Data Science in Microsoft Fabric

🧪 What is Data Science in Microsoft Fabric?

Data Science in Fabric allows you to:

- **Ingest, clean, and explore data**
- **Build and train ML models** using notebooks
- **Use PySpark or Pandas**
- **Collaborate** with data engineers and analysts in a single workspace
- **Deploy and score models** in pipelines or notebooks

When you go to a **Lakehouse** in your **Microsoft Fabric workspace** and click on **"New" → "ML model"**, you get the option to:

◇ Select the data source (a table from your Lakehouse)

Then you'll see this:

⭕ Use AutoML
⭕ Use Notebook

This means Fabric gives you **two ways to build ML models**:

✨ Option 1: Use AutoML (No Code)

When you select **AutoML**, Fabric will:

1. Analyze your dataset (automatically profiles it)
2. Ask for a **target column** (e.g., `Churn`, `SalesAmount`)
3. Auto-detect the **problem type**: Classification or Regression
4. Automatically:
    a. Preprocess data
    b. Train many models (like RandomForest, XGBoost, LightGBM)
    c. Evaluate and rank models
    d. Select the **best performing one**

5. Show:
    a. Accuracy/ROC/AUC/RMSE depending on task
    b. Feature importance
    c. Scoring scripts

✅ Great for non-coders or quick POCs
❌ Limited control over preprocessing/custom tuning

🔍 Option 2: Use Notebook (Full Control)

If you choose **Notebook**, you can:

- Build models using:
    - `scikit-learn`
    - `pyspark.ml`
    - `xgboost`, `lightgbm`, etc.
- Customize preprocessing
- Save, load, or deploy models
- Visualize using `matplotlib`, `seaborn`, `plotly`
- Schedule and automate using pipelines

✅ Best for data scientists
✅ Full flexibility
❌ Requires Python/PySpark skills

ML Models

In **Microsoft Fabric**, when you use **AutoML** or create an **ML model**, you're asked to select an **ML task** — such as **Classification**, **Regression**, or **Forecasting**. Each of these tasks serves a different purpose depending on your business problem.

☑ 1. Classification

🎯 What It Is:

Classification is used when your **target variable is categorical** — i.e., it has distinct labels or classes.

- 🔸 Example: Will a customer churn? → `Yes` or `No`
- 🔸 Example: What is the fraud risk? → `High`, `Medium`, `Low`

📊 Typical Use Cases:

Use Case	Description
Customer Churn Prediction	Predict if a customer will leave
Fraud Detection	Classify transactions as fraud or not
Lead Scoring	Classify leads into hot, warm, cold

🧠 Algorithms Used:

- Logistic Regression
- Random Forest
- XGBoost
- Decision Tree
- LightGBM

🧪 Evaluation Metrics:

- Accuracy
- Precision / Recall
- F1 Score
- ROC-AUC

Classification in machine learning is further divided into:

✅ **Binary Classification**
✅ **Multiclass Classification**

Let's break them down in detail with examples, use cases, and how Microsoft Fabric handles them in AutoML.

☑ 1. Binary Classification

🎯 What it is:

A classification problem with **only two possible outcomes**.

- Example: Will a customer churn? → `Yes` or `No`
- Example: Is a transaction fraudulent? → `Fraud` or `Not Fraud`

📊 Use Cases:

Use Case	Target Labels
Churn Prediction	Yes / No
Loan Default	Default / No Default
Email Spam Detection	Spam / Not Spam
Click Prediction	Clicked / Not Clicked

🔍 Model Output:

- Returns a **probability** for each class
  E.g., `P(Churn) = 0.87` → Predict "Yes"

📏 Evaluation Metrics:

- Accuracy
- Precision
- Recall
- F1 Score
- ROC-AUC

☑️ 2. Multiclass Classification

🎯 What it is:

A classification problem with **more than two distinct labels** (3 or more classes).

- 🔸 Example: What customer segment is this? → `Gold`, `Silver`, `Bronze`
- 🔸 Example: Predict product category → `Electronics`, `Clothing`, `Books`

📊 Use Cases:

Use Case	Target Labels
Customer Segmentation	Gold / Silver / Bronze
Product Categorization	Furniture / Electronics / Groceries
Sentiment Analysis	Positive / Neutral / Negative

☑ 2. Regression

🎯 What It Is:

Regression is used when your **target variable is numeric/continuous** — you're predicting a **value**, not a category.

- 🔸 Example: Predict next month's sales in dollars
- 🔸 Example: Estimate delivery time in minutes

📊 Typical Use Cases:

Use Case	Description
Revenue Prediction	Forecast how much a customer will spend
Price Estimation	Estimate house/car prices
Campaign ROI	Predict ROI of ad campaigns

🧠 Algorithms Used:

- Linear Regression
- Random Forest Regressor
- Gradient Boosting Regressor
- LightGBM Regressor

🧪 Evaluation Metrics:

- RMSE (Root Mean Squared Error)
- MAE (Mean Absolute Error)
- R^2 (Coefficient of Determination)

☑ 3. Forecasting

🎯 What It Is:

Forecasting is a **time series prediction task** — used when your data changes over **time**, and you're predicting **future values based on time patterns**.

- 🔸 Example: Forecast product demand for next month
- 🔸 Example: Predict hourly website traffic for tomorrow

📊 Typical Use Cases:

Use Case	Description
Sales Forecasting	Predict future sales based on historical trends
Demand Planning	Forecast product demand or stock usage
Traffic Prediction	Predict footfall or clicks on websites

🧠 Algorithms Used:

- ARIMA / SARIMA
- Prophet (by Facebook)
- LightGBM (with time-based features)
- Exponential Smoothing

🧮 Special Features:

- AutoML in Fabric detects time-based fields
- Handles **seasonality**, **trend**, **holidays**, etc.
- You may need to specify:
    - Time column (e.g., `OrderDate`)
    - Granularity (Daily/Monthly/etc.)

🩹 Evaluation Metrics:

- MAPE (Mean Absolute Percentage Error)

- RMSE
- SMAPE

🧠 Summary Table

ML Task	Target Variable	Use Cases	Example Output	Metrics
Classification	Categorical (`Yes/No`, `Segment`)	Churn, fraud detection	"Yes" or "No"	Accuracy, AUC
Regression	Numeric (continuous)	Revenue, pricing, scoring	52.6, $500	RMSE, R^2
Forecasting	Time series (with date/time)	Sales forecast, traffic, demand	400 units (next week)	MAPE, RMSE

✨ In Microsoft Fabric (AutoML or ML Model)

When creating a model, Fabric will **auto-detect the task** if:

- You pick a categorical target → It uses **Classification**
- You pick a numeric target → It uses **Regression**
- You include a **datetime column + numeric target** → It switches to **Forecasting**

To create an AutoML model in Microsoft Fabric, you can choose between two main approaches: a low-code interface for ease of use or a code-based method for greater flexibility.

Option 1: Low-Code AutoML Interface

Ideal for users who prefer a guided setup without extensive coding.

1. **Access Microsoft Fabric**: Sign in to your Microsoft Fabric workspace.
2. **Switch Experience**: Use the experience switcher at the bottom left to select the **Data Science** experience.
3. **Launch AutoML Wizard**: From an existing experiment, model, or notebook item, launch the AutoML wizard.
4. **Select Data Source**: Choose your dataset from available lakehouses.

5. **Configure ML Task**: Specify the machine learning task (e.g., classification, regression) and basic configurations.
6. **Generate Notebook**: The AutoML UI will generate a pre-configured notebook tailored to your inputs.
7. **Execute and Track**: Run the notebook to train models. All metrics and iterations are automatically logged and tracked within existing ML experiments and model items.

🧠 What is Data Wrangler?

Data Wrangler is a notebook-based, no-code data preparation tool in Microsoft Fabric that accelerates exploratory data analysis and feature engineering. It offers a spreadsheet-like interface combined with dynamic summary stats, interactive visualizations, and a library of common data-cleaning operations. Every transformation you apply is previewed in real time and generates equivalent Python code (either pandas or PySpark), which you can export into your notebook for reproducibility and automation.

🛠️ How to Use Data Wrangler in Fabric

1. Prepare Your DataFrame

- Load a dataset into a pandas or Spark DataFrame in a Fabric notebook.
- Example:

```
python
CopyEdit
df = pd.read_csv('path/to/file.csv')
display(df)
```

- Ensure the notebook kernel is idle before launching Data Wrangler.

2. Launch Data Wrangler

- Under the notebook's ribbon (usually **Home**), select **Data Wrangler** and choose the active DataFrame.

- For Spark DataFrames, you can convert to pandas first or use the Spark-enabled Data Wrangler.

3. Explore and Clean Interactively

- Supports 20+ operations: sort, filter, rename, drop duplicates, fill missing values, change data type, one-hot encode, group/aggregate, scale numeric, split text, flash fill, and more.
- Items panels include:
    - Operations list
    - Cleaning steps (undo/delete)
    - Code preview
    - Summary stats and visual insights

4. Export the Code

- After transformations, you can export the Python code (as a function or inline into the notebook).
- Run the cell to execute the data cleaning pipeline—or integrate it into ETL processes or pipelines for scheduling later.