

Demand Forecasting

Gabriele Albini, Victoria Sicking



Why Demand Forecasting Matters

- Improves inventory management and reduces costs
- Enhances production planning and resource allocation
- Supports strategic business decisions and growth
- Optimizes pricing strategies for increased revenue

Creating an App + Dashboard for automated forecasting

Hackathon: Time Series Forecasting Python Tabs: OFF ☆

File Edit View Run Help Last edit was 5 days ago

Table of contents

- Building a Forecasting Model on ...
- Data Selection
- Data Preparation
- Model Selection
- Fit Data and Build Forecast
- Evaluation
- Log the model with MLflow
- Deploy Models on Databricks
- Build Forecast and Continuous Im...
- Now its your turn!

Data Selection

Load your training data from a delta table and split it into training and testing data.

```
catalog = "vs_test_demos"
schema = "hackathon"
table = "train"
forecast_horizon = 10

# Define the catalog, schema, and model name for organizing the model within the MLflow model registry
model_catalog = "vs_test_demos" #Update it to your catalog name
model_schema = "hackathon" #Update it to your schema name
model_name = "forecastingmodel" #Update it to your model name
```

```
#Select Data
query = f"SELECT date, store, SUM(sales) as sales FROM {catalog}.{schema}.{table} GROUP BY date, store ORDER BY date desc"

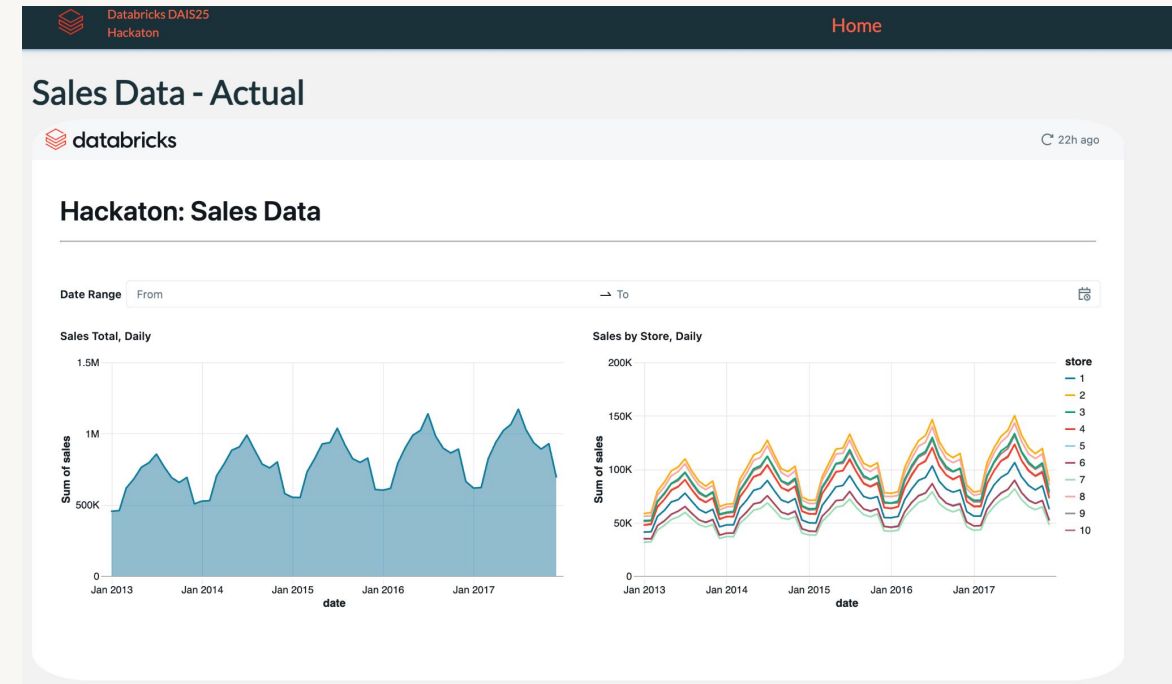
df = spark.sql(query)

# Choose a single store to make the calculations simpler
df = df.filter(df.store == 1)

# train-test-split
train_df = df.orderBy(df.date.asc()).limit(df.count() - forecast_horizon).orderBy(df.date.desc())
test_df = df.orderBy(df.date.desc()).limit(forecast_horizon).toPandas()

train_df.show(5)
test_df.head(5)
```

See performance (3)



What you will build

- Time Series Forecasting Notebook and Model Serving Endpoint
 - A Notebook is ready for you with the code and data to train a model, create forecasts and deploy a serving endpoint
 - You can add additional models and evaluation metrics
- AI/BI Dashboard
 - Based on the data and visualisations, you will build a dashboard to showcase the model performance and forecasts
- App
 - You will build a Databricks App, where you will implement a button that calls the serving endpoint to generate forecasts
 - You will embed the dashboard in the app

What we provide you and how to get started

- Sample data to train your model
- Example Notebook to get you started
- Access to a Databricks environment
- Instructions on how to embed a Dashboard in an App

Get Started

- Access the [Workspace](#)
 - Please use Compute with Databricks Runtime Version 15.4 LTS
- Clone the [Git Repository](#), and start with the notebook in the folder demand_forecasting/hackathon_resources
 - Use the backup folder only if you get stuck, and speak with your SAs to help you
- Explore the data in the schema [dev_hackathon.demand_forecasting](#)

How to embed a Dashboard in an App

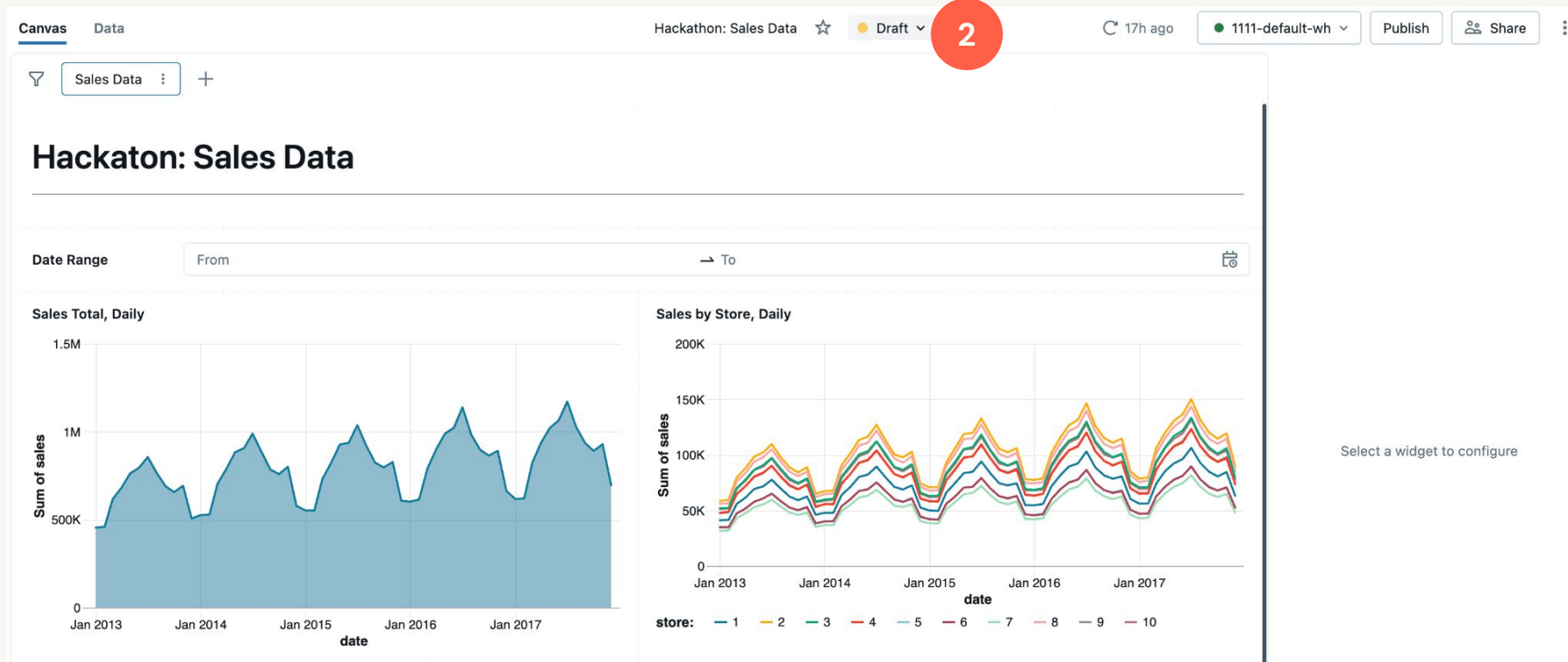
Part 1 – Creating a Dashboard

1 Create a new dashboard

The screenshot shows the Databricks Dashboards interface. On the left is a sidebar with navigation options: New, Workspace, Recents, Catalog, Workflows, Compute, Marketplace, SQL, SQL Editor, Queries, Dashboards (highlighted), Genie, Alerts, Query History, and SQL Warehouses. The main area is titled 'Dashboards' and has tabs for 'Dashboards' and 'Legacy dashboards'. In the top right corner, there is a 'View samples gallery' button and a 'Create dashboard' button with a dropdown arrow. A red circle with the number 1 highlights the 'Create dashboard' button. Below the 'Create dashboard' button is a dropdown menu with options: 'Create dashboard' and 'Import dashboard from file'. The main area displays a 'Featured' section with four dashboard cards: 'Account Usage Dashboard' by Daniel Martinez Arevalo, 'Portfolio Risk Dashboard' by ricardo portilla, 'DbDemos - AIBI - Customer Support Team Review' by Axel Richier, and 'DaisDemos - AIBI - Customer Support Team Review' by Axel Richier. At the bottom, there is a search bar 'Filter dashboards' and filters for 'All', 'Favorites', 'Popular', 'Last modified', and 'Owner: Me (gabriele.albini@databricks.com)'.

How to embed a Dashboard in an App

2 Edit datasets and canvas in Draft mode



How to embed a Dashboard in an App

- 3 Switch to Published Mode
- 4 Click on Share
- 5 Open the Embed dashboard details and 6 copy Iframe code

The screenshot illustrates the steps to embed a dashboard in an application. It shows the Databricks interface with a dashboard titled 'Hackathon: Sales Data'. The dashboard displays a line chart of 'Sum of sales' over time, with a date range selector set to 'From'. The chart shows sales data from January 2013 to January 2017, with a peak around 1.5M in early 2016.

The 'Sharing: Hackathon: Sales Data' dialog is open, showing sharing settings. The 'People with access' section lists 'Gabriele Albini' (Can Manage (inherited)), 'Admins' (Can Manage (inherited)), and 'All workspace users' (Can View (inherited)). The 'Sharing settings' section shows 'Only people with access can view'. The 'Copy link' and 'Embed dashboard' buttons are visible at the bottom of the dialog.

The 'Copy embed code' panel is also shown, displaying the HTML code for embedding the dashboard using an iframe. The code is as follows:

```
<iframe
  src="https://e2-demo-field-eng.cloud.databricks.com/embed/dashboardsv3/01f05046c
  width="100%"
  height="600"
  frameborder="0">
</iframe>
```


How to embed a Dashboard in an App

Part 2 – Create your App

- 1 Navigate to the Compute section
- 2 Click on Apps
- 3 Create a new app

The screenshot shows the Databricks web interface. On the left sidebar, the 'Compute' link is highlighted with a red circle labeled '1'. The main content area is titled 'Compute' and has a red circle labeled '2' over the 'Apps' tab. Below the tabs, there are filters and a 'Create app' button highlighted with a red circle labeled '3'. The interface includes a search bar at the top, a sidebar with navigation links, and a main content area with various tabs and filters.

Compute

All-purpose compute Job compute SQL warehouses Vector Search Pools GPU pools Policies **Apps** OLTP Database ⓘ

Filter by keyword ☐ Owned by me Compute status Creator Last updated ☐ Installed from Marketplace 200 matches [Send feedback](#) [Create app](#)

| Name | Compute | Creator | Last update | App URL |
|------|---------|---------|-------------|---------|
|------|---------|---------|-------------|---------|

How to embed a Dashboard in an App

4 Choose your support framework and 5 deploy a simple boilerplate app


Develop your app (e.g., using Notebooks), including the iFrame component 

6 Redeploy with the new source code path

Compute > Apps >






Create new app


Create custom app from scratch




Create a custom app
Bring your code and resources to build an app from scratch

Install from a template

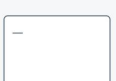
 Dash  Flask  Gradio  Shiny  Streamlit



Chatbot
Create a chat UI using a large language model and Model Serving.
Dependencies: Serving endpoint



Data app
Create an app that reads from a SQL warehouse and visualizes the data using the app user's permissions.
Dependencies: SQL warehouse
User authorized API scopes: sql



Hello world
Create a simple app.

Compute > Apps >

test-hello-world-app

[Send feedback](#) [Edit](#) [Permissions](#) [Deploy](#)

Overview Authorization Deployments Logs Environment

Running <https://test-hello-world-app-1444828305810485.aws.databricksapps.com>

Deployment
Source: [/Workspace/Users/gabriele.albini@databricks.com/databricks_apps/test-hello-world-app_2025_06_24-12_32/dash-hello-world-app](#)

✓ Last deploy completed 12 seconds ago [Inspect deployed code](#)

About this app
Creator: gabriele.albini@databricks.com
Created on: 3 minutes ago
Last update: 24 seconds ago
App ID: 94aaddbf-4815-4fd1-9745...

