ASSIGNMENT 5

TEAM MATES:
Rahavi Selvarajan – 1007346445
rahavi.selvarajan@mail.utoronto.ca

Hruday Vishal Kanna Anand - 1006874517
vishal.kanna@mail.utoronto.ca

PART - A

1. **Explain below 5 components shown in orange boxes. Explain which Azure components you will use where in this big data architecture and why.**

Raw Data → Azure Data Lake
Ingest Data → Azure Data Factory
Data Store → Azure Data Lake
Prepare and transform data → Azure Databricks and Synapse Analytics
Model and serve data → Azure Cosmos DB and Synapse Analytics

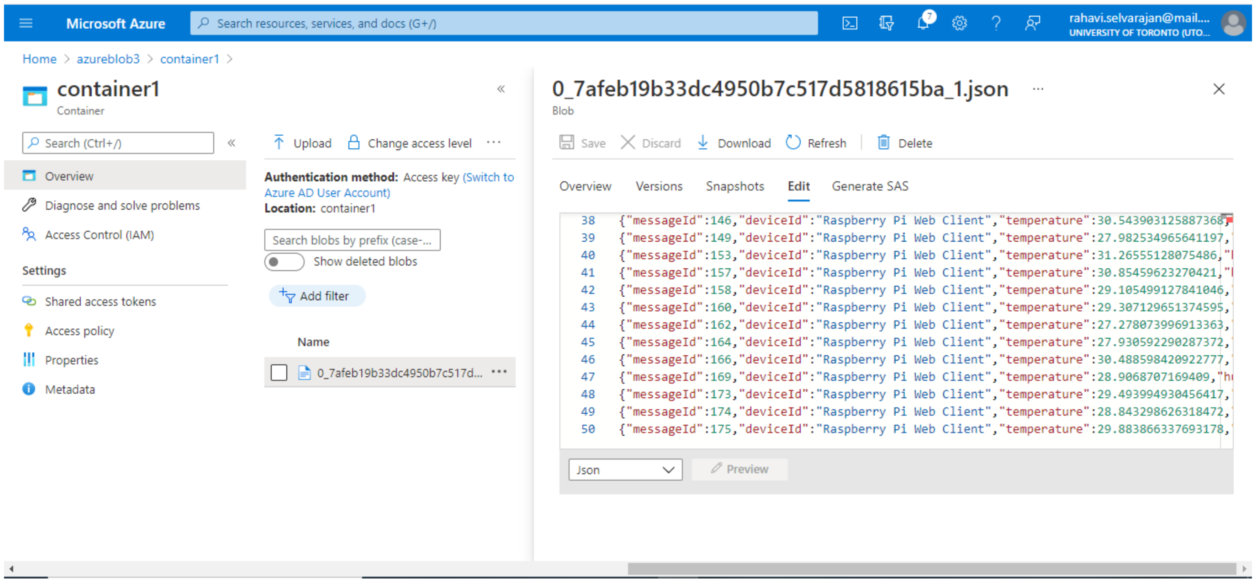
Azure Data Lake: It is a storage that contains all the raw data collected from the environment or a machine.
Azure Data Factory: Azure Data factory is a data integration offering by Azure for ETL services.
Azure Databricks: It is a data analytics platform which takes in data from the IoT hub, or blob storage and performs operations on them.
Azure Cosmos DB: It is a fully managed No-SQL database service and is helpful in modern app development.
Azure Synapse Analytics: It is an analytics service with the abilities of machine learning algorithms and supports data warehousing.

2. **Explain how Stream Analytics works in Azure.**

- An Azure Stream Analytics job consists of an input, query, and an output.
- It ingests data from Azure IoT Hub, or the Blob Storage.
- The query is based on SQL query language and can be used to easily filter, sort, aggregate, and join streaming data.

Each job has one or several outputs for the transformed data, and can be controlled in response to the information analysed.

3. **Deploy all the resources in Azure Portal. Implement a Stream Analytics job by using the Azure portal.**



Microsoft Azure
Search resources, services, and docs (G+/)
rahavi.selvarajan@mail....
UNIVERSITY OF TORONTO (UTO...
Home > azureblob3 > container1 >
container1
Container
Search (Ctrl+/)
Overview
Diagnose and solve problems
Access Control (IAM)
Settings
Shared access tokens
Access policy
Properties
Metadata
Upload
Change access level
Authentication method: Access key (Switch to Azure AD User Account)
Location: container1
Search blobs by prefix (case-...
Show deleted blobs
Add filter
Name
0_7afeb19b33dc4950b7c517d...
0_7afeb19b33dc4950b7c517d5818615ba_1.json
Blob
Save
Discard
Download
Refresh
Delete
Overview
Versions
Snapshots
Edit
Generate SAS
Json
Preview

```
{"messageId":146,"deviceId":"Raspberry Pi Web Client","temperature":30.543903125887368,
{"messageId":149,"deviceId":"Raspberry Pi Web Client","temperature":27.982534965641197,
{"messageId":153,"deviceId":"Raspberry Pi Web Client","temperature":31.2655128075486,
{"messageId":157,"deviceId":"Raspberry Pi Web Client","temperature":30.85459623270421,
{"messageId":158,"deviceId":"Raspberry Pi Web Client","temperature":29.105499127841046,
{"messageId":160,"deviceId":"Raspberry Pi Web Client","temperature":29.307129651374595,
{"messageId":162,"deviceId":"Raspberry Pi Web Client","temperature":27.278073996913363,
{"messageId":164,"deviceId":"Raspberry Pi Web Client","temperature":27.930592290287372,
{"messageId":166,"deviceId":"Raspberry Pi Web Client","temperature":30.488598420922777,
{"messageId":169,"deviceId":"Raspberry Pi Web Client","temperature":28.9068707169409,
{"messageId":173,"deviceId":"Raspberry Pi Web Client","temperature":29.493994930456417,
{"messageId":174,"deviceId":"Raspberry Pi Web Client","temperature":28.843298626318472,
{"messageId":175,"deviceId":"Raspberry Pi Web Client","temperature":29.883866337693178,
```

PART - B

1. **Explain what problem you are going to solve using this dataset. Provide a brief overview of your problem statement.**

The dataset chosen is Gait Classification Dataset. The dataset was created by calculating the walking parameters of 16 different volunteers aged between 20 and 34 years old. A total of 321 attributes are present and out of which only 26 attributes have proper labels.

Problem Statement: One of the gait attributes is gait variability. Gait variability is the fluctuation of gait measures between steps. Higher variability indicates the risk of falling or frailty or developing a neuro-degenerative disease and is more common in aged people. Lower variability indicates that a person is physically fit. Our target here is gait variability. We are trying to train a model to predict whether a person is prone to the risk of having a neuro degenerative disease or is physically fit with the help of the other gait attributes.

Cleaning: The dataset has a large number of attributes with no proper labels and these attributes are removed.

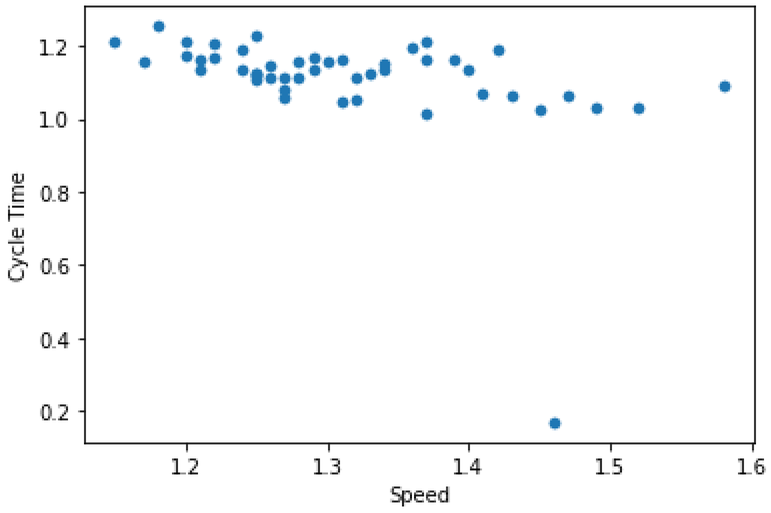
Preprocessing: We have to map the variability feature to 1's and 0's as it will be our target feature for our models.

Modelling: For the modelling, we chose two algorithms - Logistic Regression and Naive Bayes.

2. **Explain your dataset. Explore your dataset and provide at least 5 meaningful charts/graphs with explanation.**

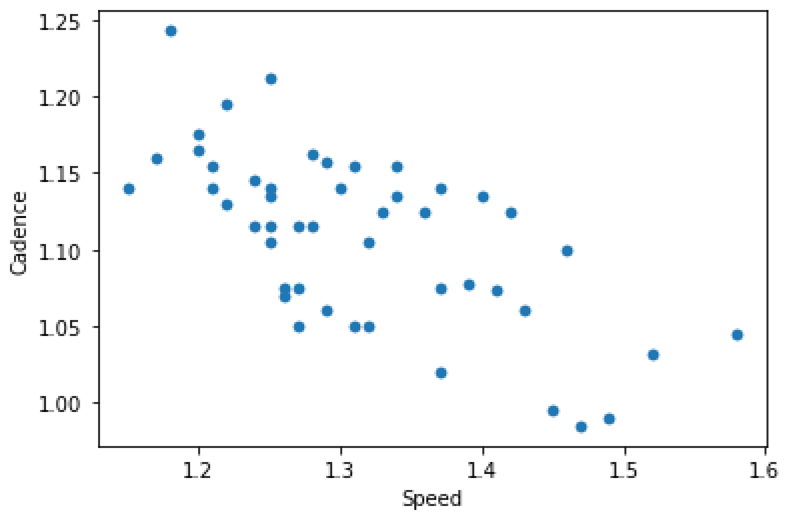
1. Cycle Time Vs Speed

Gait cycle time is the time interval between two successive occurrences of one of the repetitive events of the locomotion. Gait Speed is the time one takes to walk a specified distance. From the graph below, we could see that the cycle time remains almost equal for all the volunteers irrespective of the time taken by them to cover a specified distance.



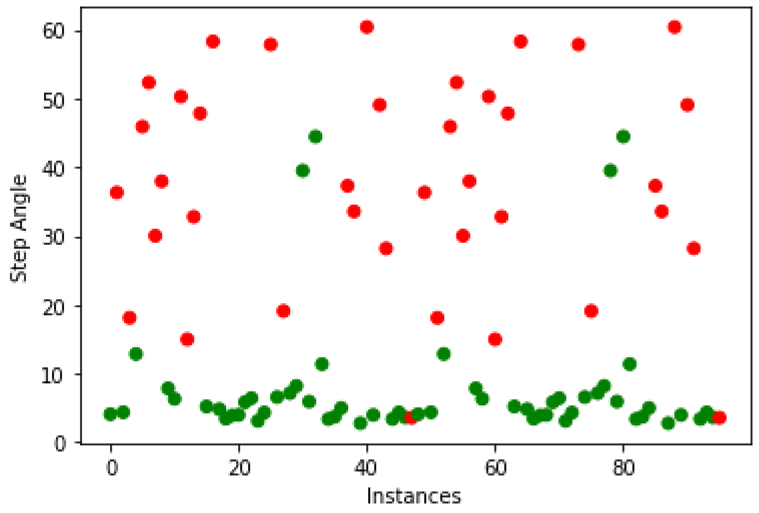
2. Speed Vs Cadence

Gait Cadence is the number of steps or cycles completed by a person in a specified period of time i.e., steps or cycles per minute. From the graph, it is evident that as the speed increases the number of steps taken to complete a specified distance decrease. The graph shows a linearly decreasing behavior.



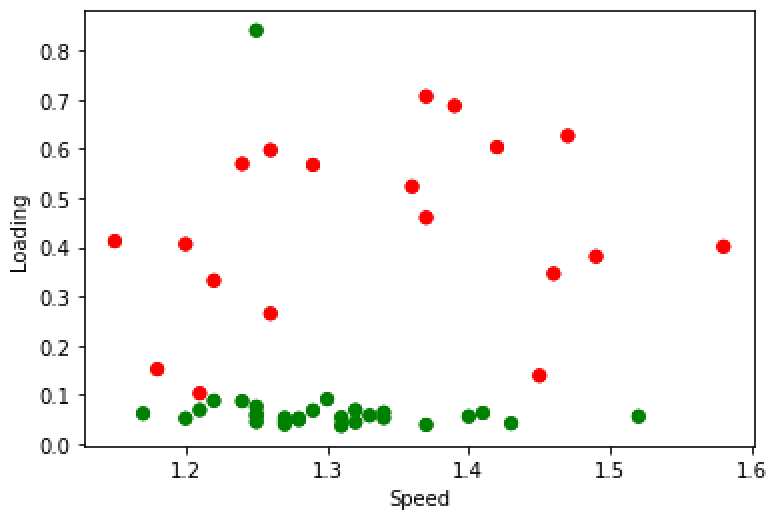
3. **Step Angle Grouped by variability**

The step angle captures the angular change (ie, rotation) in the sagittal plane of lower limbs during walking. From the graph below, it is evident that people without risks of neurodegenerative diseases have larger step angles (indicated by red) and those with the risks show smaller step angles (indicated by green).



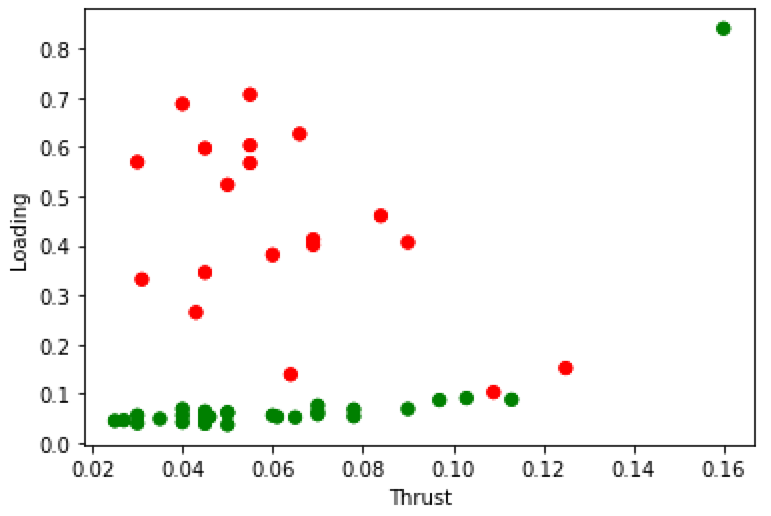
4. **Speed Vs Loading - Grouped by Variability**

Gait loading is the phase where the body absorbs the impact of the foot and the trunk of the body moves forward to align with the line of gravity. The graph shows that the loading is almost zero in people with the risks of developing the neurodegenerative diseases.



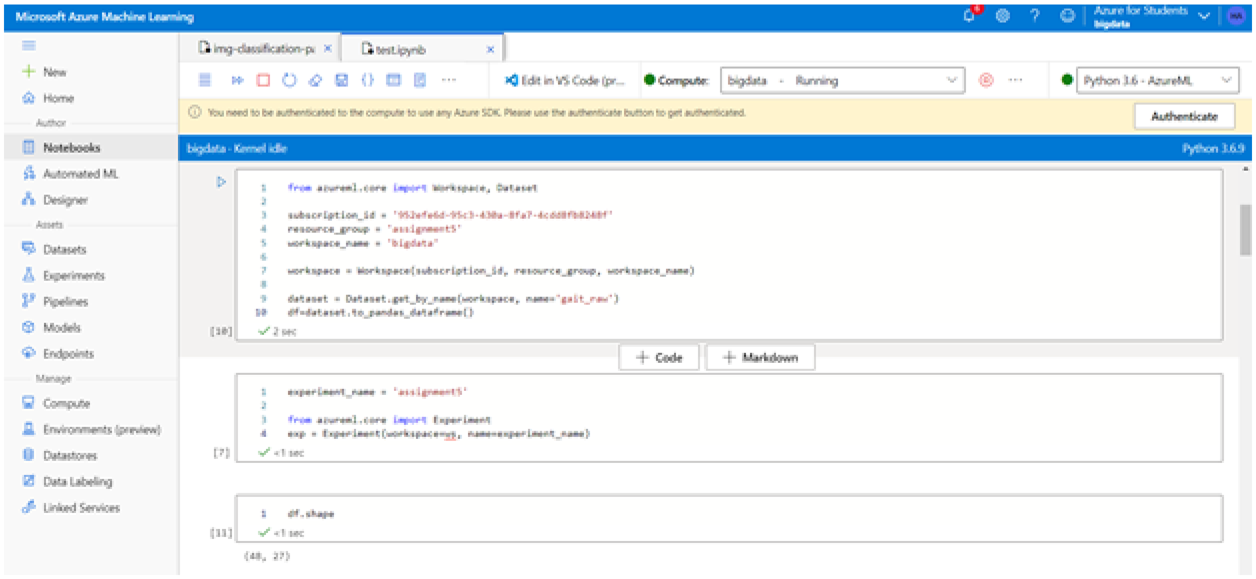
5. **Thrust Vs Loading - Grouped by Variability**

Like the previous graph, irrespective of the thrust, loading remains zero.



3. **Do data cleaning/pre-processing as required and explain what you have done for your dataset and why?**

Loading dataset into notebook and exploring



You need to be authenticated to the compute to use any Azure SDK. Please use the authenticate button to get authenticated.

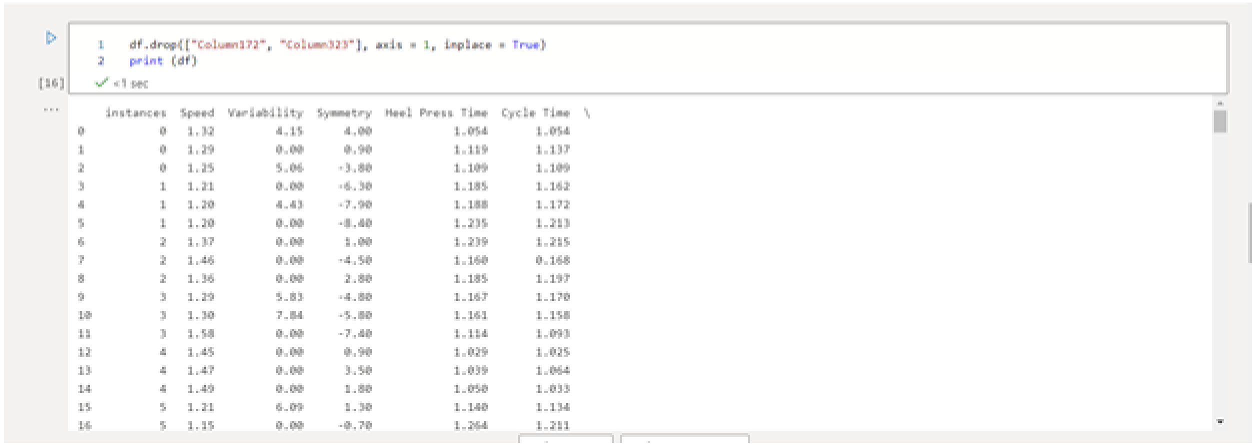
```
from azureml.core import Workspace, Dataset

subscription_id = '952efe6d-95c3-430a-8fa7-4cdd8fb8248f'
resource_group = 'assignment5'
workspace_name = 'bigdata'

workspace = Workspace(subscription_id, resource_group, workspace_name)

dataset = Dataset.get_by_name(workspace, name='gait_raw')
df=dataset.to_pandas_dataframe()
```
[10] 2 sec

```
experiment_name = 'assignment5'

from azureml.core import Experiment
exp = Experiment(workspace=ws, name=experiment_name)
```
[7] <1 sec

```
df.shape
```
[11] <1 sec

```
(48, 27)
```

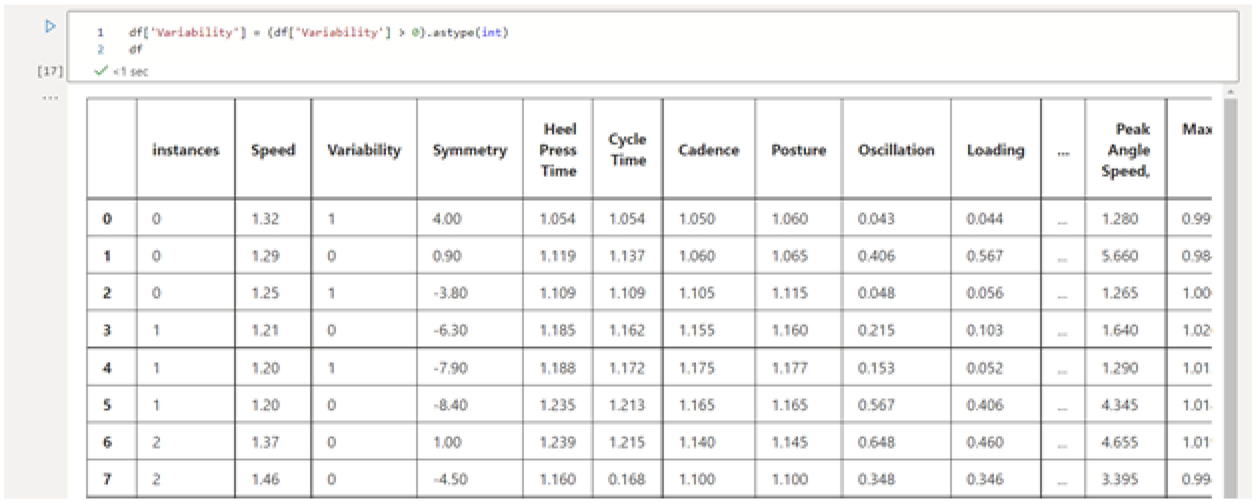
Removing unwanted columns- while we uploaded the file as dataset to azure ml service we removed most of the unlabeled columns but some still persisted. These columns were removed with the code below.



```
df.drop(["Column172", "Column323"], axis = 1, inplace = True)
print (df)
```
[16] <1 sec

```
    instances  Speed  Variability  Symmetry  Heel Press Time  Cycle Time  \
0           0   1.32         4.15      4.00            1.054       1.054
1           0   1.29         0.00      0.90            1.119       1.137
2           0   1.25         5.06     -3.80            1.109       1.109
3           1   1.21         0.00     -6.30            1.185       1.162
4           1   1.20         4.43     -7.90            1.188       1.172
5           1   1.20         0.00     -8.40            1.235       1.213
6           2   1.37         0.00      1.00            1.239       1.215
7           2   1.46         0.00     -4.50            1.160       0.168
8           2   1.36         0.00      2.80            1.185       1.197
9           3   1.29         5.83     -4.80            1.167       1.170
10          3   1.30         7.84     -5.80            1.161       1.158
11          3   1.58         0.00     -7.40            1.114       1.093
12          4   1.45         0.00      0.90            1.029       1.025
13          4   1.47         0.00      3.50            1.039       1.064
14          4   1.49         0.00      1.80            1.050       1.033
15          5   1.21         6.09      1.30            1.140       1.134
16          5   1.15         0.00     -0.70            1.264       1.211
```

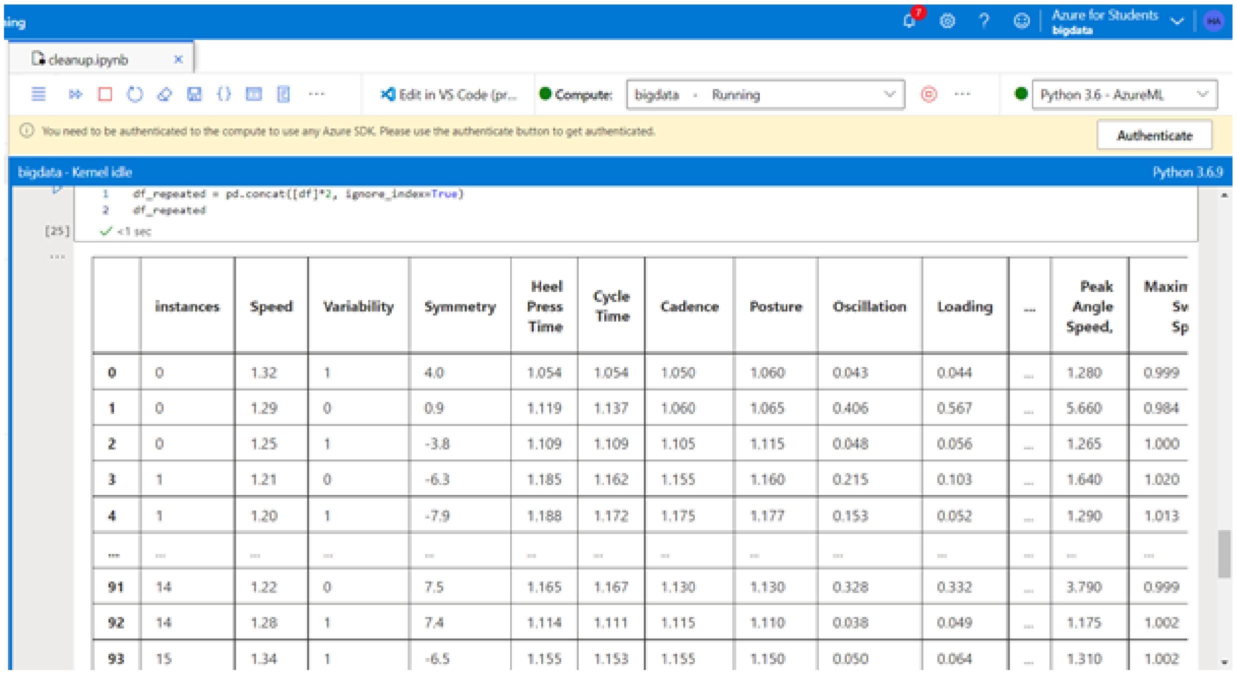
Updating target column(Variability)- to 1 and 0 so that we can proceed to make a valid classification model. All the classification models the target feature needs to be distinct classes for the model to function properly.



```
df['Variability'] = (df['Variability'] > 0).astype(int)
df
```
[17] ✓ <1 sec

	instances	Speed	Variability	Symmetry	Heel Press Time	Cycle Time	Cadence	Posture	Oscillation	Loading	...	Peak Angle Speed,	Max
0	0	1.32	1	4.00	1.054	1.054	1.050	1.060	0.043	0.044	...	1.280	0.99
1	0	1.29	0	0.90	1.119	1.137	1.060	1.065	0.406	0.567	...	5.660	0.98
2	0	1.25	1	-3.80	1.109	1.109	1.105	1.115	0.048	0.056	...	1.265	1.00
3	1	1.21	0	-6.30	1.185	1.162	1.155	1.160	0.215	0.103	...	1.640	1.02
4	1	1.20	1	-7.90	1.188	1.172	1.175	1.177	0.153	0.052	...	1.290	1.01
5	1	1.20	0	-8.40	1.235	1.213	1.165	1.165	0.567	0.406	...	4.345	1.01
6	2	1.37	0	1.00	1.239	1.215	1.140	1.145	0.648	0.460	...	4.655	1.01
7	2	1.46	0	-4.50	1.160	0.168	1.100	1.100	0.348	0.346	...	3.395	0.99

Replicating rows- to create more data for automated ml, as the process requires a minimum of 50 rows to run.

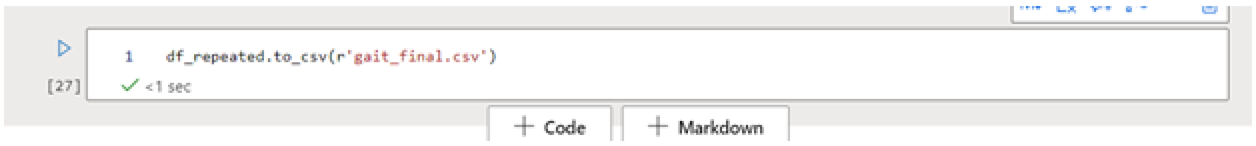


You need to be authenticated to the compute to use any Azure SDK. Please use the authenticate button to get authenticated.

```
df_repeated = pd.concat([df]*2, ignore_index=True)
df_repeated
```
[25] ✓ <1 sec

	instances	Speed	Variability	Symmetry	Heel Press Time	Cycle Time	Cadence	Posture	Oscillation	Loading	...	Peak Angle Speed,	Maxin Sv Sp
0	0	1.32	1	4.0	1.054	1.054	1.050	1.060	0.043	0.044	...	1.280	0.999
1	0	1.29	0	0.9	1.119	1.137	1.060	1.065	0.406	0.567	...	5.660	0.984
2	0	1.25	1	-3.8	1.109	1.109	1.105	1.115	0.048	0.056	...	1.265	1.000
3	1	1.21	0	-6.3	1.185	1.162	1.155	1.160	0.215	0.103	...	1.640	1.020
4	1	1.20	1	-7.9	1.188	1.172	1.175	1.177	0.153	0.052	...	1.290	1.013
...	...	...	...	...	...	...	...	...	...	...	...	...	...
91	14	1.22	0	7.5	1.165	1.167	1.130	1.130	0.328	0.332	...	3.790	0.999
92	14	1.28	1	7.4	1.114	1.111	1.115	1.110	0.038	0.049	...	1.175	1.002
93	15	1.34	1	-6.5	1.155	1.153	1.155	1.150	0.050	0.064	...	1.310	1.002

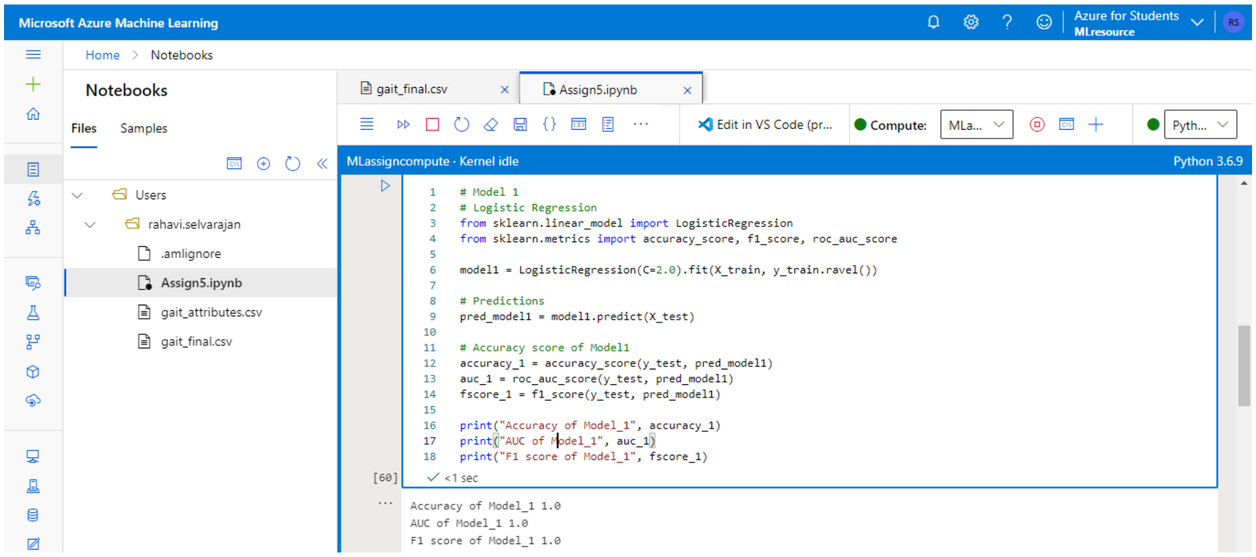
Saving cleaned and processed file-



```
df_repeated.to_csv(r'gait_final.csv')
```
[27] ✓ <1 sec

4. **Implement 2 machine learning models, explain which algorithms you have selected and why. Compare them and show success metrics (Accuracy/RMSE/Confusion Matrix) as per your problem. Explain results.**

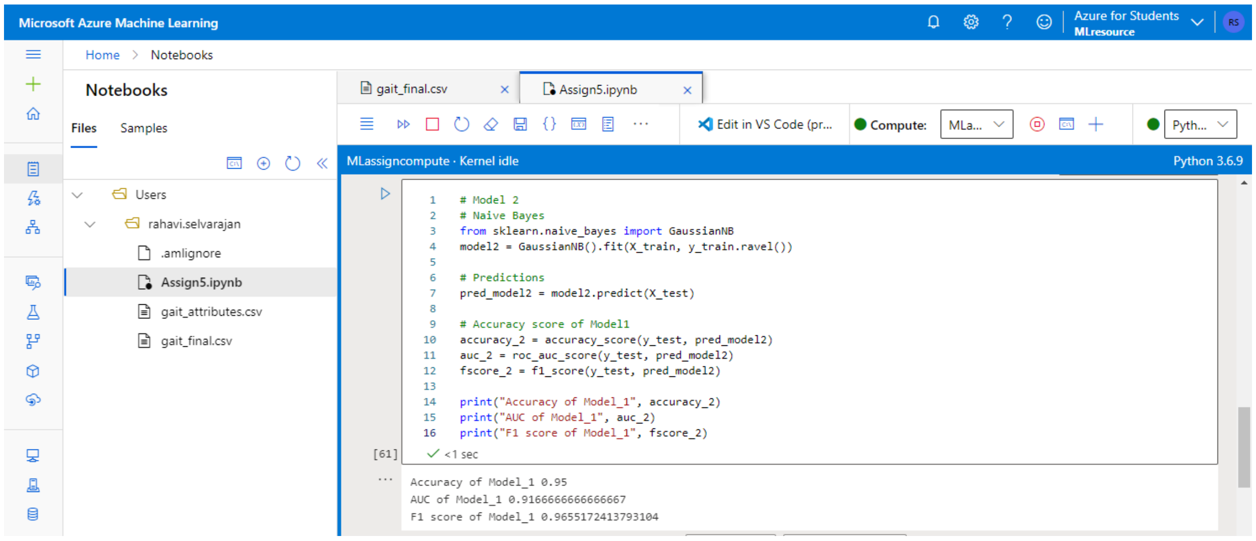
Model 1:
Algorithm: Logistic Regression
Regularization Parameter: 2.0
Reason for choosing Logistic Regression: Logistic Regression is the most common algorithm used for binary classification involving numerical values and is known to prevent overfitting.



```
# Model 1
# Logistic Regression
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, f1_score, roc_auc_score

model1 = LogisticRegression(C=2.0).fit(X_train, y_train.ravel())

# Predictions
pred_model1 = model1.predict(X_test)

# Accuracy score of Model1
accuracy_1 = accuracy_score(y_test, pred_model1)
auc_1 = roc_auc_score(y_test, pred_model1)
fscore_1 = f1_score(y_test, pred_model1)

print("Accuracy of Model_1", accuracy_1)
print("AUC of Model_1", auc_1)
print("F1 score of Model_1", fscore_1)
```

```
Accuracy of Model_1 1.0
AUC of Model_1 1.0
F1 score of Model_1 1.0
```

Model 2:
Algorithm: Naive Bayes
Reason for choosing Naive Bayes: It is easy to implement and performs well in test dataset.



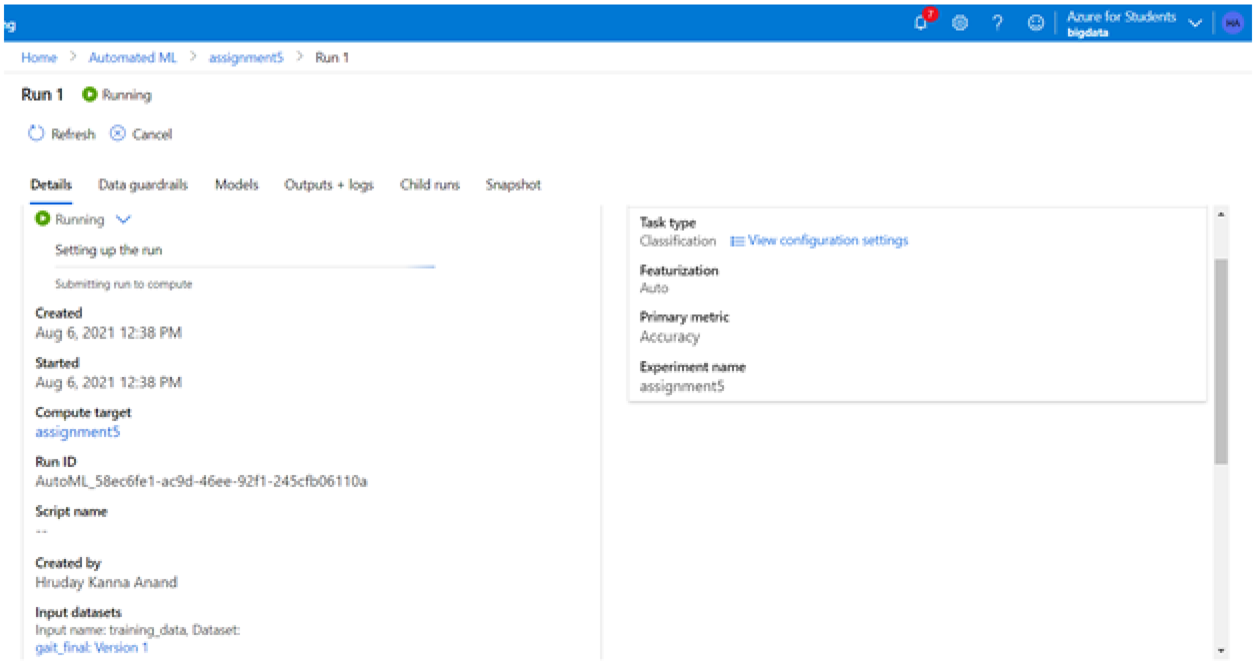
```
# Model 2
# Naive Bayes
from sklearn.naive_bayes import GaussianNB
model2 = GaussianNB().fit(X_train, y_train.ravel())

# Predictions
pred_model2 = model2.predict(X_test)

# Accuracy score of Model1
accuracy_2 = accuracy_score(y_test, pred_model2)
auc_2 = roc_auc_score(y_test, pred_model2)
fscore_2 = f1_score(y_test, pred_model2)

print("Accuracy of Model_1", accuracy_2)
print("AUC of Model_1", auc_2)
print("F1 score of Model_1", fscore_2)
```

```
Accuracy of Model_1 0.95
AUC of Model_1 0.9166666666666667
F1 score of Model_1 0.9655172413793104
```

From the above two models, Logistic Regression performs better compared to the Naive Bayes. The Accuracy, F1 score and AUC scores of the first model are better compared to the model 2 i.e, Naive Bayes. F1 score shows the accuracy of the test set and logistic regression shows better performance on the test set.

5. **Use Automated ML for your data set. Explain best model results.**

Automated ml run initiated-


ng Azure for Students bigdata HA

Home > Automated ML > assignment5 > Run 1

Run 1 Running

Refresh Cancel

Details Data guardrails Models Outputs + logs Child runs Snapshot

Running

Setting up the run

Submitting run to compute

Created
Aug 6, 2021 12:38 PM

Started
Aug 6, 2021 12:38 PM

Compute target
assignment5

Run ID
AutoML_58ec6fe1-ac9d-46ee-92f1-245cfb06110a

Script name
--

Created by
Hruday Kanna Anand

Input datasets
Input name: training_data, Dataset:
gait_final: Version 1

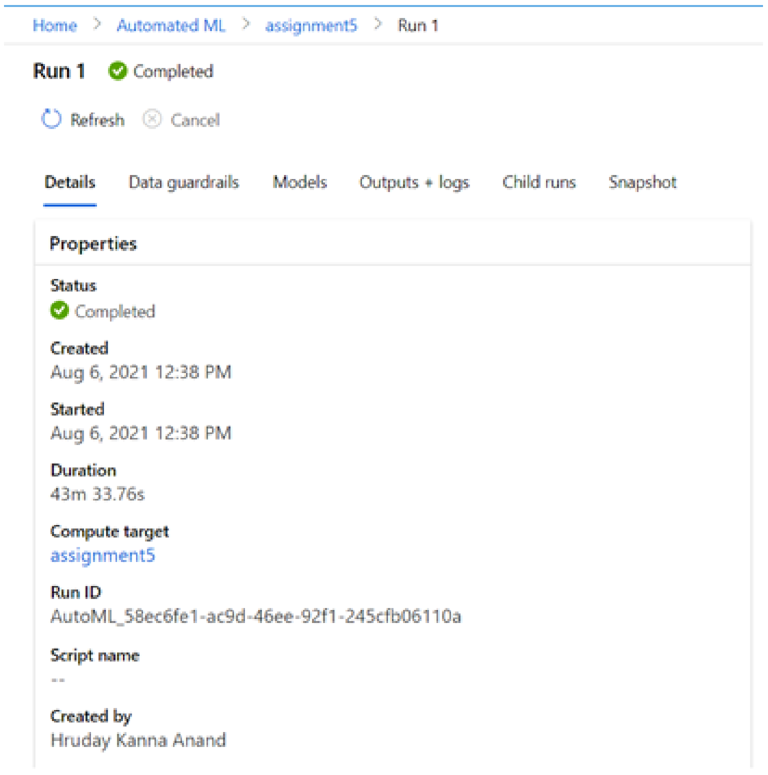
Task type
Classification View configuration settings

Featurization
Auto

Primary metric
Accuracy

Experiment name
assignment5

Automated run completed


Home > Automated ML > assignment5 > Run 1

Run 1 Completed

Refresh Cancel

Details | Data guardrails | Models | Outputs + logs | Child runs | Snapshot

Properties

Status
Completed

Created
Aug 6, 2021 12:38 PM

Started
Aug 6, 2021 12:38 PM

Duration
43m 33.76s

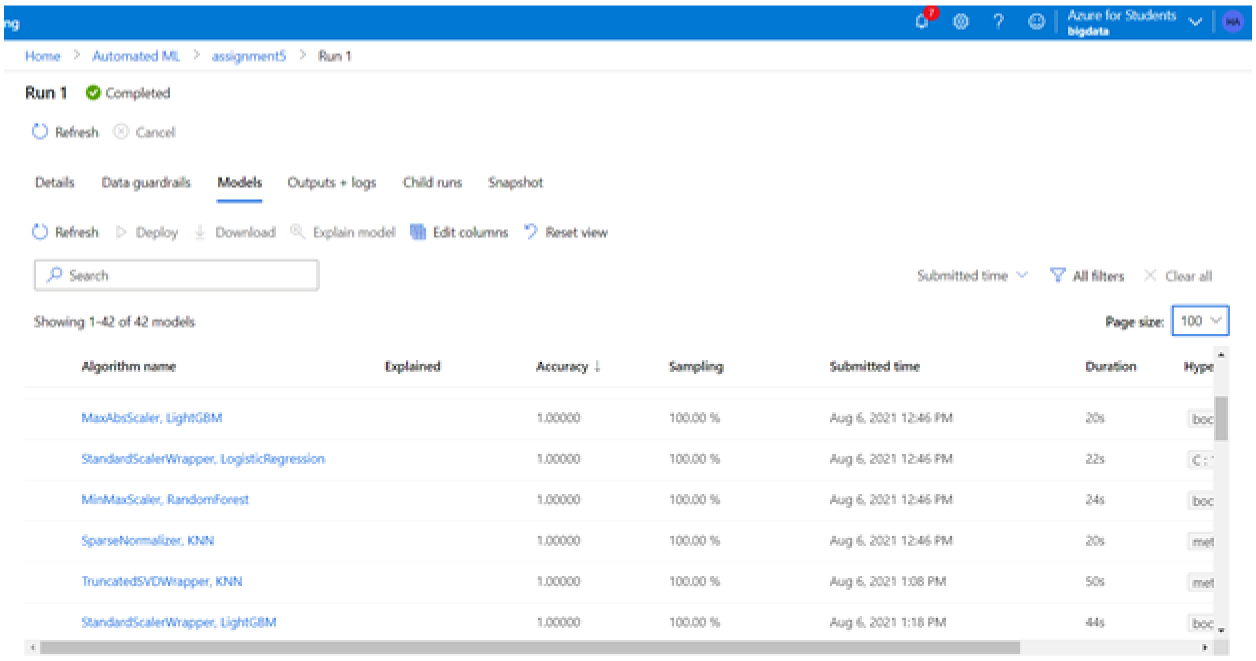
Compute target
assignment5

Run ID
AutoML_58ec6fe1-ac9d-46ee-92f1-245cfb06110a

Script name
--

Created by
Hruday Kanna Anand

Models-



Home > Automated ML > assignment5 > Run 1

Run 1 Completed

Refresh Cancel

Details | Data guardrails | Models | Outputs + logs | Child runs | Snapshot

Refresh | Deploy | Download | Explain model | Edit columns | Reset view

Search | Submitted time | All filters | Clear all

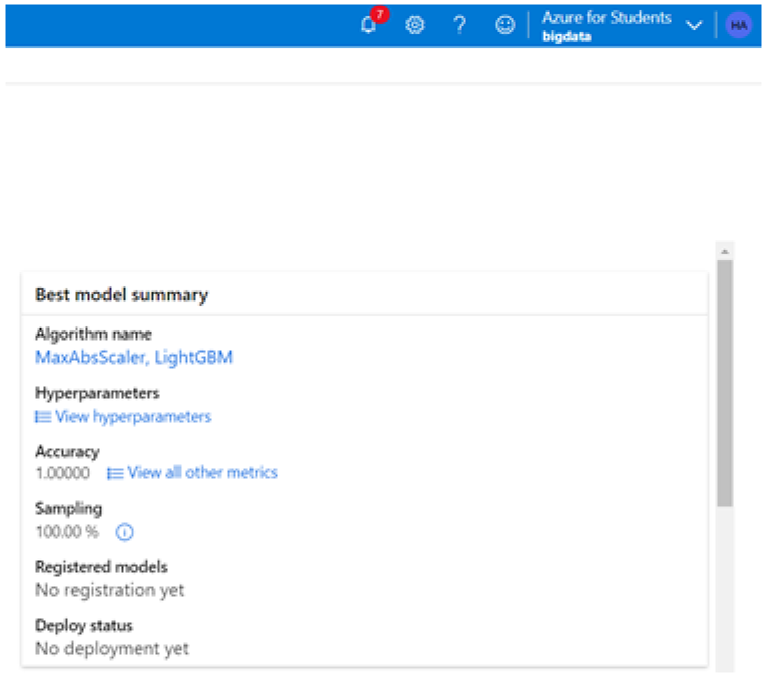
Showing 1-42 of 42 models | Page size: 100

Algorithm name	Explained	Accuracy ↓	Sampling	Submitted time	Duration	Hype
MaxAbsScaler, LightGBM		1.00000	100.00 %	Aug 6, 2021 12:46 PM	20s	boc
StandardScalerWrapper, LogisticRegression		1.00000	100.00 %	Aug 6, 2021 12:46 PM	22s	C :
MinMaxScaler, RandomForest		1.00000	100.00 %	Aug 6, 2021 12:46 PM	24s	boc
SparseNormalizer, KNN		1.00000	100.00 %	Aug 6, 2021 12:46 PM	20s	met
TruncatedSVDWrapper, KNN		1.00000	100.00 %	Aug 6, 2021 1:08 PM	50s	met
StandardScalerWrapper, LightGBM		1.00000	100.00 %	Aug 6, 2021 1:18 PM	44s	boc

We can see that 42 models have been made and trained and almost all of them have an accuracy of 1.

Best model-

The best model with Light GBM with max abs scaler. Light GBM is a gradient boosting framework that uses tree based learning algorithms. Max abs scaler is used to preprocess our features dividing our features by the maximum value to scale down the features to a common scale.



Best model summary

Algorithm name
MaxAbsScaler, LightGBM

Hyperparameters
View hyperparameters

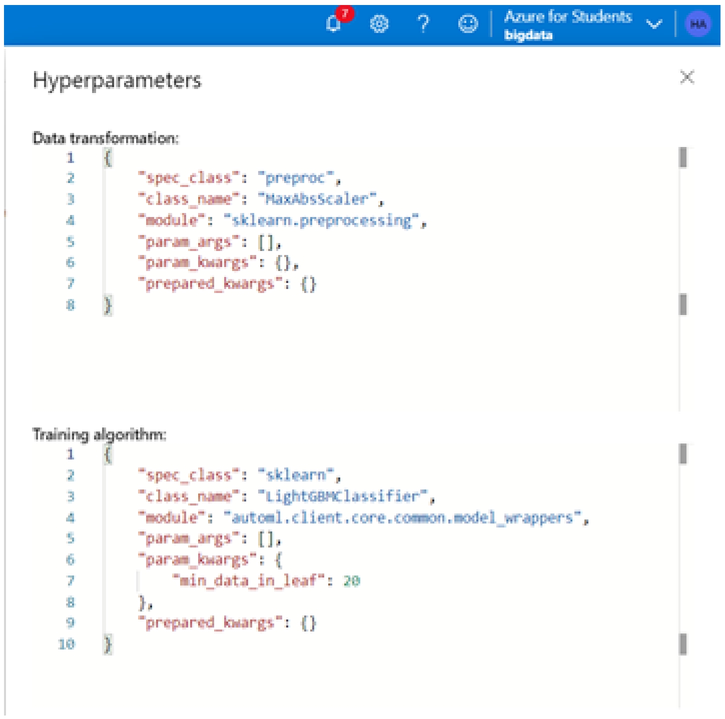
Accuracy
1.00000 View all other metrics

Sampling
100.00 %

Registered models
No registration yet

Deploy status
No deployment yet

Hyperparameters of the best model-



Hyperparameters

Data transformation:

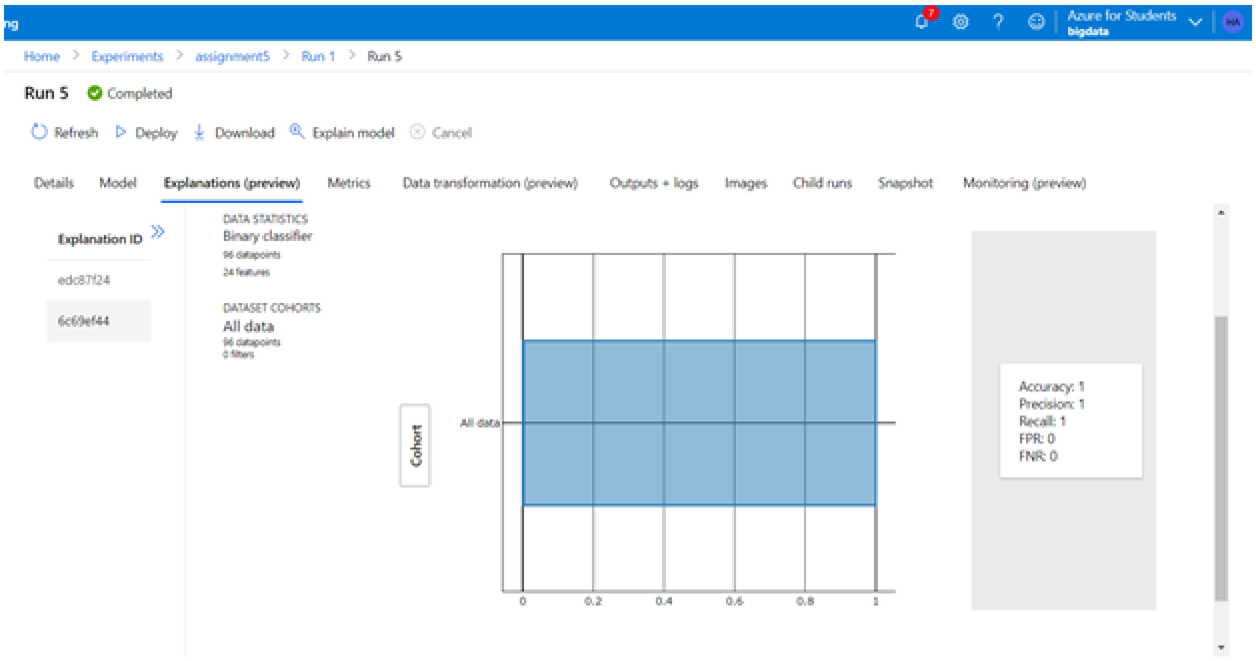
```
{
    "spec_class": "preproc",
    "class_name": "MaxAbsScaler",
    "module": "sklearn.preprocessing",
    "param_args": [],
    "param_kwargs": {},
    "prepared_kwargs": {}
}
```

Training algorithm:

```
{
    "spec_class": "sklearn",
    "class_name": "LightGBMClassifier",
    "module": "automl.client.core.common.model_wrappers",
    "param_args": [],
    "param_kwargs": {
        "min_data_in_leaf": 20
    },
    "prepared_kwargs": {}
}
```

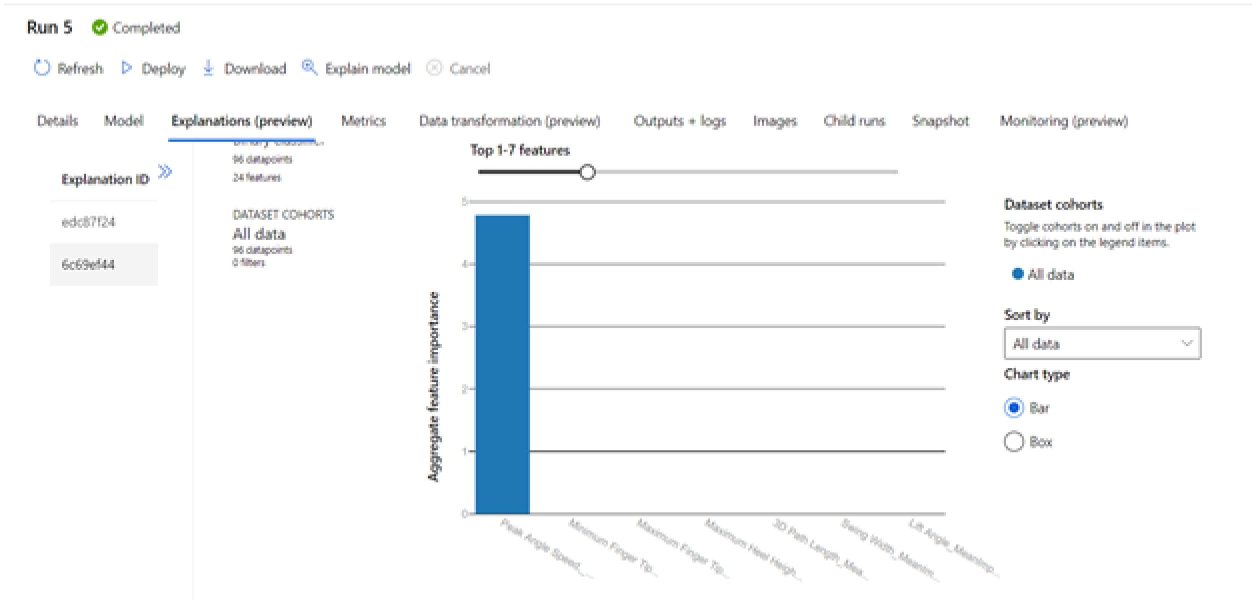
Model explanation-

We can see the model has an accuracy, precision and recall 1.



Accuracy: 1
Precision: 1
Recall: 1
FPR: 0
FNR: 0

From this graph we can see our model relies largely on the peak angle speed feature to come up with an accurate output.


Run 5 Completed
Refresh
Deploy
Download
Explain model
Cancel
Details
Model
Explanations (preview)
Metrics
Data transformation (preview)
Outputs + logs
Images
Child runs
Snapshot
Monitoring (preview)
Explanation ID
edc87f24
6c69ef44
96 datapoints
24 features
DATASET COHORTS
All data
96 datapoints
0 filters
Top 1-7 features
Aggregate feature importance
0
1
2
3
4
5
Peak Angle Speed_...
Minimum Finger Tip...
Maximum Finger Tip...
Maximum Heel Heigh...
3D Path Length_Mea...
Swing Width_Meanin...
Lift Angle_Meaning...
Dataset cohorts
Toggle cohorts on and off in the plot by clicking on the legend items.
All data
Sort by
All data
Chart type
Bar
Box