

Azure spark cluster (synapse) creation steps

Step1: Create an Azure account.

Step2: Go to azure and search for Synapse.

Step3: Create a Synapse resource.

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with 'Microsoft Azure', a search bar, and various icons. On the right, it shows the user's email 'abotta@stevens.edu' and the Stevens logo. Below the navigation bar, the main content area displays the 'bigdatasynapse' Synapse workspace. The left sidebar has a tree view with nodes like 'Overview', 'Essentials', 'Getting started', and sections for 'Analytics pools', 'Security', 'Monitoring', 'Automation', and 'Help'. The 'Essentials' section contains detailed information about the workspace, including its Resource group ('Bigdata'), Status ('Succeeded'), Location ('East US'), Subscription ('Azure subscription 1'), and various endpoints. The 'Getting started' section provides links to 'Open Synapse Studio' and 'Read documentation'. The 'Analytics pools' section shows a table with columns 'Name', 'Type', and 'Size', which is currently empty. At the bottom left, there's a note: 'Add or remove favorites by pressing Ctrl+Shift+F'.

Step4: Launch the synapse workspace through Workspace URL.

Step5: Now go to manage and create an Apache Spark Pools.

The screenshot shows the Microsoft Azure Synapse Analytics interface. The left sidebar has 'Manage' selected. The main area shows the 'Apache Spark pools' section, which is currently empty. A large button at the bottom right says 'New Apache Spark pool'.

The screenshot shows the 'Create Apache Spark pool' wizard. The 'Basics' tab is selected. The form includes fields for 'Apache Spark pool name' (set to 'bigdatapool'), 'Isolated compute' (set to 'Disabled'), 'Node size family' (set to 'Memory Optimized'), 'Node size' (set to 'Large (16 vCores / 128 GB)'), 'Autoscale' (set to 'Enabled'), 'Number of nodes' (set to '3'), 'Estimated price' (showing 'Est. cost per hour: 6.62 to 6.62 USD'), and 'Dynamically allocate executors' (set to 'Enabled').

Step6: Once a pool is created then go to Develop page and create/import an notebook.

Step7: Select the pool which you created.

Microsoft Azure | Synapse Analytics > bigdatasynapse

We use optional cookies to provide a better experience. Learn more

Accept | Reject | More options

Home Data Develop Integrate Monitor Manage

Synapse live Validate all Publish all

Develop Random_Forest_2E... Filter resources by name

Notebooks Random_Forest_2E_100D Not started

Attach to bigdatapool Large, 3 to 3 nodes

Manage pools

Properties General Related (0)

Name * Random_Forest_2E_100D

Description

Type .ipynb notebook

Size 75,660 bytes

Notebook settings

Include cell output when saving

Enable unpublished notebook reference

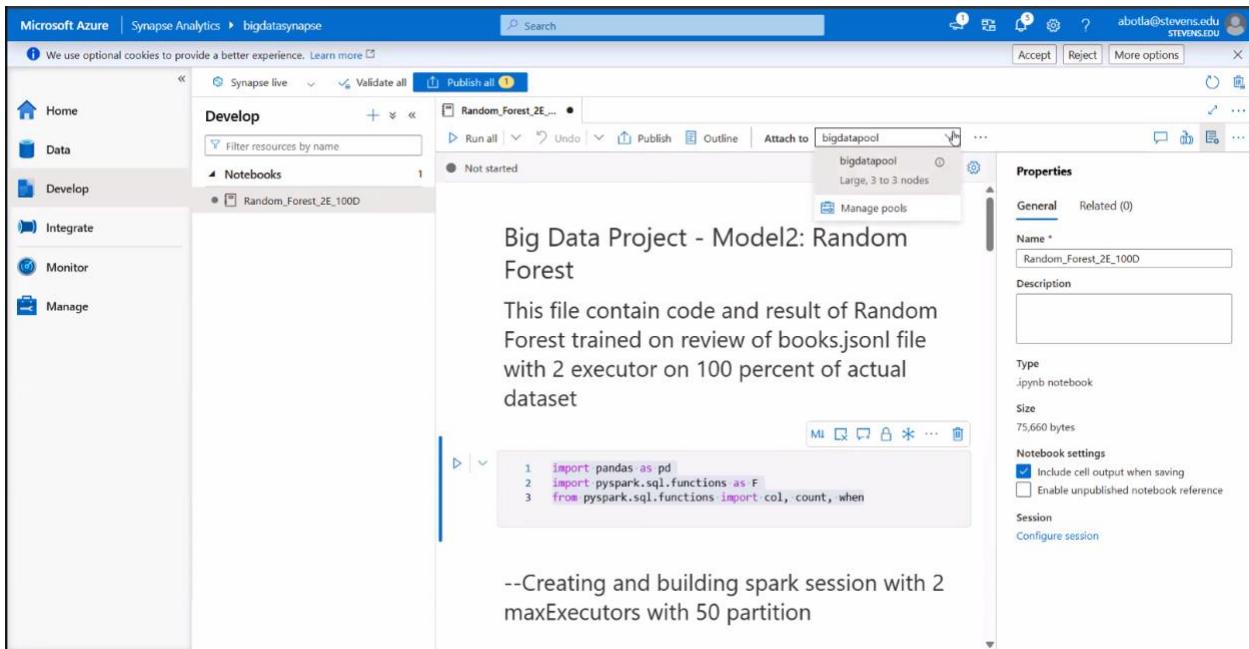
Session Configure session

Big Data Project - Model2: Random Forest

This file contain code and result of Random Forest trained on review of books.jsonl file with 2 executor on 100 percent of actual dataset

```
1 import pandas as pd
2 import pyspark.sql.functions as F
3 from pyspark.sql.functions import col, count, when
```

--Creating and building spark session with 2 maxExecutors with 50 partition



Step7: Use your Azure blob storage and get the SAS token like below to load the dataset.

Microsoft Azure | Synapse Analytics > bigdatasynapse

We use optional cookies to provide a better experience. Learn more

Accept | Reject | More options

Home Data Develop Integrate Monitor Manage

Synapse live Validate all Publish all

Develop Random_Forest_2E... Filter resources by name

Notebooks Random_Forest_2E_100D Not started

Attach to bigdatapool Language PySpark (Python)

Variables

Properties General Related (0)

Name * Random_Forest_2E_100D

Description

Type .ipynb notebook

Size 75,919 bytes

Notebook settings

Include cell output when saving

Enable unpublished notebook reference

Session Configure session

Big Data Project - Model2: Random Forest

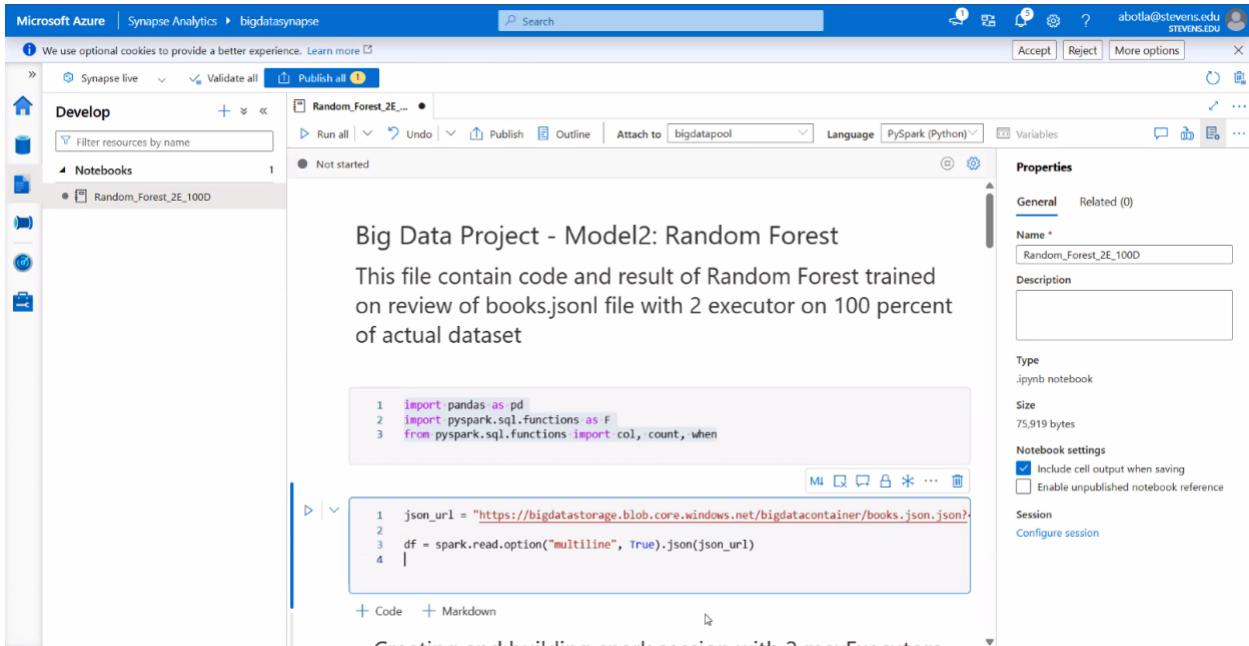
This file contain code and result of Random Forest trained on review of books.jsonl file with 2 executor on 100 percent of actual dataset

```
1 import pandas as pd
2 import pyspark.sql.functions as F
3 from pyspark.sql.functions import col, count, when
```

```
1 json_url = "https://bigdatastorage.blob.core.windows.net/bigdatacontainer/books.json?&sv=2018-03-28&ss=bf&sr=c&sig=...&tso=t&st=2018-03-28T12%3A00%3A00Z&se=2018-03-28T12%3A30%3A00Z"
2 df = spark.read.option("multiline", True).json(json_url)
3
```

+ Code + Markdown

Creating and building spark session with 2 maxExecutors



If you have not loaded the dataset then load the dataset in Azure storage account and it should be like below.

The screenshot shows the Microsoft Azure Storage accounts interface. On the left, there's a sidebar with a search bar and links for 'Create', 'Storage accounts', and 'Group by none'. A message box says: 'You are viewing a new version of Browse experience. Some features may be missing. Click here to access the old experience.' Below this are sections for 'Name' (with 'accountbookreview' selected), 'Tags' (with 'bookreviewcluster'), and a list of other storage accounts. At the bottom of the sidebar, it says 'Showing 1 - 2 of 2. Display count: 10'. The main area is titled 'accountbookreview' and shows the 'Overview' tab. It includes a 'Search' bar and a toolbar with 'Upload', 'Open in Explorer', 'Delete', 'Move', 'Refresh', 'Open in mobile', 'CLI / PS', and 'Feedback'. The 'Essentials' section displays details such as Resource group (move), Location (eastus), Subscription (Azure subscription 1), Disk state (Available), and Tags. The 'Properties' tab is active, showing sections for 'Blob service' and 'Security'. Under 'Blob service', settings include Hierarchical namespace (Disabled), Default access tier (Hot), Blob anonymous access (Disabled), Blob soft delete (Enabled (7 days)), and Container soft delete (Enabled (7 days)). Under 'Security', settings include Require secure transfer for REST API operations (Enabled), Storage account key access (Enabled), Minimum TLS version (Version 1.2), and Infrastructure encryption (Disabled). The top right corner shows the user's email (abotta@stevens.edu) and name (STEVENSLDU STEVENS ONMIC...).

Step8: With the Spark pool configured and the data loaded, you can now run your PySpark code successfully within the Synapse notebook environment.

Thank You