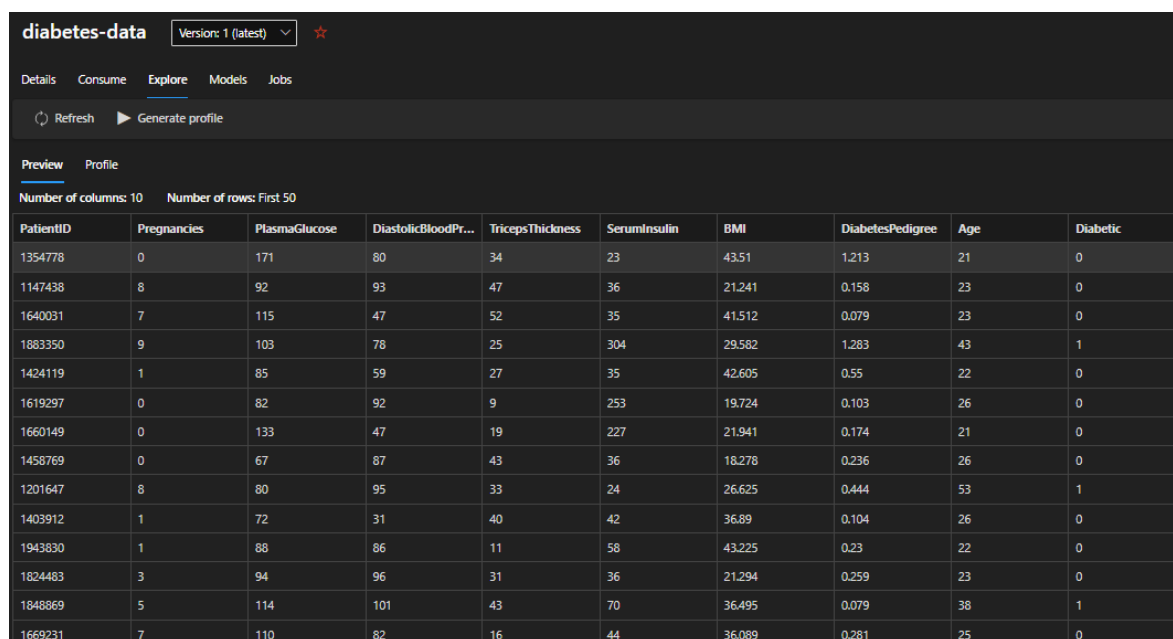


Predictive model – Classification and diagnosis of diabetes.

The objective of this Project is to develop a machine learning model that can do predictions based on prior training and evaluation. The **target feature** is the **diagnosis**, which can be positive or negative; therefore, I'll train and deploy the following classification model: **Two-Class Logistic Regression**. The entire project is hosted on the Azure cloud infrastructure.

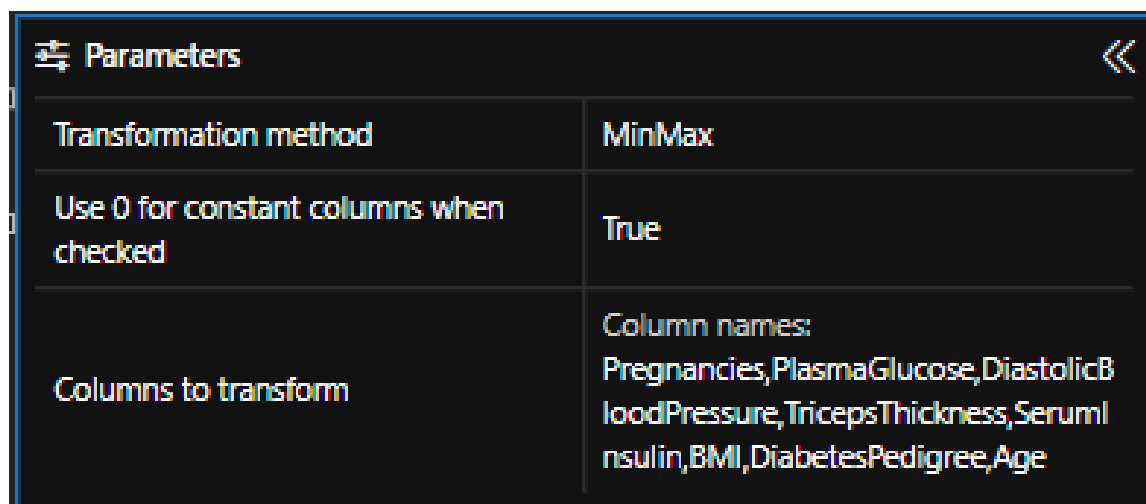
Dataset

The dataset that I'll analyze is a medical record of 10 k people with 9 features



PatientID	Pregnancies	PlasmaGlucose	DiastolicBloodPr...	TricepsThickness	SerumInsulin	BMI	DiabetesPedigree	Age	Diabetic
1354778	0	171	80	34	23	43.51	1.213	21	0
1147438	8	92	93	47	36	21.241	0.158	23	0
1640031	7	115	47	52	35	41.512	0.079	23	0
1883350	9	103	78	25	304	29.582	1.283	43	1
1424119	1	85	59	27	35	42.605	0.55	22	0
1619297	0	82	92	9	253	19.724	0.103	26	0
1660149	0	133	47	19	227	21.941	0.174	21	0
1458769	0	67	87	43	36	18.278	0.236	26	0
1201647	8	80	95	33	24	26.625	0.444	53	1
1403912	1	72	31	40	42	36.89	0.104	26	0
1943830	1	88	86	11	58	43.225	0.23	22	0
1824483	3	94	96	31	36	21.294	0.259	23	0
1848869	5	114	101	43	70	36.495	0.079	38	1
1669231	7	110	82	16	44	36.089	0.281	25	0

The first transformation is to normalize the columns in the follow box











Parameters

Transformation methodMinMax

Use 0 for constant columns when checkedTrue

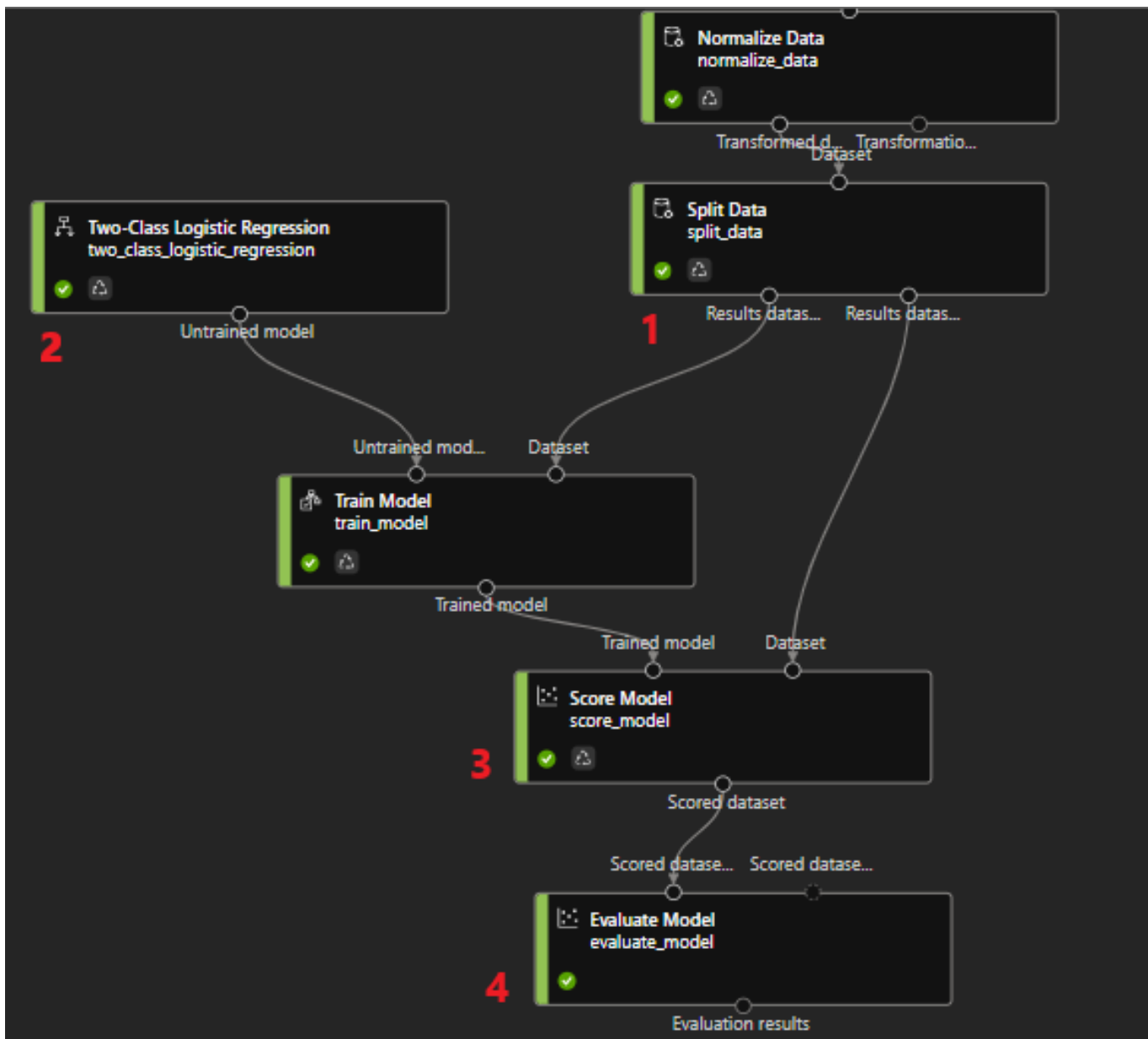
Columns to transformPregnancies,PlasmaGlucose,DiastolicBloodPressure,TricepsThickness,SerumInsulin,BMI,DiabetesPedigree,Age

Here a preview of the data used to train the model

Transformed_dataset ×							
Pregnancies	PlasmaGlucose	DiastolicBloodPressure	TricepsThickness	SerumInsulin	BMI	DiabetesPedigree	Age
							
0	0.858108	0.602151	0.317647	0.011509	0.66895	0.510511	0
0.571429	0.324324	0.741935	0.470588	0.028133	0.080345	0.036123	0.035714
0.5	0.47973	0.247312	0.529412	0.026854	0.616134	0.000438	0.035714
0.642857	0.398649	0.580645	0.211765	0.370844	0.300826	0.541848	0.392857
0.071429	0.277027	0.376344	0.235294	0.026854	0.645024	0.212047	0.017857
0	0.256757	0.731183	0.023529	0.305627	0.040264	0.011414	0.089286
0	0.601351	0.247312	0.141176	0.272379	0.098868	0.043226	0
0	0.155405	0.677419	0.423529	0.028133	0.002033	0.071112	0.089286
0.571429	0.243243	0.763441	0.305882	0.012788	0.222661	0.164558	0.571429
0.071429	0.189189	0.075269	0.388235	0.035806	0.49397	0.011648	0.089286
0.071429	0.297297	0.666667	0.047059	0.056266	0.661425	0.068467	0.017857
0.214286	0.337838	0.774194	0.282353	0.028133	0.08177	0.081391	0.035714
0.357143	0.472973	0.827957	0.423529	0.071611	0.483549	0.000516	0.303571
0.5	0.445946	0.623656	0.105882	0.038363	0.472817	0.0914	0.071429
0	0.702703	0.365591	0.047059	0.210997	0.554828	0.037231	0.428571
0.214286	0.439189	0.569892	0.458824	0.060102	0.043519	0.056802	0
0.214286	0.418919	0.430108	0.211765	0.047315	0.286616	0.229878	0.375

Pipeline

To train the model I create a pipeline on the **Designer** of Azure **Machine Learning Studio**



1. This step splits the dataset into two subsets: train and test.
2. Import the model from the catalog of Azure Machine Learning Designer
3. Compare the predicted values with the validation data.

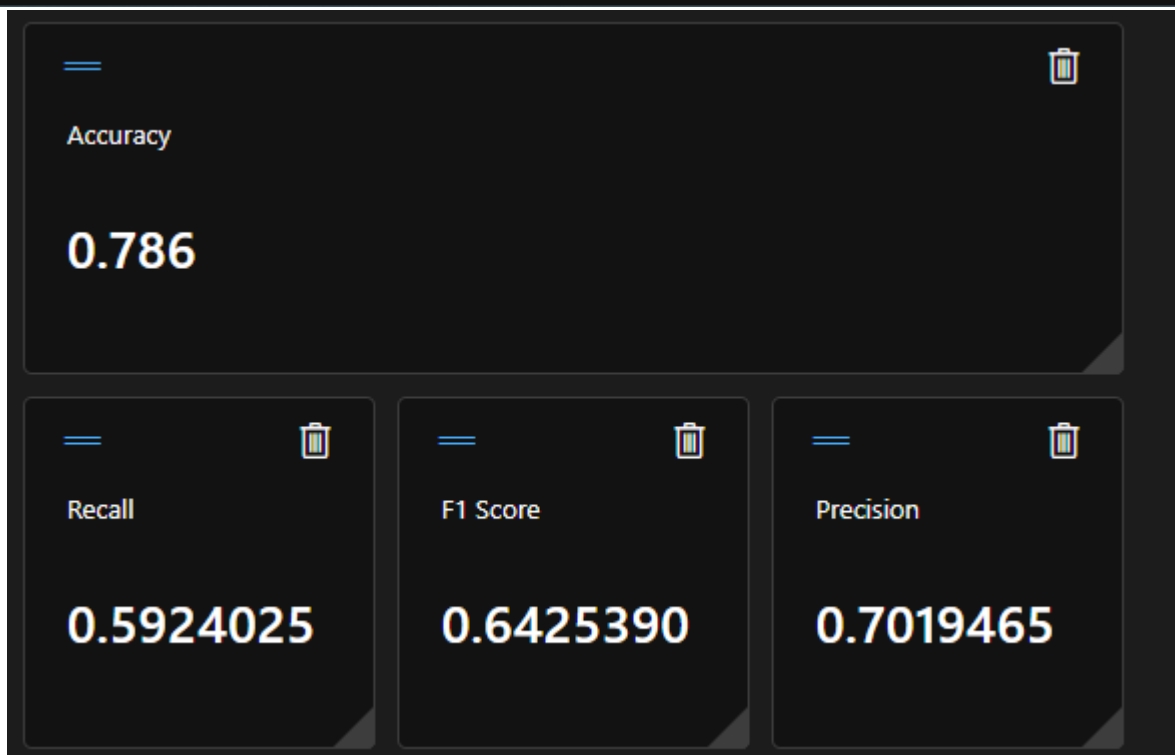
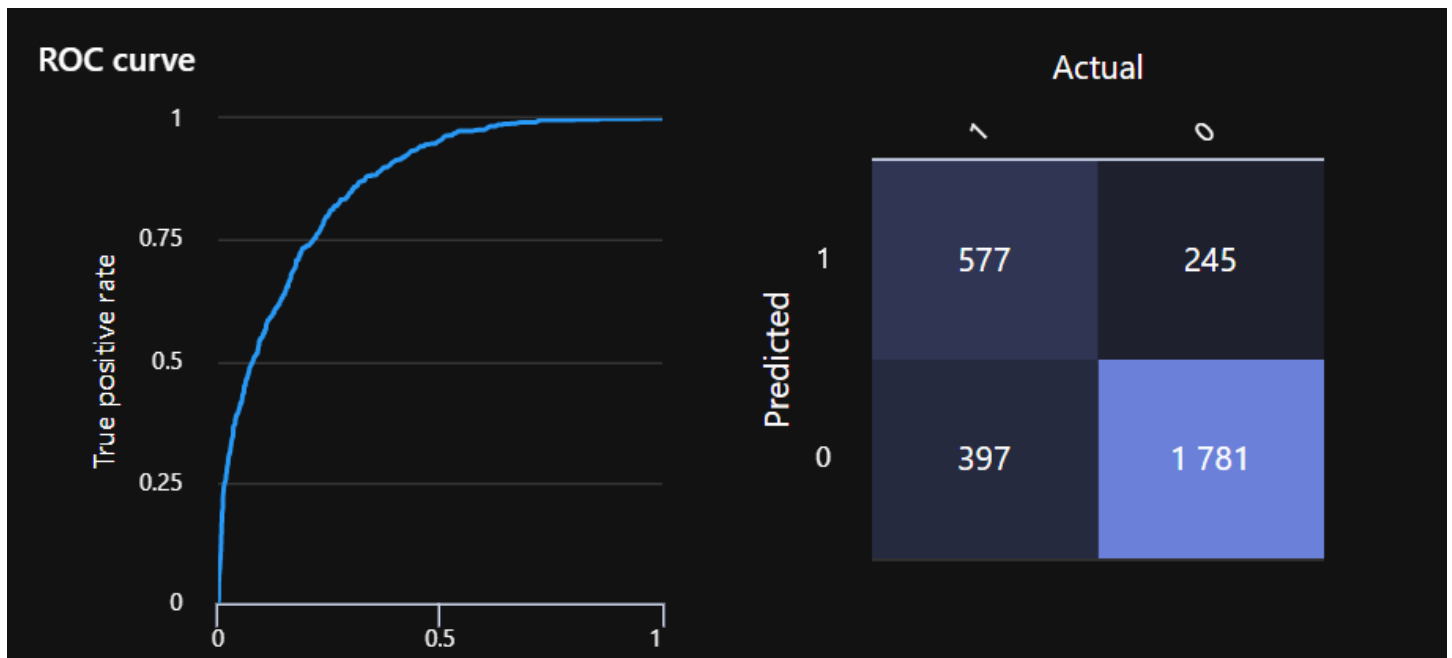
4. Extract metrics from the model's score, which serve to understand the model's performance from different angles.

Results

The output of the model's score is shown below. It consists of the original column, the category prediction and the decimal value that the model estimated from the values of each of the features analyzed. This estimate is rounded, so that an estimate of 0.48 will be classified as a negative diagnostic (0) while an estimation of 0.51 will be classified as a positive diagnostic (1).

Scored_dataset		
Rows ?	Columns ?	
3.000	12	
Diabetic	Scored Labels	Scored Probabilities
0	0	0.064733
1	1	0.914436
0	1	0.542986
0	0	0.03942
0	0	0.094209
0	0	0.193651
0	0	0.433454
1	0	0.347773
0	1	0.660992
1	0	0.416886
0	0	0.309801
0	0	0.046019
0	0	0.38558
1	1	0.548734
0	0	0.117651
0	0	0.365581
0	0	0.457425
0	0	0.187647
1	0	0.302171
0	0	0.163509

Finally, let's see below the useful metrics to evaluate the model performance.



- **Accuracy:** The ratio of correct predictions (true positives + true negatives) to the total number of predictions. In other words, what proportion of diabetes predictions did the model get right?

- **Precision:** The fraction of positive cases correctly identified (the number of true positives divided by the number of true positives plus false positives). In other words, out of all the patients that the model predicted as having diabetes, how many are actually diabetic?
- **Recall:** The fraction of the cases classified as positive that are actually positive (the number of true positives divided by the number of true positives plus false negatives). In other words, out of all the patients who actually have diabetes, how many did the model identify?
- **F1 Score:** An overall metric that essentially combines precision and recall.

Conclusions:

Considering the values of these metrics, it's possible to say that the applied model has substantial accuracy, precision and F1 score values and a moderate recall value. This means that a considerable number of diabetic predictions were right, that a considerable number of patients identified with diabetes by the model had the disease and that a moderate number of patients with diabetes were identified like positives by the model. Since a considerable precision was obtained but a moderate recall value, we obtained a considerable but not so high F1 score. Considering these conclusions and looking at the ROC curve shown before, it's possible to affirm that the model has a certain degree of predictability, without being very high as such.