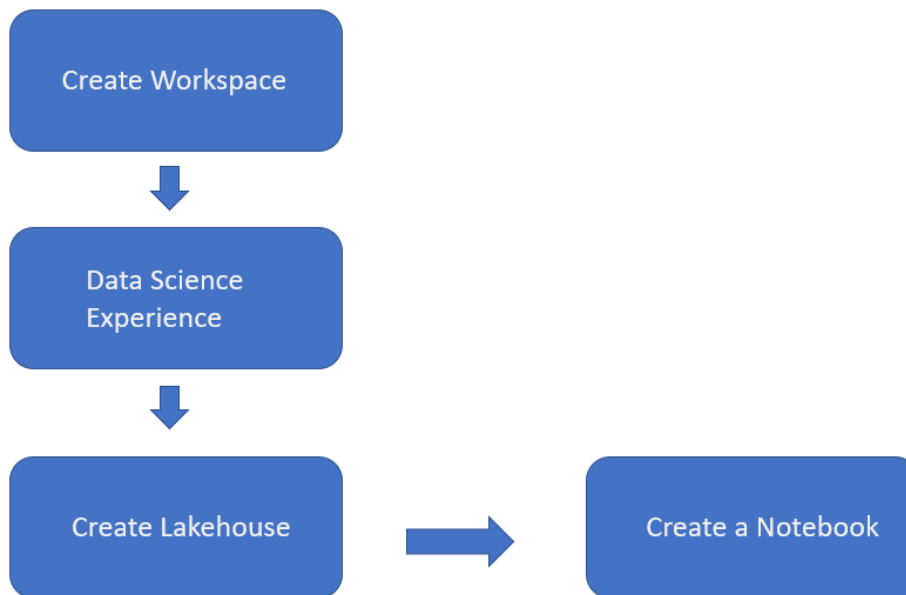


How is it easy to train machine learning models in Fabric?



Setting Up the Environment

A lakehouse was set up in Microsoft Fabric to store and organize the data in one place. It made accessing and managing datasets much easier and kept the workflow smooth.

The lakehouse is created under the Data science experience.

Utilizing Python in Notebooks

We can use notebooks to train machine learning models.

Python notebooks within Fabric were used to implement the logistic regression model. The flexibility of Python and its libraries, such as pandas, matplotlib, and scikit-learn, made data preprocessing, model training, and evaluation efficient and effective.

The workflow involved key steps, including

- Importing the Data from the lakehouse
- Data cleaning and preparation.
- Splitting data into training and testing sets.
- Training and evaluating the logistic regression model with relevant metrics.

The screenshot shows a Databricks Notebook interface. The top bar includes a search bar, a trial status (23 days left), and various utility icons. The left sidebar contains navigation options like Home, Create, Browse, OneLake, Monitor, Workspaces, and Data Science. The main area displays a Python code cell [37] that connects to a DeltaTable, reads data from a storage path, and displays the first 1000 rows. Below the code, a table view shows the results with columns for various geometric features. The bottom status bar indicates a session timeout and autosave status.

```

1 from deltalake import DeltaTable, write_deltalake
2 table_path = 'abfss://ml@onelake.dfs.fabric.microsoft.com/ml_lakehouse/Lakehouse/Tables/dry_bean_dataset'
3 storage_options = {"bearer_token": notebookutils.credentials.getToken('storage'), "use_fabric_endpoint": "true"}
4 dt = DeltaTable(table_path, storage_options=storage_options)
5 df = dt.to_pyarrow_dataset().head(1000000).to_pandas()
6 display(df)
7
8 # Write data frame to Lakehouse
9 # write_deltalake(table_path, limited_data, mode='overwrite')
10
11 # If the table is too large and might cause an out of Memory (OOM) error,
12 # you can try using the code below. However, please note that delta_scan with default lakehouse is currently in preview.
13 # import duckdb
14 # display(duckdb.sql("select * from delta_scan('/lakehouse/default/Tables/dbo/bigdeltatable') limit 1000 ").df())

```

	12 area	12 perimeter	12 major_axis_length	12 minor_axis_length	12 aspect_ratio	12 eccentricity	12 convex_area	12 equiv_diameter	12 extent	12 solidity	12 roundness
20	137075	1432.713	526.8980615	334.1175121	1.576984272	0.773233229	139517	417.767053	0.67699716	0.982496757	0.839169
21	137115	1427.056	519.1997446	337.4746463	1.538485188	0.759942754	138970	417.8280031	0.789974016	0.986651795	0.846082
22	137358	1364.645	507.9858904	345.2228517	1.4711472378	0.733590829	138093	418.1980839	0.798073348	0.9946775	0.926881
23	137518	1417.944	519.7799999	339.9302724	1.529078291	0.756504273	139153	418.4415798	0.778239315	0.988250343	0.859510
24	137748	1389.634	499.3705441	353.3013142	1.413440947	0.706720051	138869	418.7913571	0.809463366	0.991927644	0.896384
25	137890	1410.302	522.1875739	338.3644715	1.543269515	0.76166164	139207	419.0071608	0.782239216	0.990539269	0.871201
26	138059	1459.686	540.6778226	334.1664759	1.61798942	0.786137994	143275	419.2638528	0.767369603	0.963594486	0.814246

Importing Necessary Libraries,

The screenshot shows a Databricks Notebook interface. The top bar is identical to the previous screenshot. The left sidebar is also identical. The main area displays two Python code cells. Cell [37] imports various libraries including pandas, matplotlib, seaborn, and sklearn. Cell [38] executes a command to inspect the data, showing the output of df.isna().sum(). The bottom status bar indicates a session timeout and autosave status.

```

1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4 from sklearn.model_selection import train_test_split
5 from sklearn.linear_model import LogisticRegression
6 from sklearn.metrics import classification_report
7 from sklearn.metrics import confusion_matrix

```

```

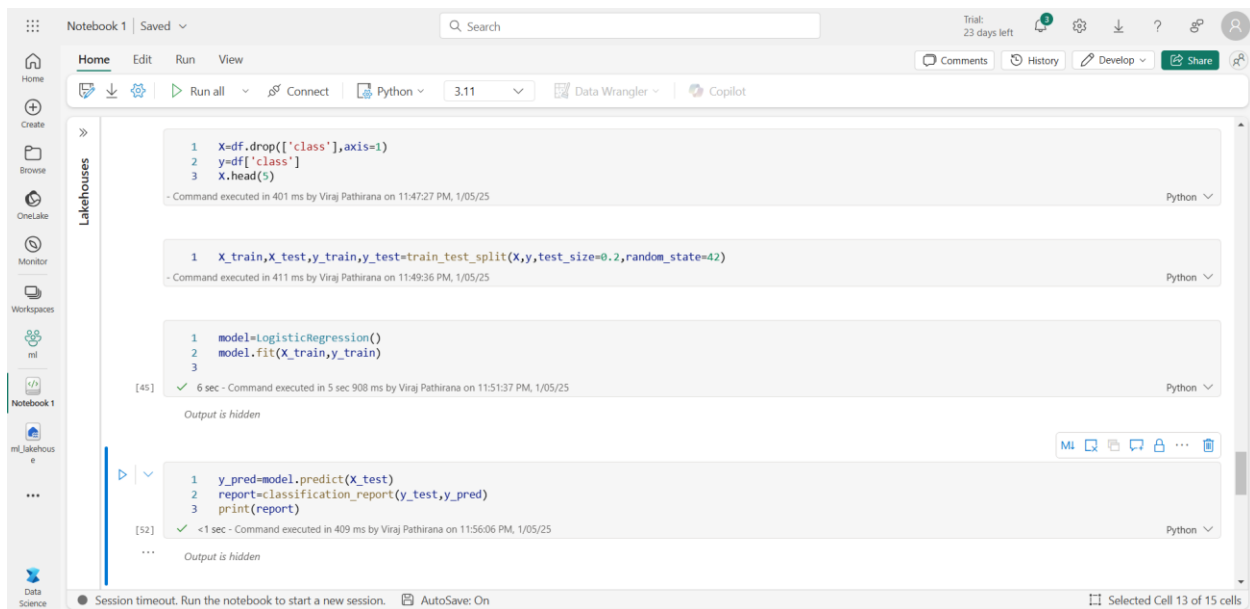
1 df.isna().sum()

```

area	0
perimeter	0
major_axis_length	0
minor_axis_length	0
aspect_ratio	0
eccentricity	0
convex_area	0
equiv_diameter	0
extent	0
solidity	0
roundness	0
compactness	0
shape_factor1	0
shape_factor2	0
shape_factor3	0

Model Training and Evaluation

The performance of the logistic regression model was thoroughly assessed using a range of evaluation metrics, including precision, recall, and F1-score, which provided a deeper understanding of the model's ability to make accurate predictions. These metrics helped identify how well the model performed in terms of minimizing false positives and false negatives, and the F1-score offered a balanced measure of precision and recall. By analyzing these insights, it became possible to pinpoint areas where the model could be improved, whether by fine-tuning hyperparameters, adjusting data preprocessing steps, or exploring alternative approaches to address any performance gaps.



```

1 y_pred=model.predict(X_test)
2 report=classification_report(y_test,y_pred)
3 print(report)

```

[52] ✓ <1 sec - Command executed in 409 ms by Viraj Pathirana on 11:56:06 PM, 1/05/25

	precision	recall	f1-score	support
BARBUNYA	0.61	0.49	0.54	265
BOMBAY	1.00	0.99	1.00	117
CALI	0.71	0.74	0.72	316
DERMASON	0.80	0.85	0.82	671
HOROZ	0.60	0.53	0.56	391
SEKER	0.69	0.62	0.65	427
SIRA	0.57	0.66	0.61	536
accuracy			0.69	2723
macro avg	0.71	0.70	0.70	2723
weighted avg	0.69	0.69	0.69	2723

Why Microsoft Fabric Stands Out?

Microsoft Fabric's unified platform simplified every aspect of the machine learning workflow. The integration of lakehouses, notebooks, and computational resources in a single environment eliminated the need for switching between tools.

The Data Science experience in Fabric offers an intuitive and efficient way to manage data science projects while leveraging Python for machine learning tasks.

Lessons and Insights

- The project underscored the importance of effective data preprocessing, rigorous evaluation of model performance, and leveraging a unified platform for data science workflows

- Using a lakehouse for data organization enhanced efficiency and streamlined the data access process

What are the Advantages of using Fabric?

To conclude, there is a variety of benefits that arise from the use of Microsoft Fabric notebooks, making them especially useful tools in data science and machine learning activities. Even though VS Code is an advanced and custom environment for writing code, allowing the user to have endless capabilities, Microsoft Fabric notebooks offer a unified experience that integrates many tools within itself and relieves a great deal of work related to tracing data in extensive projects. Writing and executing python code in microsoft fabric is possible, just as it is in VS Code, however it is made more efficient due to being integrated with other microsoft fabric services such as lakehouses, data pipelines, and power bi.

Working in Fabric notebooks has one of its major advantages in that everything can be accomplished in one place. In VS Code, data management, storage, computation and visualization systems would need to be independently organized, usually needing extra processes and tools to achieve the desired integration. All the numerous services supported by Microsoft Fabric are tightly integrated and collocated in a single system, allowing for better productivity and teamwork.

Microsoft Fabric also has the benefit of being able to scale. When dealing with extensive data or more technical ML Models Machine Managed can automate infrastructure provisioning on-demand without setting up another infrastructure beforehand.

In addition, Microsoft Fabric has a friendly interface and sophisticated analytical features so that even inexperienced users and experienced ones can leverage its capabilities. Microsoft Fabric provides the necessary tools that allow you to customize your solution whether you need to create a straightforward model or execute complicated data science procedures.

Ultimately, with Microsoft Fabric notebooks, activities related to data science are further organized, strengthened by teamwork and are designed in a way so that these activities can grow in quantity. As a result a professional is able to use such tools more comfortably because there is no need to physically operate any structure. Because a single platform is offered to the client for everything, cross-team collaboration, working with big data, and model building are all possible, and the need to use diverse tools and platforms is reduced.