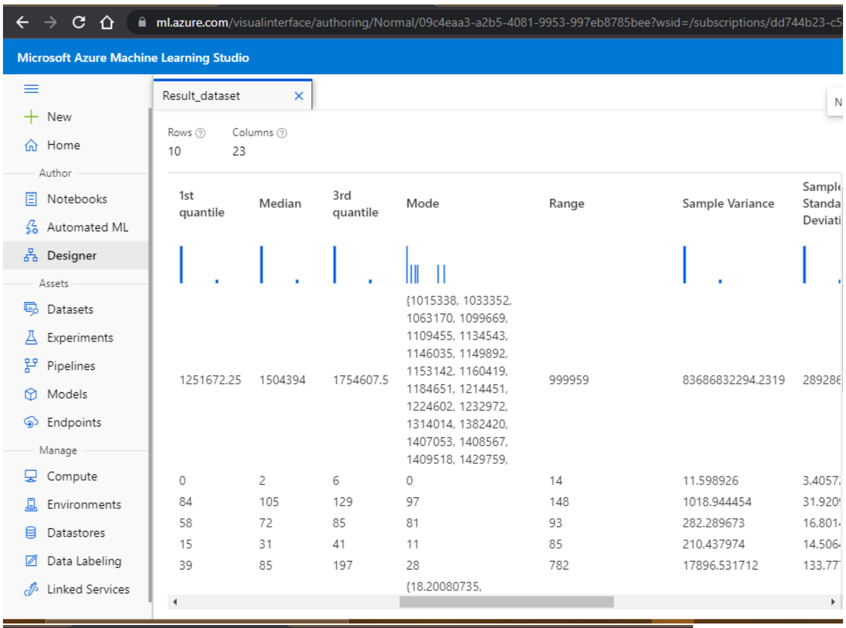
Task 1: Generate a descriptive statistics report for the columns in your dataset

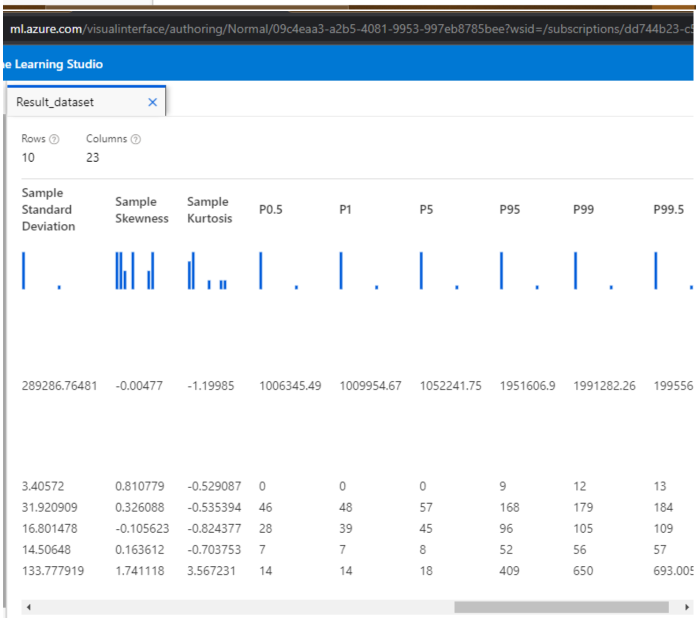




Above, you can see how I have obtained a statistical report for columns in the dataset in the Microsoft Azure Machine Learning Studio Designer.

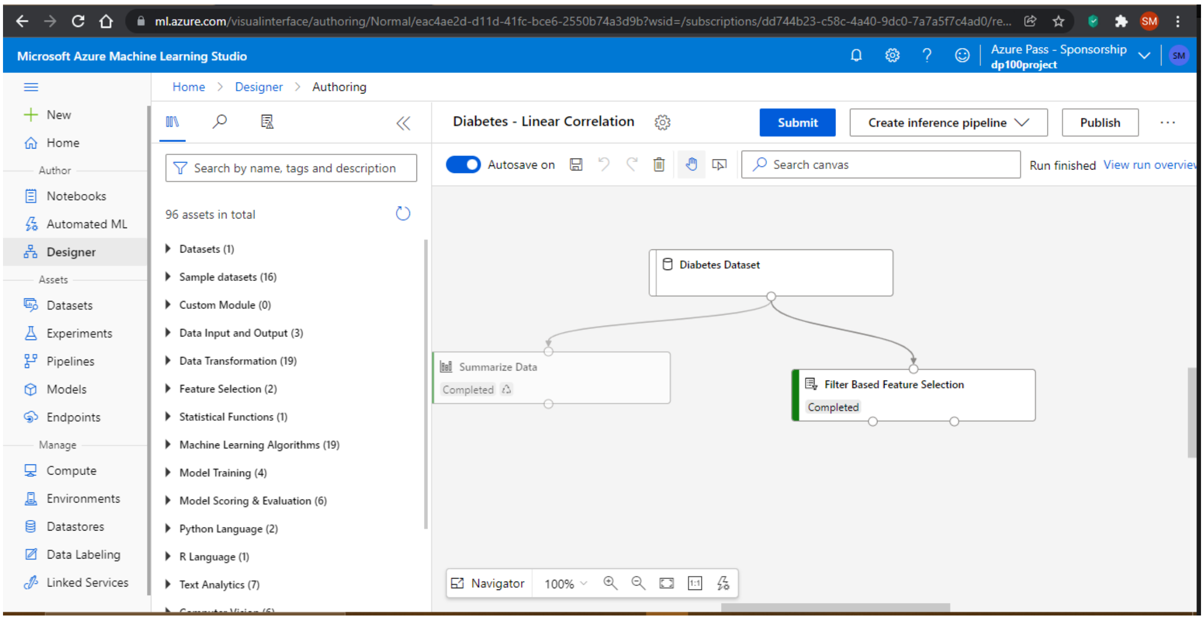
Task 2: Calculate a single statistical measure for each column which is useful for determining central tendency, dispersion, and shape of your dataset or whatever you deem necessary.



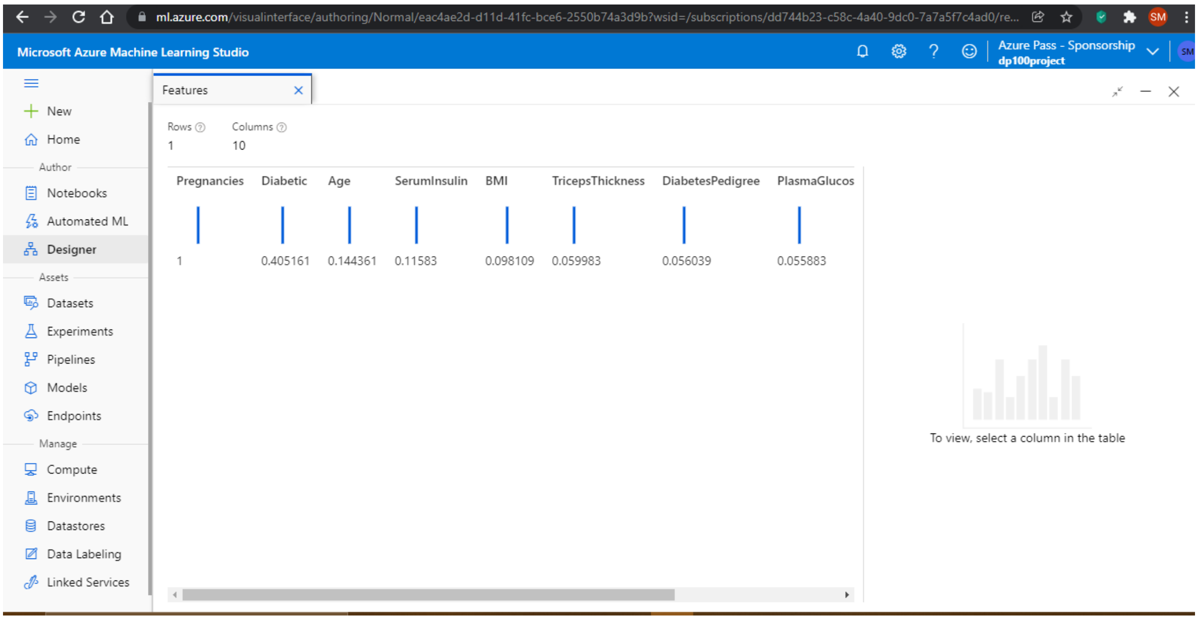


Please also see task 1 for single statistical measures for each column of dataset.

Task 3: Calculate the linear correlation between column values in your dataset



Above model is used to obtain the linear correlation between column values in dataset.

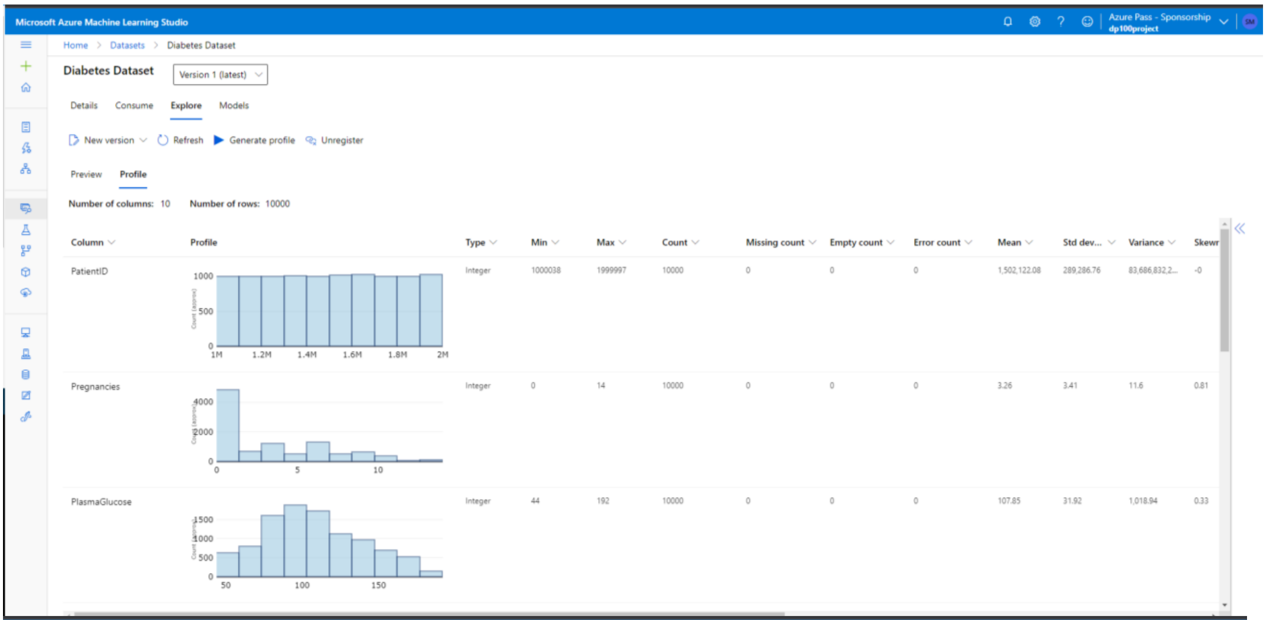


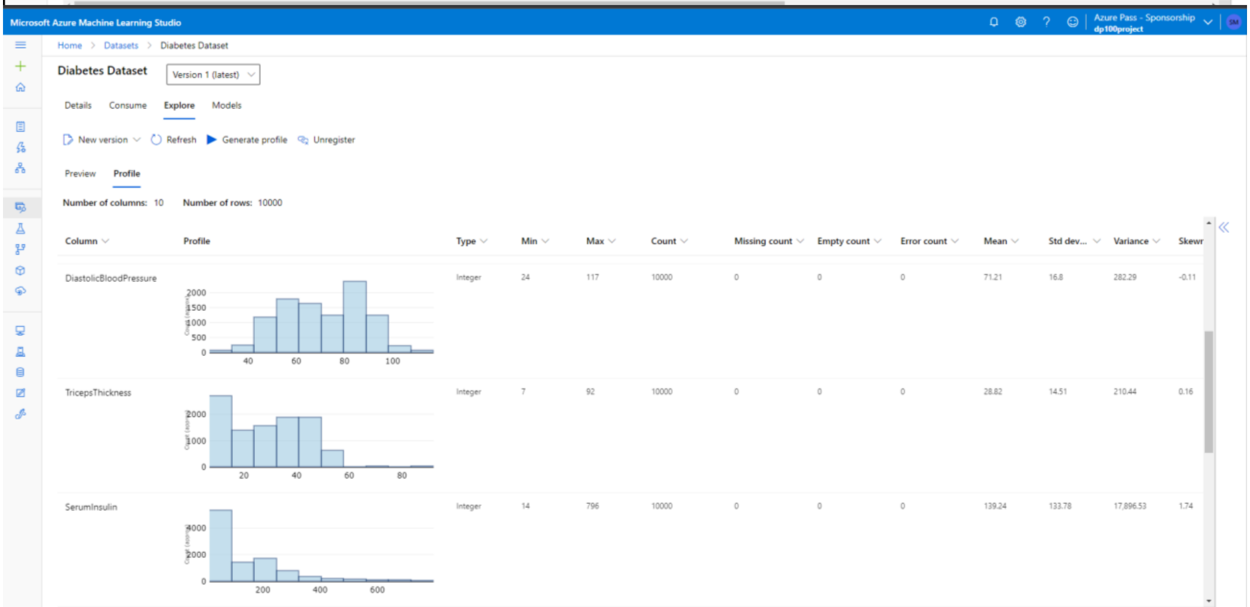
As you can see from above screenshot that Diabetic, Age, Seruminsulin and BMI are all positively correlated.

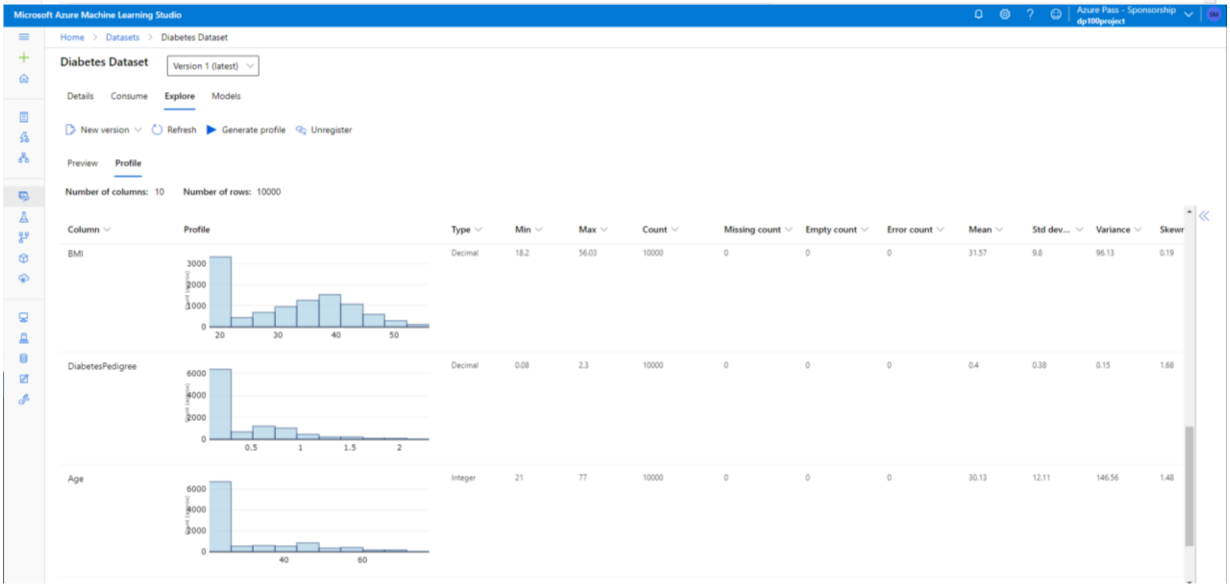
Task 4: Identify any data quality issues that may impact analysis

There are currently no data quality issues with this data sets, it is a pure dataset. There are no missing values or outliers to list. Also, there are no Nulls in this dataset.

Task 5: Conduct exploratory visual analysis of the dataset and comment on at least 2 of your key findings.


Microsoft Azure Machine Learning Studio
Azure Pass - Sponsorship
dp100project
Home > Datasets > Diabetes Dataset
Diabetes Dataset
Version 1 (latest)
Details Consume Explore Models
New version Refresh Generate profile Unregister
Preview Profile
Number of columns: 10 Number of rows: 10000
Column Profile Type Min Max Count Missing count Empty count Error count Mean Std dev... Variance Skew
PatientID Integer 1000038 1999997 10000 0 0 0 1,502,122.08 289,286.76 83,686,832,2... -0
Pregnancies Integer 0 14 10000 0 0 0 3.26 3.41 11.6 0.81
PlasmaGlucose Integer 44 192 10000 0 0 0 107.85 31.92 1,018.94 0.33


Microsoft Azure Machine Learning Studio
Azure Pass - Sponsorship
dp100project
Home > Datasets > Diabetes Dataset
Diabetes Dataset
Version 1 (latest)
Details Consume Explore Models
New version Refresh Generate profile Unregister
Preview Profile
Number of columns: 10 Number of rows: 10000
Column Profile Type Min Max Count Missing count Empty count Error count Mean Std dev... Variance Skew
DiastolicBloodPressure Integer 24 117 10000 0 0 0 71.21 16.8 282.29 -0.11
TricepsThickness Integer 7 92 10000 0 0 0 28.82 14.51 210.44 0.16
SerumInsulin Integer 14 796 10000 0 0 0 139.24 133.78 17,896.53 1.74


Microsoft Azure Machine Learning Studio
Azure Pass - Sponsorship
dp100project
Home > Datasets > Diabetes Dataset
Diabetes Dataset
Version 1 (latest)
Details Consume Explore Models
New version Refresh Generate profile Unregister
Preview Profile
Number of columns: 10 Number of rows: 10000
Column Profile Type Min Max Count Missing count Empty count Error count Mean Std dev... Variance Skew
BMI Decimal 18.2 56.03 10000 0 0 0 31.57 9.8 96.13 0.19
DiabetesPedigree Decimal 0.08 2.3 10000 0 0 0 0.4 0.38 0.15 1.68
Age Integer 21 77 10000 0 0 0 30.13 12.11 146.56 1.48

From the visual analysis above, all columns in dataset have a positive skewness. The maximum BMI value is 56.03 and minimum BMI is 18.2. Also, the mean age of the sample in this data is 30.13 yrs, which is 30 yrs.

Note: Tasks may be completed by identifying and using the relevant modules in Designer.

Part 3: Data Preparation and Transformation

Question 1:

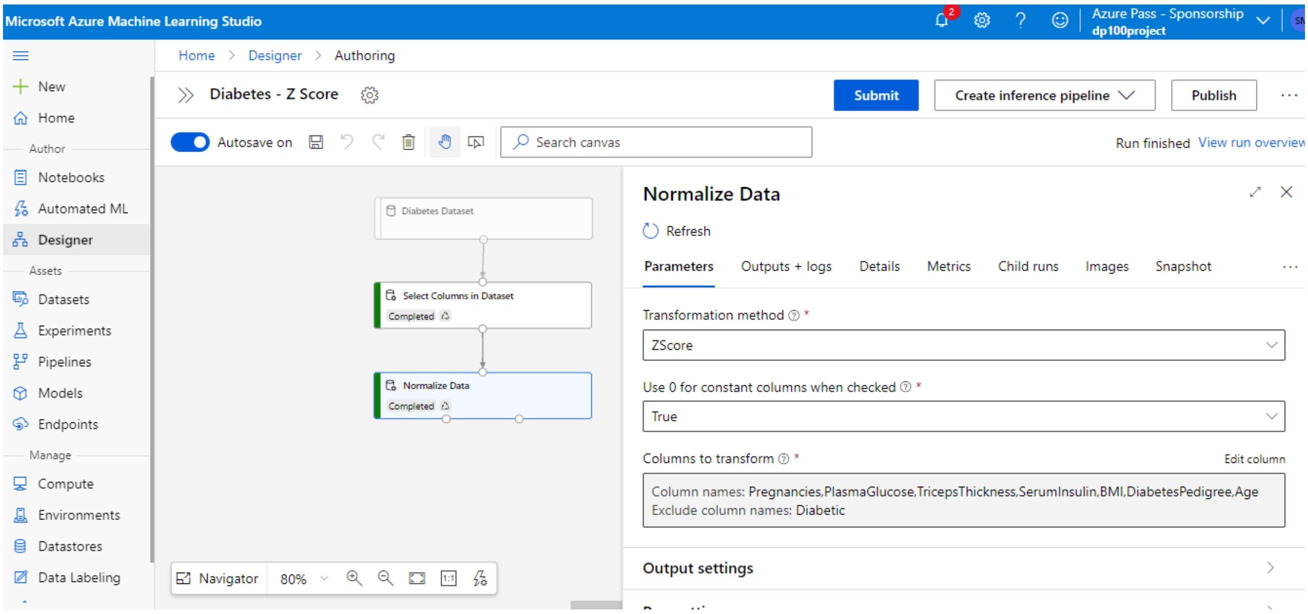
Discuss and apply the appropriate data processing techniques to address the data quality issues you identified in your exploratory analysis of the data.

No data quality issues were identified in the exploratory analysis of the data. All values in dataset are numerical and there is not any categorical values in dataset.

Question 2:

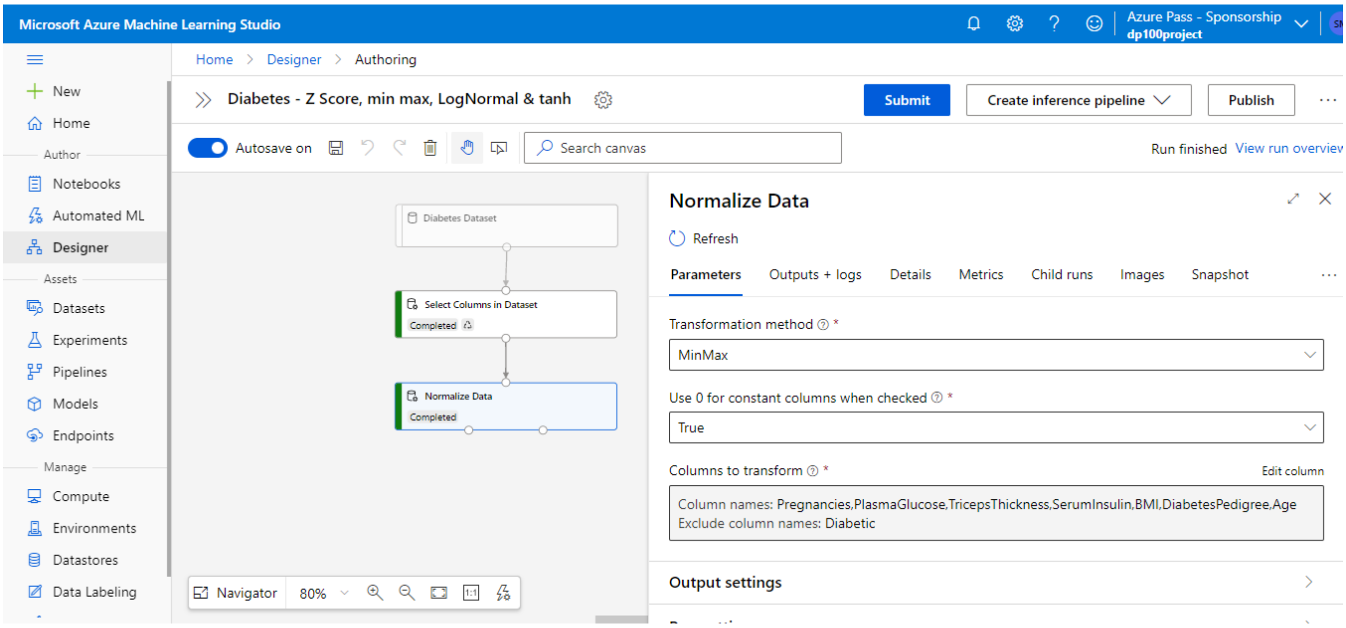
Conduct a comparative analysis of the mathematical techniques below for data normalization and apply the relevant one to your dataset.

- z-score



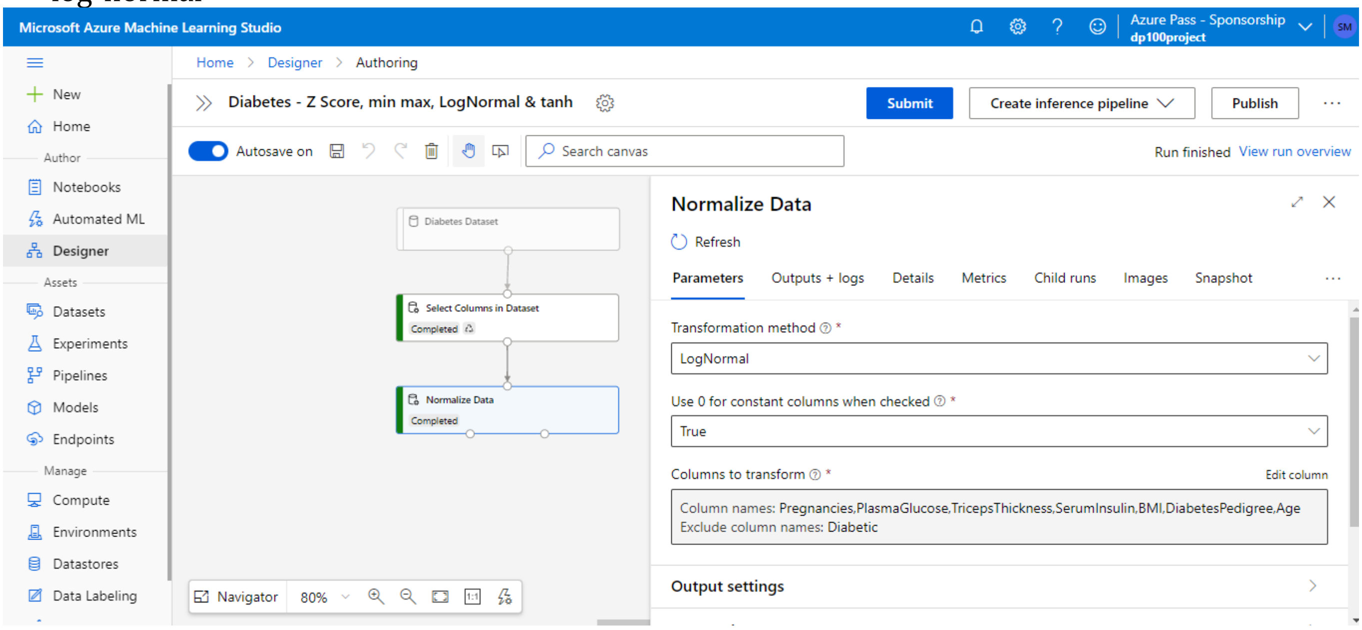
Above shows process of obtaining the Z Score.

- min-max



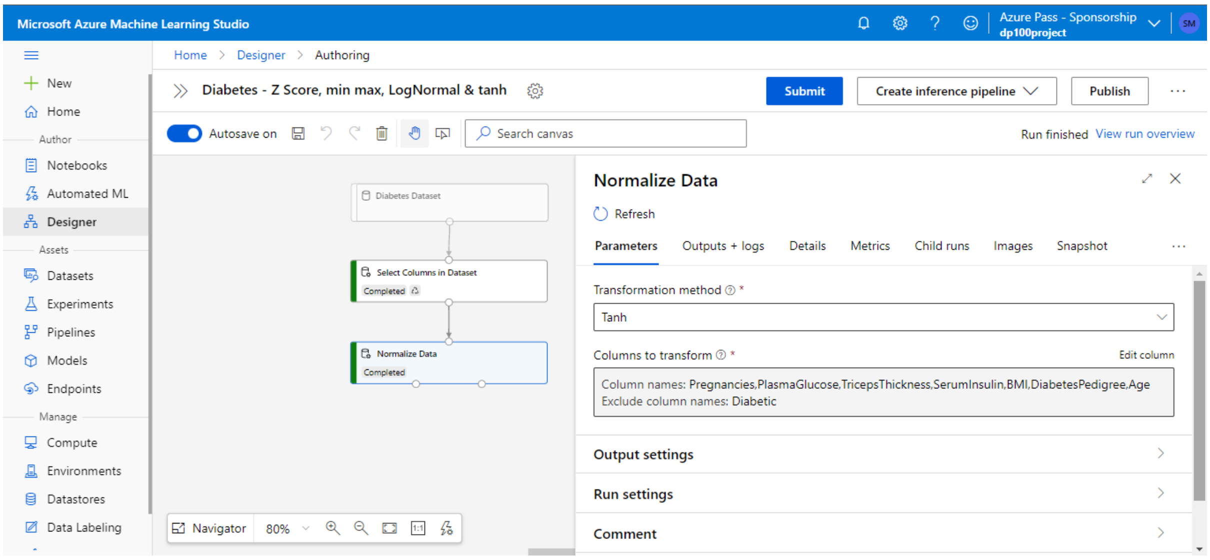
Above shows process of obtaining MinMax.

- log-normal



Above shows process of obtaining log-normal

- tanh



Above shows process of obtaining tanh.

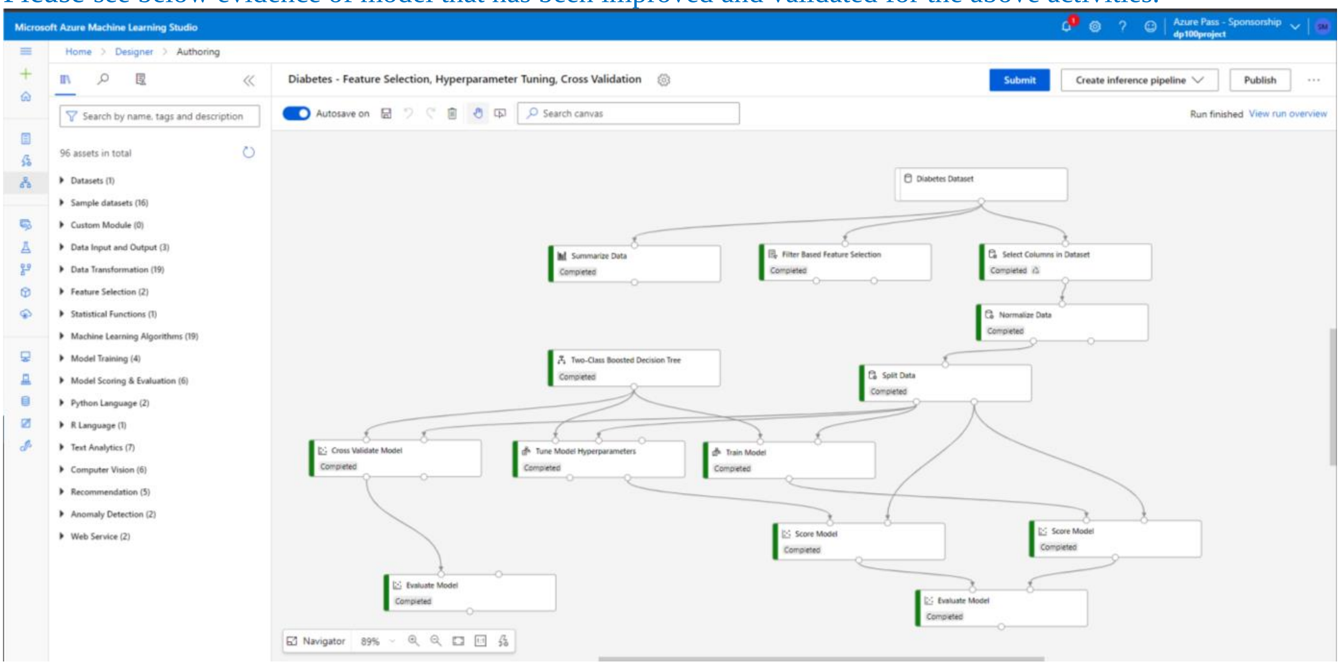
Part 4: Model Validation and Improvement

Question 1:

Discuss and evidence how your model can be improved and validated through the following activities.

- Feature Selection
- Hyperparameter Tuning
- Cross Validation

Please see below evidence of model that has been improved and validated for the above activities.



Part 5: Evaluation and Metrics

Question 1:

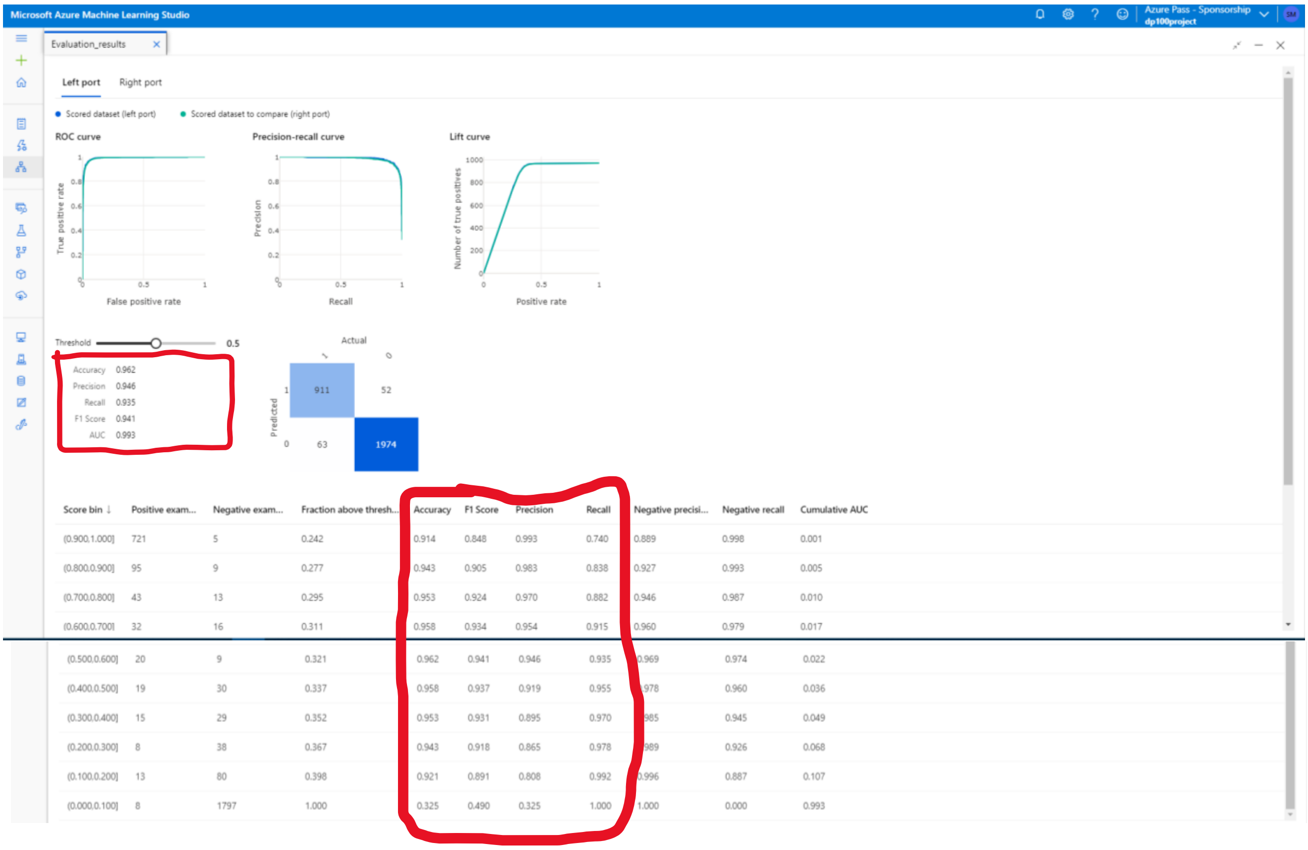
Critically evaluate the metrics for classification models below and identify which of the statistical measures are suitable for the evaluation of your data science solution. The selected metric should be used in the evaluation of your model.

Classification Metrics:

- Accuracy
- Precision
- Recall
- F1 Score

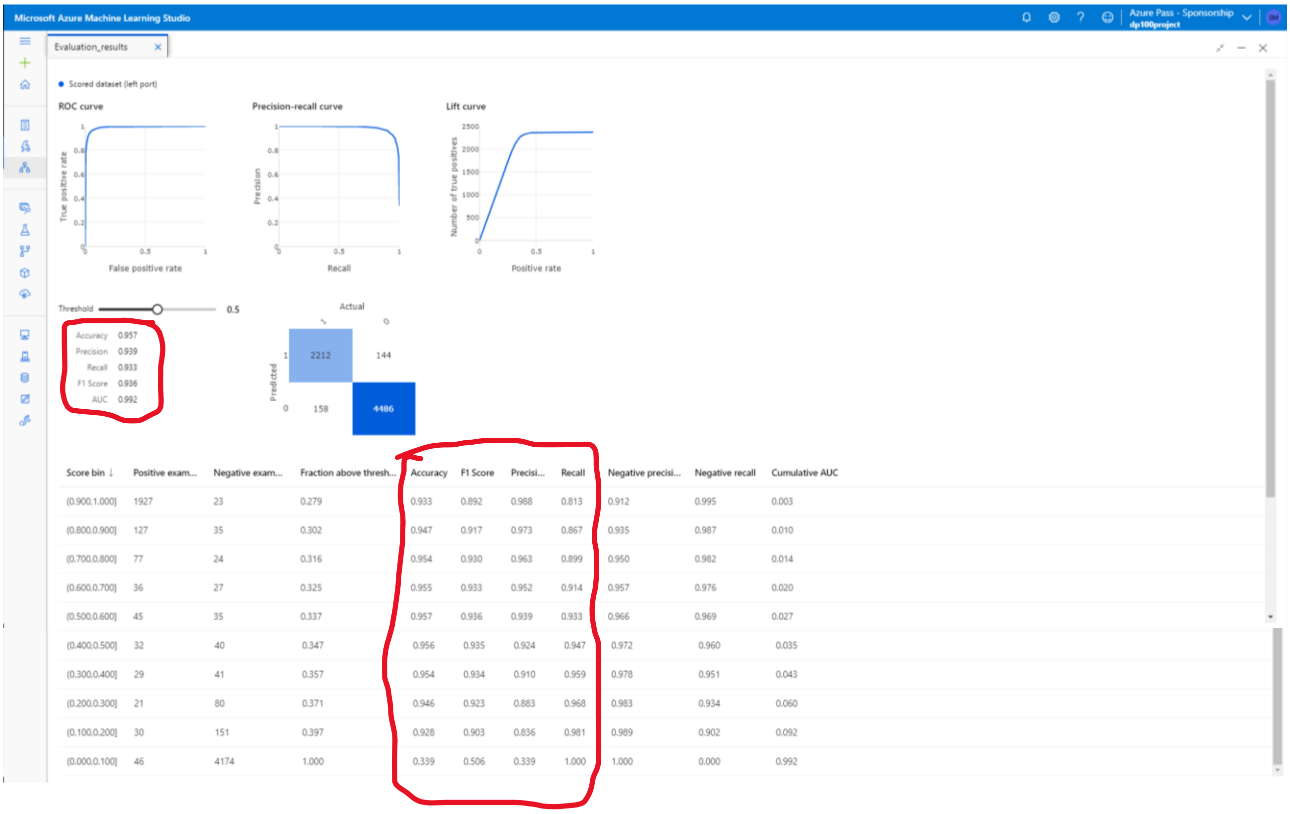
Using the designer model from Part 4, I was able to obtain the above classification metrics via the evaluate model.

Below shows a screenshot of classification metrics via the hyperparameter tuning evaluate model.



Score bin ↓	Positive exam...	Negative exam...	Fraction above thresh...	Accuracy	F1 Score	Precision	Recall	Negative precisi...	Negative recall	Cumulative AUC
(0.900,1.000]	721	5	0.242	0.914	0.848	0.993	0.740	0.889	0.998	0.001
(0.800,0.900]	95	9	0.277	0.943	0.905	0.983	0.838	0.927	0.993	0.005
(0.700,0.800]	43	13	0.295	0.953	0.924	0.970	0.882	0.946	0.987	0.010
(0.600,0.700]	32	16	0.311	0.958	0.934	0.954	0.915	0.960	0.979	0.017
(0.500,0.600]	20	9	0.321	0.962	0.941	0.946	0.935	0.969	0.974	0.022
(0.400,0.500]	19	30	0.337	0.958	0.937	0.919	0.955	.978	0.960	0.036
(0.300,0.400]	15	29	0.352	0.953	0.931	0.895	0.970	985	0.945	0.049
(0.200,0.300]	8	38	0.367	0.943	0.918	0.865	0.978	989	0.926	0.068
(0.100,0.200]	13	80	0.398	0.921	0.891	0.808	0.992	0.996	0.887	0.107
(0.000,0.100]	8	1797	1.000	0.325	0.490	0.325	1.000	1.000	0.000	0.993

Below shows a screenshot of classification metrics via the cross validation evaluate model.

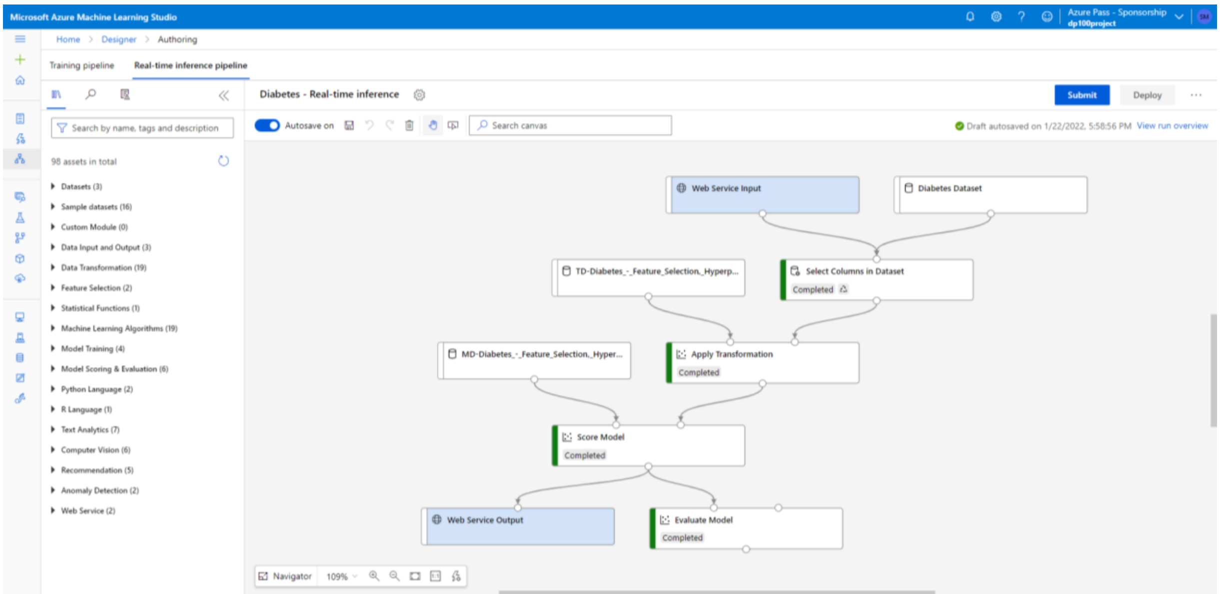


Part 6: Deployment

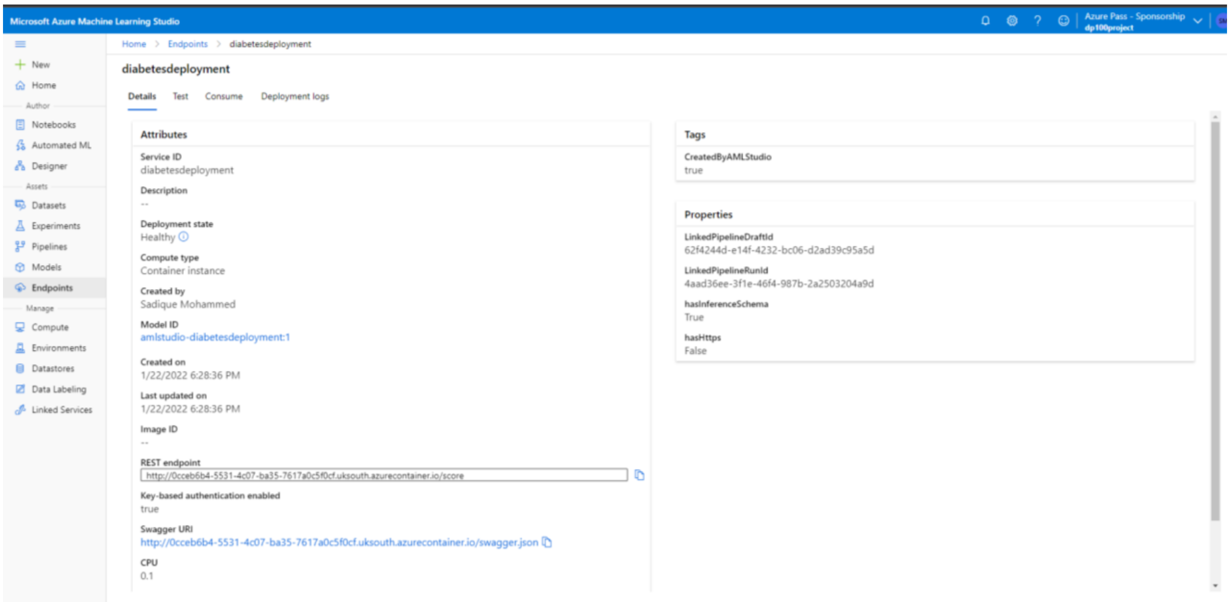
Question 1:

Deploy your model as a real-time inferencing service.

Below shows a model for a inference pipeline for real-time inferencing to be deployed.



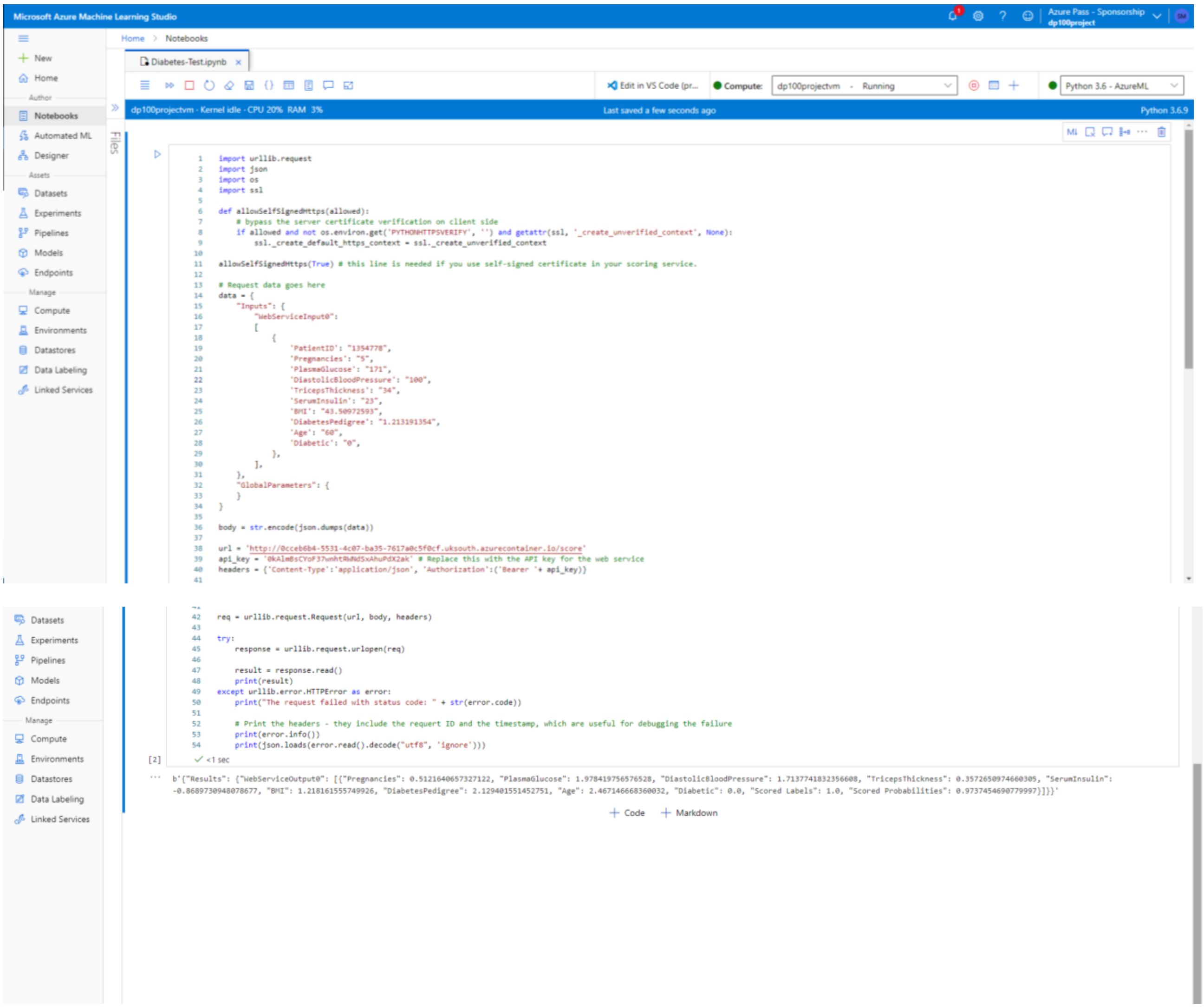
Below shows deployment of Diabetes dataset as a real time inferencing service.



Question 2:

Write a simple test script in Jupyter Notebook to consume the service.

Please see below simple test script in Jupyter Notebook to consume the service.



Python Script in Jupyter Notebook

```
import urllib.request
import json
import os
import ssl

def allowSelfSignedHttps(allowed):
    # bypass the server certificate verification on client side
    if allowed and not os.environ.get('PYTHONHTTPSVERIFY', '') and getattr(ssl, '_create_unverified_context', None):
        ssl._create_default_https_context = ssl._create_unverified_context
```

```
allowSelfSignedHttps(True) # this line is needed if you use self-
signed certificate in your scoring service.

# Request data goes here
data = {
    "Inputs": {
        "WebServiceInput0":
        [
            {
                'PatientID': "1354778",
                'Pregnancies': "5",
                'PlasmaGlucose': "171",
                'DiastolicBloodPressure': "100",
                'TricepsThickness': "34",
                'SerumInsulin': "23",
                'BMI': "43.50972593",
                'DiabetesPedigree': "1.213191354",
                'Age': "60",
                'Diabetic': "0",
            },
        ],
    },
    "GlobalParameters": {
    }
}

body = str.encode(json.dumps(data))

url = 'http://0cceb6b4-5531-4c07-ba35-
7617a0c5f0cf.uksouth.azurecontainer.io/score'
api_key = '0kAlmBsCYoF37wnhtRWNdSxAhuPdX2ak' # Replace this with the API key for t
he web service
headers = {'Content-
Type':'application/json', 'Authorization':('Bearer '+ api_key)}

req = urllib.request.Request(url, body, headers)

try:
    response = urllib.request.urlopen(req)

    result = response.read()
    print(result)
except urllib.error.HTTPError as error:
    print("The request failed with status code: " + str(error.code))

    # Print the headers - they include the requert ID and the timestamp, which are
 useful for debugging the failure
    print(error.info())
    print(json.loads(error.read().decode("utf8", 'ignore')))
```