

Azure Data Science Project

Solution:

Connect to SQL Server instance using these credentials:

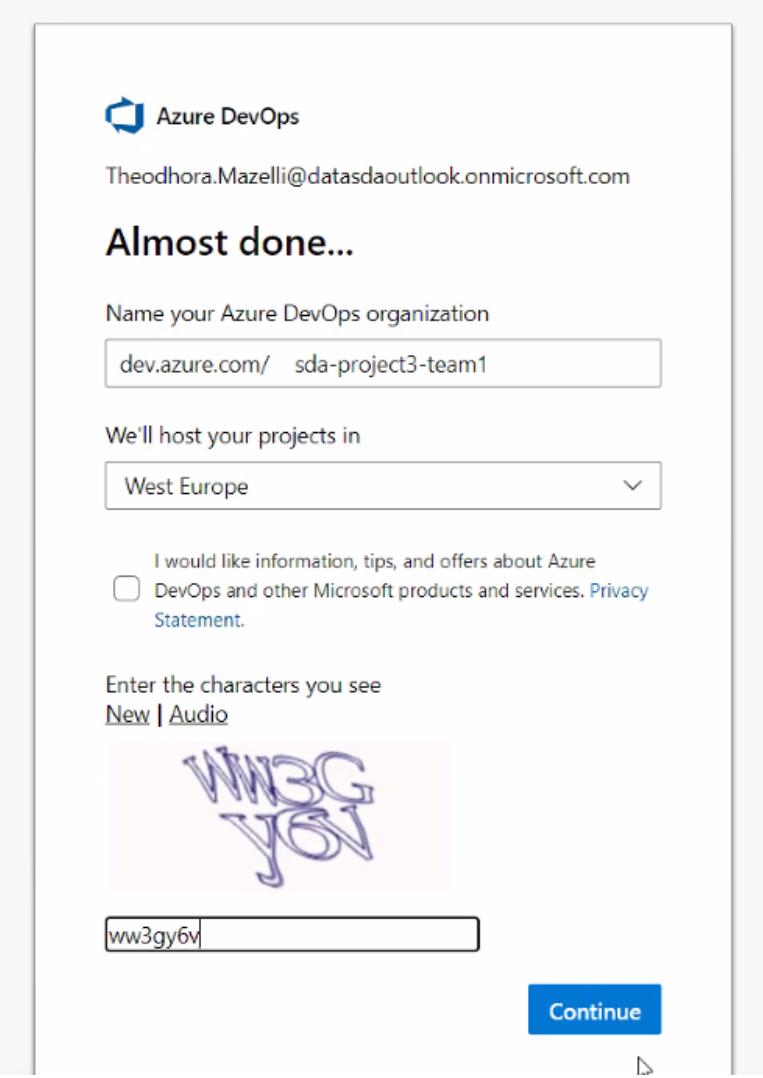
Server Name: rg-data-project.database.windows.net

Login: sqladminuser

Password: Admin123+

For this project we will use Azure Data Factory, Azure Portal, Azure Devops Portal and SQL Server and Streamlit.

In Azure DevOps let's create a new organisation:



The screenshot shows the second step of the Azure DevOps organization creation wizard. The title "Almost done..." is displayed prominently. The user has entered "dev.azure.com/ sda-project3-team1" as the organization name. They have selected "West Europe" as the location. A checkbox for receiving Azure DevOps information is checked, with a link to the Privacy Statement. A CAPTCHA challenge "WW3BG Y6V" is shown with the input field containing "ww3gy6v". A "Continue" button is at the bottom right, and a "Back" button is at the bottom center.

Azure DevOps
Theodhora.Mazelli@datasdaoutlook.onmicrosoft.com

Almost done...

Name your Azure DevOps organization
dev.azure.com/ sda-project3-team1

We'll host your projects in
West Europe

I would like information, tips, and offers about Azure DevOps and other Microsoft products and services. [Privacy Statement](#).

Enter the characters you see
[New](#) | [Audio](#)

WW3BG Y6V

ww3gy6v

Continue

Back

After the organisation is created we should create a new project:

Create a project to get started

Project name *

data-project3



Visibility



Public ⓘ

Anyone on the internet can view the project. Certain features like TFVC are not supported.



Private

Only people you give access to will be able to view this project.



Public projects are disabled for your organization. You can turn on public visibility with [organization policies](#).

+ Create project

Add the members in the project:

Invite members to data-project3



Search and add users to your data-project3

Users

XB Xhoana Balla X MH Mohammad Hovaidi-Ardestani X

Use semicolons to separate multiple email addresses.

Add to team(s)

data-project3 Team



- (i) Xhoana Balla, Mohammad Hovaidi-Ardestani has not been assigned an access level, we will attempt to assign Stakeholder.
[Learn more](#)

In the azure portal let's create a new datafactory:

The screenshot shows the 'Data factories' page in the Azure portal. At the top, there's a search bar and several filter options: 'Subscription -- all', 'Type -- all', 'Resource group -- all', 'Location -- all', and a 'Add filter' button. Below the filters, there's a dropdown for 'No grouping' and a 'List view' button. The main area displays a list of data factories, with two items visible: 'dfgroup2' and 'Team3-ProjectData'. The 'dfgroup2' item has three vertical dots on its right side.

Fill in the information for the creation of the data factory.

The screenshot shows the 'Microsoft.DataFactory-20220419102225 | Overview' page. At the top, there's a deployment status message: 'Deployment in progress... Deployment to resource group 'rg-data-project' is in progress.' Below the message, there's a 'Search (Ctrl+/' button and a row of actions: 'Delete', 'Cancel', 'Redeploy', and 'Refresh'. The left sidebar includes 'Overview', 'Inputs', 'Outputs', and 'Template' sections. The main content area shows 'Deployment details' with a table header: 'Resource', 'Type', 'Status', and 'Operation details'. The table body says 'No results.'

The data factory is created:

The screenshot shows the 'df-team-1 Data factory (V2)' overview page. The left sidebar contains sections like 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Settings', 'Networking', 'Managed identities', 'Properties', 'Locks', 'Getting started', 'Quick start', 'Monitoring', 'Alerts', and 'Metrics'. The main content area includes a 'Delete' button, an 'Essentials' section with details such as Resource group (rg-data-project), Status (Succeeded), Location (East US), Subscription (Azure subscription SDA), and Subscription ID (a1dafe08-210f-4407-912c-af39132fa32); a 'Getting started' section with links to 'Open Azure Data Factory Studio' and 'Read documentation'; and a 'Monitoring' section with 'PipelineRuns' and 'ActivityRuns' tabs.

We connect the datafactory to the project we have created in devOpd portal:

The screenshot shows the 'Configure a repository' section of the Azure Data Factory setup. It includes fields for 'Repository type' (set to 'Azure DevOps Git'), 'Cross tenant sign in' (unchecked), and 'Azure Active Directory' (set to 'Katalog domyślny').

For configuring the repository we fill these information;

Configure a repository

Katalog domyślny (21920c0e-892c-4db1-96a8-f16f40d56cef)

Specify the settings that you want to use when connecting to your repository.

Select repository Use repository link

Azure DevOps organization name * ⓘ

sda-project3-team1

Project name * ⓘ

data-project3

Repository name * ⓘ

data-project3

Collaboration branch * ⓘ

dev-branch

Publish branch * ⓘ

adf_publish

adf_publish

Root folder * ⓘ

/

Import existing resource

Import existing resources to repository

Import resource into this branch ⓘ

Apply

Back

Cancel

New linked service

 Azure Data Lake Storage Gen2 [Learn more](#) 



Connect via integration runtime * 

AutoResolveIntegrationRuntime 

Authentication type

Account key 

Account selection method 

From Azure subscription Enter manually

Azure subscription 

Azure subscription SDA (a1dafe08-210f-4407-912c-afd39132fa32) 



Storage account name *

dataprojectsda 

Test connection 

To linked service To file path

Annotations

 New

> Parameters 

> Advanced 

Let's create linked service with sql:

New linked service

Data store Compute

database

All Azure Database File Generic protocol NoSQL Services and apps

Azure Database for MariaDB Azure Database for MySQL Azure Database for PostgreSQL

Azure SQL Database Azure SQL Database Managed Instance

New linked service

Azure SQL Database [Learn more](#)

Connect via integration runtime * ⓘ

AutoResolveIntegrationRuntime

Connection string **Azure Key Vault**

Account selection method ⓘ

From Azure subscription Enter manually

Azure subscription

Azure subscription SDA (a1dade08-210f-4407-912c-af39132fa32)

Server name *

rg-data-project

Database name *

rg-data-project-team1

Authentication type *

SQL authentication

User name *

sqladminuser

Add dynamic content [Alt+Shift+D]

Password **Azure Key Vault**

Password *

Create import a dataset:

New dataset

In pipeline activities and data flows, reference a dataset to specify the location and structure of your data within a data store. [Learn more](#)

Select a data store

Search

All Azure Database File Generic protocol NoSQL Services and apps

Azure Blob Storage	(MongoDB API)	API)
 Azure Data Explorer (Kusto)	 Azure Data Lake Storage Gen1	 Azure Data Lake Storage Gen2
 Azure Database for MariaDB	 Azure Database for MySQL	 Azure Database for PostgreSQL

Set properties

Name

loadCustomerInfo

Linked service *

AzureDataLakeStorage1



File path

wwi-02

/ customer-info

/ customerinfo.csv



First row as header



Import schema

From connection/store From sample file None

Browse

Select a file or folder.

Root folder > wwi-02 > customer-info



Import another csv file:

Factory Resources ▼ «

Filter resources by name +

▶ Pipelines	0
◀ Datasets	2
● campaignAnalytics	
loadCustomerInfo	

Now we will import the json file:

Select format

Choose the format type of your data

 Avro	 Binary	 DelimitedText
 Excel	 JSON	 ORC
 Parquet	 XML	

Continue Back Cancel

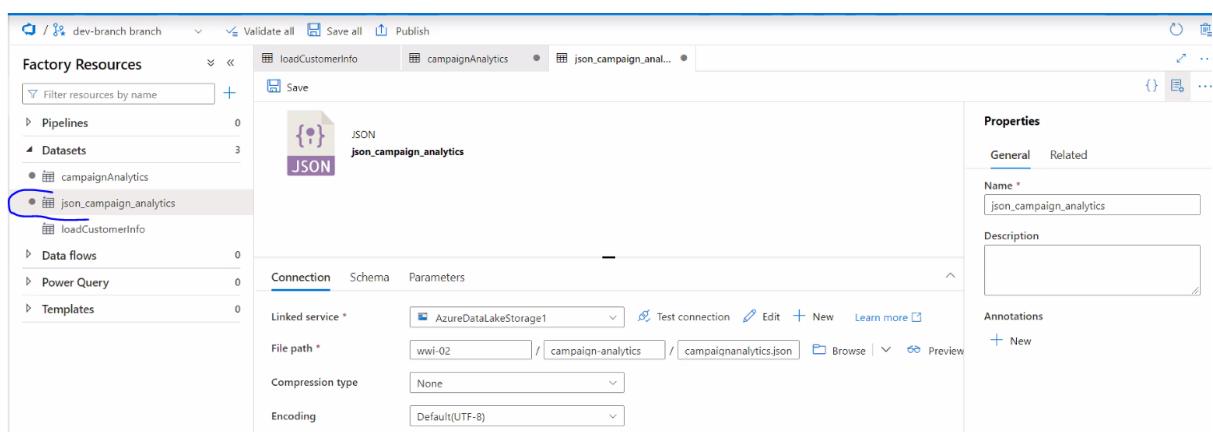
Set properties

Name
json_campaign_analytics

Linked service *
AzureDataLakeStorage1

File path
File system / Directory / File

Import schema
 From connection/store From sample file None



The screenshot shows the Azure Data Factory studio interface. On the left, there's a navigation pane with 'Factory Resources' like Pipelines, Datasets, Data flows, Power Query, and Templates. A dataset named 'json_campaign_analytics' is selected and highlighted with a blue circle. The main workspace shows the dataset configuration. It includes a preview icon, a save button, and tabs for Connection, Schema, and Parameters. Under Connection, it specifies 'AzureDataLakeStorage1' as the linked service and the file path 'wwi-02/campaign-analytics/campaignanalytics.json'. The 'Import schema' section has 'None' selected. The 'Properties' panel on the right shows the dataset's name is 'json_campaign_analytics'.

Now we will create the pipelines for displaying the data we have into the SQL server. For doing this we will create the tables in our database in SQL Server.

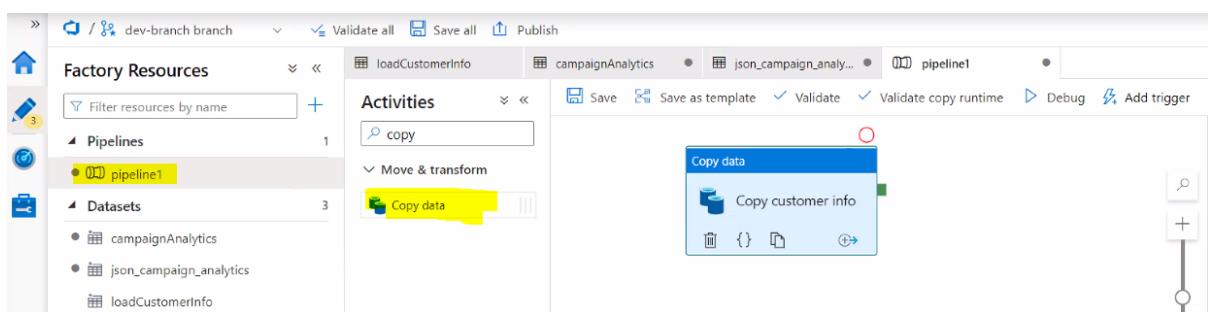
```
CREATE TABLE customerInfo
(
username VARCHAR(50),
gender VARCHAR(10),
phone_number VARCHAR(50),
email VARCHAR(50),
creditcard VARCHAR(50)
)
```

```

CREATE TABLE customerInfo
(
username VARCHAR(50),
gender VARCHAR(10),
phone_number VARCHAR(50),
email VARCHAR(50),
creditcard VARCHAR(50)
)

```

In Azure Data Factory let's create a pipeline to load the data:



We create a dataset for loading the data:

New dataset

In pipeline activities and data flows, reference a dataset to specify the location and structure of your data within a data store. [Learn more](#)

Select a data store

Azure Database for MariaDB	Azure Database for MySQL	Azure Database for PostgreSQL
Azure SQL Database	Azure SQL Database Managed Instance	

Set properties

Name

Linked service *

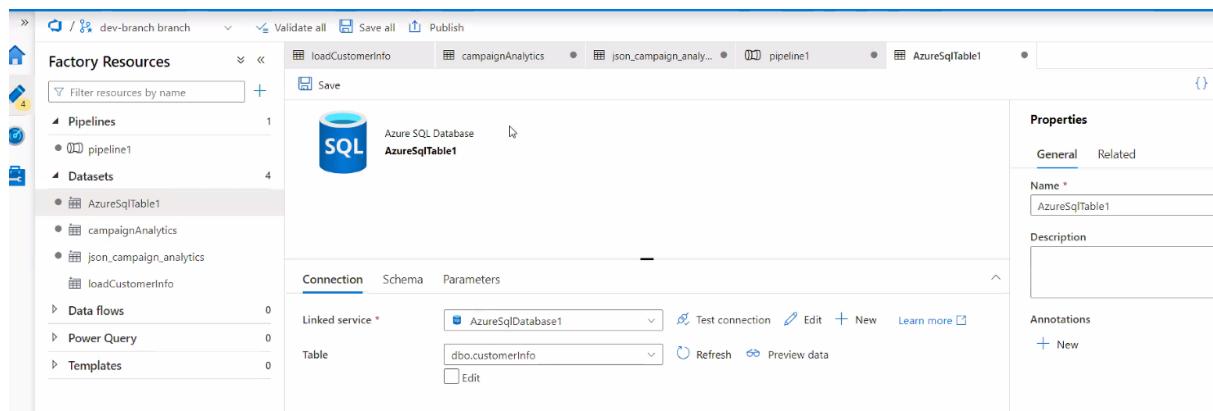
Table name

  Edit

Import schema

From connection/store None

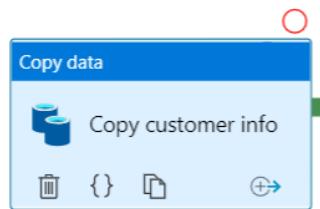
The file is created:



The screenshot shows the Azure Data Factory interface. On the left, there's a navigation pane titled 'Factory Resources' with sections for Pipelines, Datasets, Data flows, Power Query, and Templates. Under 'Datasets', 'AzureSqlTable1' is selected. In the center, a preview window shows a blue cylinder icon labeled 'SQL' and the text 'Azure SQL Database' and 'AzureSqlTable1'. Below this, there are tabs for 'Connection', 'Schema', and 'Parameters'. The 'Connection' tab is active, showing 'Linked service *' set to 'AzureSqlDatabase1' and 'Table' set to 'dbo.customerInfo'. On the right, a 'Properties' panel is open with tabs for 'General' and 'Related'. The 'General' tab shows 'Name *' set to 'AzureSqlTable1' and a 'Description' field. There are also 'Annotations' and a '+ New' button.

Now let's configure the pipeline:

See below:



General Source Sink Mapping Settings User properties

Name * [Learn more](#)

Description

Timeout ⓘ

Retry ⓘ

Retry interval (sec) ⓘ

Secure output ⓘ

Secure input ⓘ

Source:

General **Source** Sink Mapping Settings User properties

Source dataset * [Open](#) [New](#) [Preview data](#) [Learn more](#)

File path type File path in dataset Wildcard file path List of files ⓘ

Start time (UTC)
End time (UTC)

Filter by last modified ⓘ
Recursively

Enable partition discovery

Max concurrent connections ⓘ

Skip line count

Additional columns ⓘ [New](#)

Sink:

General Source **Sink** Mapping Settings User properties

Sink dataset *

Write behavior Insert Upsert Stored procedure

Bulk insert table lock Yes No

Table option None Auto create table

Pre-copy script

Write batch timeout

Write batch size

Max concurrent connections

Mapping:

General Source Sink **Mapping** Settings User properties

> Type conversion settings

Source	Type		Destination	Type
<input type="text" value="UserName"/>	<input type="text" value="String"/>	<input type="text"/>	<input type="text" value="username"/>	<input type="text" value="varchar"/>
<input type="text" value="Gender"/>	<input type="text" value="String"/>	<input type="text"/>	<input type="text" value="gender"/>	<input type="text" value="varchar"/>
<input type="text" value="Phone"/>	<input type="text" value="String"/>	<input type="text"/>	<input type="text" value="phone_number"/>	<input type="text" value="varchar"/>
<input type="text" value="Email"/>	<input type="text" value="String"/>	<input type="text"/>	<input type="text" value="email"/>	<input type="text" value="varchar"/>
<input type="text" value="CreditCard"/>	<input type="text" value="String"/>	<input type="text"/>	<input type="text" value="creditcard"/>	<input type="text" value="varchar"/>

Let's debug the pipeline:

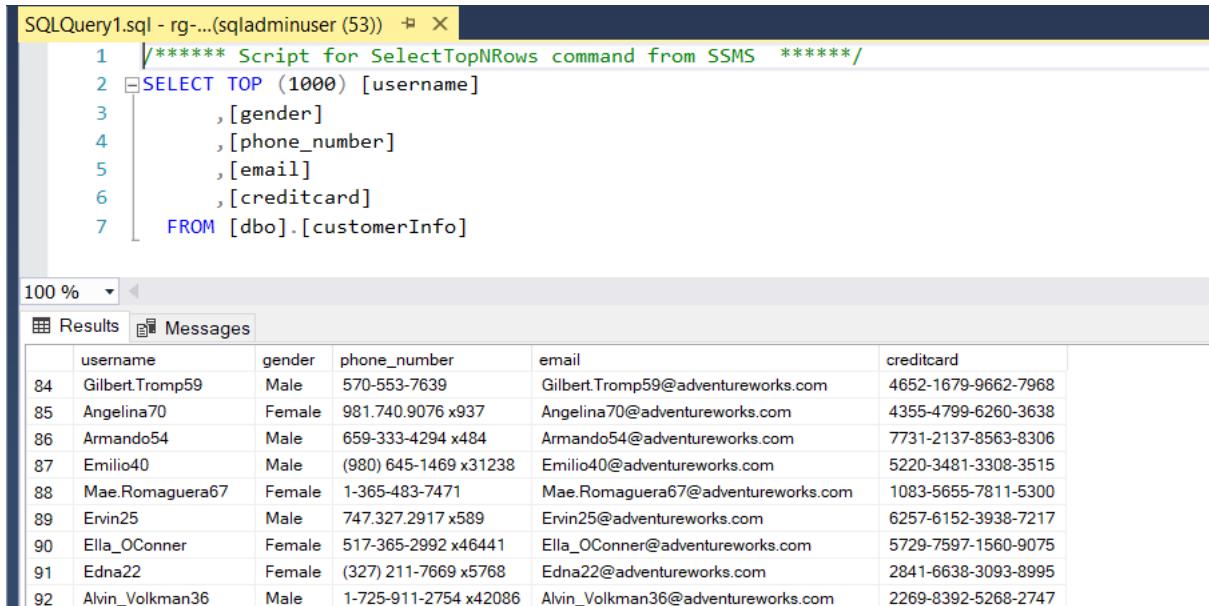
The screenshot shows the Azure Data Factory pipeline editor. A 'Copy customer info' activity is selected. The 'Output' tab is active, displaying a table with one row: 'Copy customer info' (Copy data), run at 2022-04-19T09:37:11.08, duration 0:00:15, status Queued, and integration runtime AutoResolveIntegrationRuntime.

The screenshot shows the 'Output' details for the 'Copy customer info' activity. It displays performance metrics: data read (9844), data written (18288), files read (1), source peak connections (1), sink peak connections (2), rows read (110), rows copied (110), copy duration (4), throughput (2.403), and errors (0).

There are the details telling how data are passing from one location to another.

The screenshot shows the 'Details' view for the 'Copy customer info' activity. It details the flow from 'Azure Data Lake Storage Gen2' (Region: North Europe) to 'Azure SQL Database' (Region: West Europe). The status is 'Succeeded'. Performance metrics include data read (9.613 KB), files read (1), rows read (110), peak connections (1), data written (17.859 KB), rows written (110), and peak connections (2). Copy duration was 0:00:05 with a throughput of 1.923 KB/s. The 'Start time' was 4/19/22, 2:23:00 PM, and 'Used DIUs' were 4. The 'Duration' table shows working duration (0:00:02) and total duration (0:00:00). A note indicates the activity ran from 'Queue'.

If we go to SQL server and do a select statement we will see the data are loaded there:



The screenshot shows the SSMS interface with a query window titled "SQLQuery1.sql - rg...(sqladminuser (53))". The query is:

```
1  /***** Script for SelectTopNRows command from SSMS *****/
2  SELECT TOP (1000) [username]
3      ,[gender]
4      ,[phone_number]
5      ,[email]
6      ,[creditcard]
7  FROM [dbo].[customerInfo]
```

The results grid displays 10 rows of data:

	username	gender	phone_number	email	creditcard
84	Gilbert.Tromp59	Male	570-553-7639	Gilbert.Tromp59@adventureworks.com	4652-1679-9662-7968
85	Angelina70	Female	981.740.9076 x937	Angelina70@adventureworks.com	4355-4799-6260-3638
86	Armando54	Male	659-333-4294 x484	Armando54@adventureworks.com	7731-2137-8563-8306
87	Emilio40	Male	(980) 645-1469 x31238	Emilio40@adventureworks.com	5220-3481-3308-3515
88	Mae.Romaguera67	Female	1-365-483-7471	Mae.Romaguera67@adventureworks.com	1083-5655-7811-5300
89	Ervin25	Male	747.327.2917 x589	Ervin25@adventureworks.com	6257-6152-3938-7217
90	Ella_OConner	Female	517-365-2992 x46441	Ella_OConner@adventureworks.com	5729-7597-1560-9075
91	Edna22	Female	(327) 211-7669 x5768	Edna22@adventureworks.com	2841-6638-3093-8995
92	Alvin_Volkman36	Male	1-725-911-2754 x42086	Alvin_Volkman36@adventureworks.com	2269-8392-5268-2747

Now let's import the json dataset.

Set properties

Name

Json_campaign_analytics

Linked service *

AzureDataLakeStorage1



File path

wwi-02

/ campaign-analytics

/ campaignanalytics.json



Import schema

From connection/store From sample file None

Create a new dataset to store the data:

Set properties

Name
AzureSqlTable2

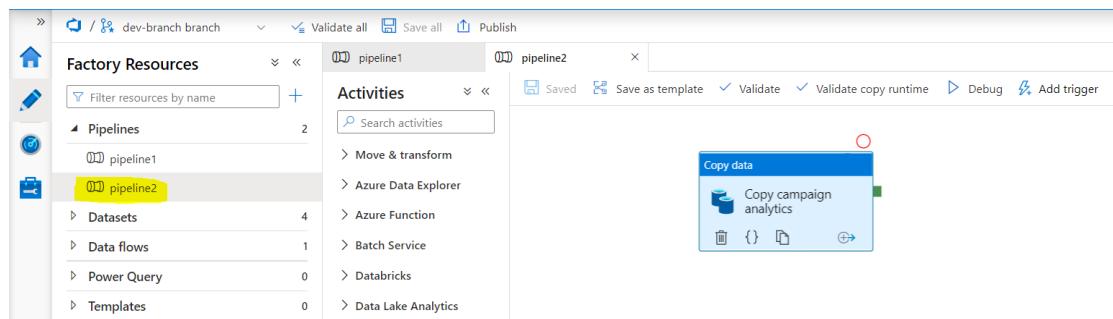
Linked service *
AzureSqlDatabase1

Table name
dbo.CampaignAnalytics

Edit

Import schema
 From connection/store None

Here we create a new pipeline for loading the data to the SQL server.



The screenshot shows the Azure Data Factory pipeline editor. On the left, the 'Factory Resources' sidebar lists 'Pipelines' (with 'pipeline1' and 'pipeline2' selected), 'Datasets' (4 datasets), 'Data flows' (1 flow), 'Power Query' (0 queries), and 'Templates' (0 templates). In the center, the 'Activities' pane shows a search bar and a list of available activities: 'Move & transform', 'Azure Data Explorer', 'Azure Function', 'Batch Service', 'Databricks', and 'Data Lake Analytics'. A modal window titled 'Copy data' is open, showing the 'Copy campaign analytics' activity. At the top, there are tabs for 'Saved', 'Save as template', 'Validate', 'Validate copy runtime', 'Debug', and 'Add trigger'.

The configuration of the pipeline is as below:

General Source Sink Mapping Settings User properties

Name * Copy campaign analytics [Learn more](#)

Description

Timeout ⓘ 7.00:00:00

Retry ⓘ 0

Retry interval (sec) ⓘ 30

Secure output ⓘ

Secure input ⓘ

Source:

General **Source** Sink Mapping Settings User properties

Source dataset * [Open](#) [New](#) [Preview data](#) [Learn more](#)

File path type File path in dataset Wildcard file path List of files

Filter by last modified Start time (UTC)
End time (UTC)

Recursively

Enable partition discovery

Max concurrent connections

Additional columns [New](#)

Sink:

General Source **Sink** Mapping Settings User properties

Sink dataset * [Open](#) [New](#) [Learn more](#)

Write behavior Insert Upsert Stored procedure

Bulk insert table lock Yes No

Table option None Auto create table

Pre-copy script

Write batch timeout

Write batch size

Max concurrent connections

Mapping:

General Source Sink **Mapping** Settings User properties

Import schemas + New mapping ⚡ Clear ⓘ Delete Advanced editor

Collection reference ⓘ

Map complex values to string

Name	Type	Collection reference	Column name	Include
Region	abc string	→	Region	<input checked="" type="checkbox"/>
Country	abc string	→	Country	<input checked="" type="checkbox"/>
Product_Category	abc string	→	ProductCategory	<input checked="" type="checkbox"/>
Campaign_Name	abc string	→	undefined	<input checked="" type="checkbox"/>
Revenue	123 integer	→	Revenue	<input checked="" type="checkbox"/>
Revenue_Target	ANY null	→	undefined	<input checked="" type="checkbox"/>
City	abc string	→	City	<input checked="" type="checkbox"/>
State	abc string	→	State	<input checked="" type="checkbox"/>
Column_1	ANY any	→	undefined	<input checked="" type="checkbox"/>

Add dynamic content [Alt+Shift+D]

Setting:

General Source Sink Mapping **Settings** User properties

ⓘ You will be charged # of used DIUs * copy duration * \$0.25/DIU-hour. Local curre

Data integration unit ⓘ

Auto

Edit

Degree of copy parallelism ⓘ

Edit

Data consistency verification ⓘ

Fault tolerance ⓘ

Enable logging ⓘ

Enable staging ⓘ

Now we will create a dataflow in azure in order to transform/clean the json file.

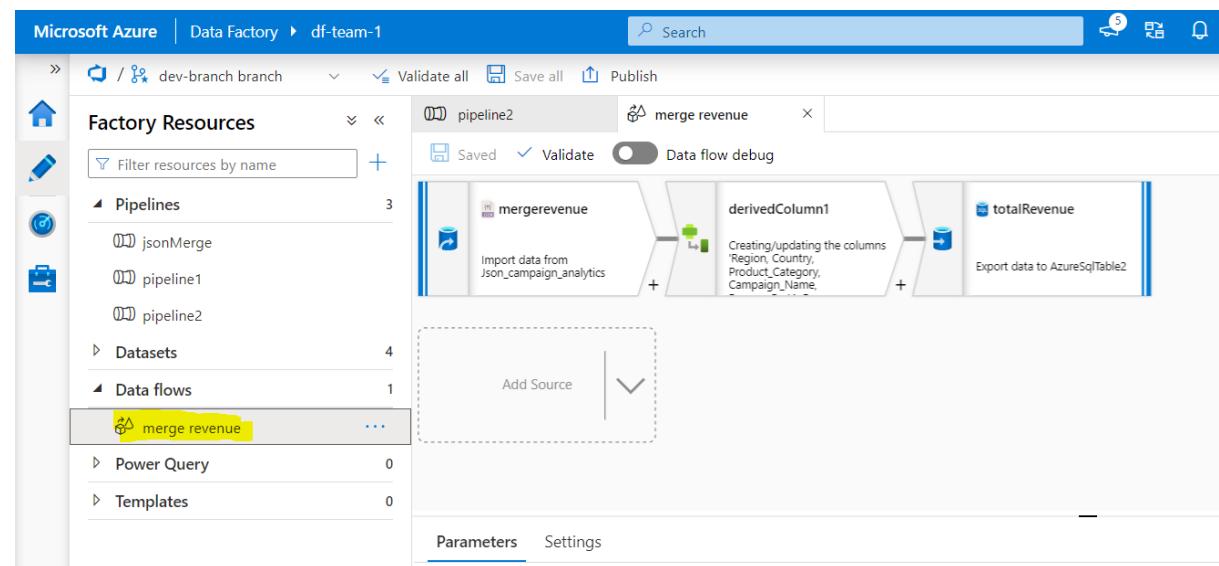
This is the original form of the json file, and as we can see there are some files which need to be transformed and merged.



The screenshot shows the 'Preview data' window in the Azure Data Factory interface. It displays a JSON object representing campaign analytics data. The JSON structure is as follows:

```
[  
  {  
    "Region": "Europe",  
    "Country": "Germany",  
    "Product_Category": "Apparel and Footwear",  
    "Campaign_Name": "Fun with Colors",  
    "RevenuePart1": "$14\\\",  
    "Revenue": 865,  
    "RevenueTargetPart1": "$15\\\",  
    "Revenue_Target": 960,  
    "City": "",  
    "State": ""  
  },  
  {  
    "Region": "Far West",  
    "Country": "US",  
    "Product_Category": "Books",  
    "Campaign_Name": "EnjoyTheMoment; BeUnique; TailoredForYou",  
    "RevenuePart1": "$14\\\",  
    "Revenue": 992,  
    "RevenueTargetPart1": "$15\\\",  
    "Revenue_Target": 699,  
    "City": "San Diego",  
    "State": "California"  
  },  
  {  
    "Region": "Europe",  
    "Country": "Germany",  
    "Product_Category": "Electronics",  
    "Campaign_Name": "TechLovers",  
    "RevenuePart1": "$16\\\",  
    "Revenue": 1050,  
    "RevenueTargetPart1": "$17\\\",  
    "Revenue_Target": 1100,  
    "City": "Berlin",  
    "State": "Berlin"  
  }]
```

For this we have created this data flow:



Merge revenue:

Source setting:

The diagram illustrates a data flow process. It starts with a 'mergeRevenue' source (represented by a blue cylinder icon) containing 10 total columns. This is followed by a 'derivedColumn1' step (represented by a grey box with a green plus sign icon), which creates/updating columns like 'Region', 'Country', 'Product_Category', and 'Campaign_Name'. Finally, the data is exported to an 'AzureSqlTable2' (represented by a blue cylinder icon). Below the diagram, the 'Source settings' tab is selected, showing the following configuration:

- Output stream name ***: mergerevenue
- Source type ***: Dataset (selected)
- Dataset ***: Json_campaign_analytics
- Options**:
 - Allow schema drift
 - Infer drifted column types
 - Validate schema
- Sampling ***:
 - Enable
 - Disable

We should also do this configuration to read the array of documents:

The diagram shows the same data flow structure as the previous screenshot. Below the diagram, the 'Source options' tab is selected, with the 'capture' section expanded. The configuration includes:

- Column to store file name**: (empty input field)
- After completion ***:
 - No action
 - Delete source files
 - Move
- Start time (UTC)**: (empty input field)
- End time (UTC)**: (empty input field)
- Filter by last modified**: (empty input field)
- JSON settings**:
 - Document form**:
 - Single document
 - Document per line
 - Array of documents
 - Unquoted column names**:
 - Has comments**:
 - Single quoted**:
 - Backslash escaped**:

DerivedColumns configuration:

The screenshot shows the Azure Data Factory pipeline editor. A pipeline named "pipeline2" is open, with a step titled "merge revenue". The "merge revenue" step contains a "derivedColumn1" component. This component has an "Incoming stream" of "mergerevenue" and an "Output stream name" of "derivedColumn1". The "derivedColumn1" component is connected to a "totalRevenue" component, which then connects to an "Export data to AzureSqlTable2" step. Below the pipeline, the "Derived column's settings" tab is selected, showing the configuration for "derivedColumn1". The "Columns" section lists two columns: "RevenueTotal" and "RevenueTargetTotal". The expression for "RevenueTotal" is `toDecimal(concat(toString(regexReplace(RevenuePart1, '[^0-9]+', '')),toString(Revenue)))`. The expression for "RevenueTargetTotal" is `toDecimal(concat(toString(regexReplace(RevenueTargetPart1, '[^0-9]+', '')),toString(Revenue_Target)))`.

The function for RevenueTotal:

```
toDecimal(concat(toString(regexReplace(RevenuePart1,
'[0-9]+', '')),toString(Revenue)))
```

The screenshot shows the "Visual expression builder" interface. On the left, there is a sidebar with icons for Home, Data Flow, and Pipelines. Under "Data Flow", there is a "derivedColumn1" node. In the main area, under "Derived Columns", there is a "Create new" button. To its right, there is a "Column name" field set to "RevenueTotal" and an "Expression" field containing the expression `toDecimal(concat(toString(regexReplace(RevenuePart1, '[0-9]+', '')),toString(Revenue)))`.

The function for RevenueTargetTotal:

```
toDecimal(concat(toString(regexReplace(RevenueTargetPart1,
'[0-9]+', '')),toString(Revenue_Target)))
```

Visual expression builder

derivedColumn1

Derived Columns

+ Create new

RevenueTotal
RevenueTargetTotal

Column name *
RevenueTargetTotal

Expression

```
toDecimal(concat(toString(regexReplace(RevenueTargetPart1, '[^0-9]+', '')),toString(Revenue_Target)))
```

+ - * / || && ! ^ == === <=> !=

TotalRevenue:

pipeline2 merge revenue

Saved Validate Data flow debug

merge revenue
Import data from Json_campaign_analytics

derivedColumn1
Creating/updating the columns 'Region', 'Country', 'Product_Category', 'Campaign_Name',

totalRevenue
Columns: 8 total

Sink Settings Mapping Optimize Inspect Data preview

Output stream name * totalRevenue Learn more

Incoming stream * derivedColumn1

Sink type * Dataset

Dataset * AzureSqlTable2 Test connection Open New

Options Allow schema drift Validate schema

Mapping: Here we removed some columns and added the derived columns.

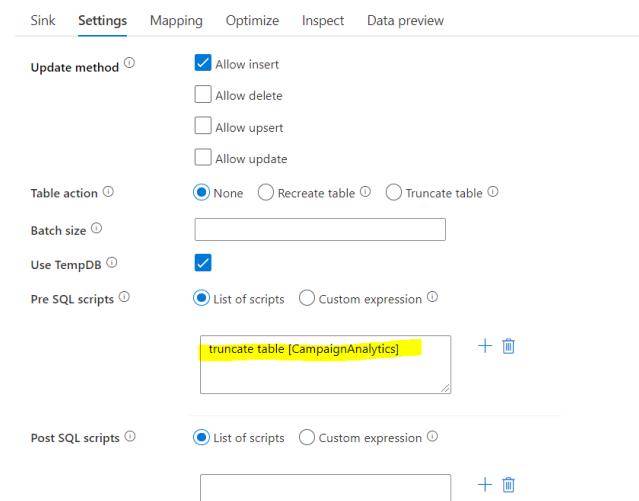
Sink Settings **Mapping** Optimize Inspect Data preview

Options Skip duplicate input columns Skip duplicate output columns

Auto mapping Reset Add mapping Delete Output format

Input columns	Output columns
abc Region	abc Region
abc Country	abc Country
abc Product_Category	abc ProductCategory
abc Campaign_Name	abc CampaignName
e ^x RevenueTotal	e ^x Revenue
e ^x RevenueTargetTotal	e ^x RevenueTarget
abc City	abc City
abc State	abc State

In settings we have to add this:



In SQL server we will see the data properly loaded:

```
SQLQuery2.sql - rg-... (sqladminuser (79))  X  SQLQuery1.sql - rg-... (sqladminuser (53))
1  ***** Script for SelectTopNRows command from SSMS *****/
2  SELECT TOP (1000) [Region]
3    ,[Country]
4    ,[ProductCategory]
5    ,[CampaignName]
6    ,[Revenue]
7    ,[RevenueTarget]
8    ,[City]
9    ,[State]
10   FROM [dbo].[CampaignAnalytics]
```

100 %

Results Messages

Region	Country	ProductCategory	CampaignName	Revenue	RevenueTarget	City	State
Europe	Germany	Apparel and Footwear	Fun with Colors	14865.00	15960.00		
Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	14992.00	15699.00	San Diego	California
Europe	Germany	Apparel and Footwear	Fall into Winter	5117.00	8713.00		
Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	9935.00	15232.00	San Diego	California
Europe	France	Apparel and Footwear	Enjoy the Moment	13221.00	8584.00		
Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	15119.00	17269.00	San Diego	California
Europe	UK	Lighting	Fall into Winter	5117.00	9305.00		
Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	15740.00	7685.00	San Diego	California
South America	Mexico	Electronics	Be Unique	16240.00	16038.00		
Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	14778.00	13122.00	San Diego	California
North & Central America	USA	Décor	Spring into Summer	6689.00	13088.00		
Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	10296.00	7313.00	San Diego	California
South America	Mexico	Apparel and Footwear	Enjoy the Moment	1398.00	5663.00		
Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	14605.00	18971.00	San Diego	California
South America	Brazil	Décor	Fun with Colors	15142.00	7147.00		
Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	14328.00	15577.00	San Diego	California
South America	Mexico	Exercise	Spring into Summer	17637.00	6876.00		
Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	11247.00	10339.00	San Diego	California
South America	Mexico	Décor	Be Unique	8284.00	9840.00		
Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	12409.00	10914.00	San Diego	California
South America	Mexico	Apparel and Footwear	Tailored for You	5766.00	17620.00		
Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	5418.00	13970.00	San Diego	California
Europe	Spain	Furniture	Enjoy the Moment	12488.00	18299.00		
Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	15387.00	6961.00	San Diego	California

Create a table in SQL server for the parquet file:

..
...
..

In Azure create a linked service to connect to the files we want to load.

Set properties

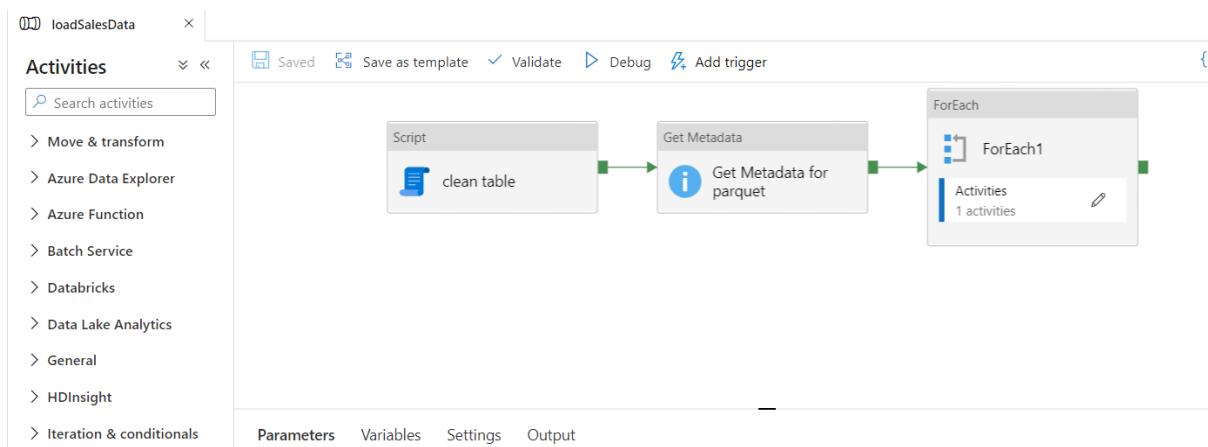
Name

Linked service *
 

File path
 / /  

Import schema
 From connection/store From sample file None

Now let's create a pipeline for loading the parquet file:



For the Items add this dynamic activity:

Add dynamic content

```
@activity('Get Metadata for parquet').output.childItems
```

The configuration for the activities in the pipeline for parquet files is as below.

General:

General Settings User properties

Name * [Learn more](#)

Description

Timeout ⓘ

Retry ⓘ

Retry interval (sec) ⓘ

Secure output ⓘ

Secure input ⓘ

Setting:

General **Settings** User properties

Dataset * [Open](#) [New](#) [Learn more](#)

Field list * [New](#) | [Delete](#)

Argument

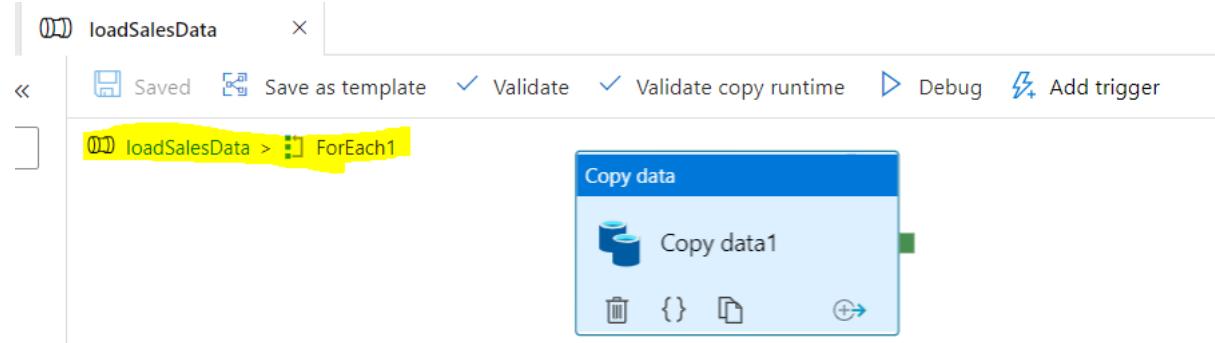
Child items

Start time (UTC)

End time (UTC)

Filter by last modified ⓘ

For Each loop configuration:



General:

General Source Sink **Mapping** Settings User properties

Name *	<input type="text" value="Copy data1"/> Learn more
Description	<input type="text"/>
Timeout ⓘ	<input type="text" value="7.00:00:00"/>
Retry ⓘ	<input type="text" value="0"/>
Retry interval (sec) ⓘ	<input type="text" value="30"/>
Secure output ⓘ	<input type="checkbox"/>
Secure input ⓘ	<input type="checkbox"/>

Source:

General **Source** Sink Mapping Settings User properties

Source dataset *	<input type="text" value="Parquet1"/> Open New Preview data Learn more
File path type	<input type="radio"/> File path in dataset <input checked="" type="radio"/> Wildcard file path <input type="radio"/> List of files ⓘ
Wildcard paths	<input type="text" value="wwi-02 / */**/*"/> / <input type="text" value="*.parquet"/>
Filter by last modified ⓘ	<input type="text"/> Start time (UTC) <input type="text"/> End time (UTC)
Recursively ⓘ	<input checked="" type="checkbox"/>
Enable partition discovery	<input type="radio"/> <input type="checkbox"/>
Max concurrent connections	<input type="radio"/> <input type="text"/>
Additional columns ⓘ	+ New

Sink:

General Source **Sink** Mapping Settings User properties

Sink dataset *	<input type="text" value="AzureSqlTable3"/> Open New Learn more
Write behavior	<input checked="" type="radio"/> Insert <input type="radio"/> Upsert <input type="radio"/> Stored procedure
Bulk insert table lock ⓘ	<input type="radio"/> Yes <input checked="" type="radio"/> No
Table option	<input checked="" type="radio"/> None <input type="radio"/> Auto create table
Pre-copy script ⓘ	<input type="text"/>
Write batch timeout	<input type="text"/>
Write batch size	<input type="text"/>
Max concurrent connections	<input type="radio"/> <input type="text"/>

Mapping:

General Source Sink **Mapping** Settings User properties ^

> Type conversion settings

Import schemas

Source	Type	Destination	Type
TransactionId	UTF8	TransactionId	uniqueidentifier
CustomerId	INT32	CustomerId	int
ProductId	INT_16	ProductId	smallint
Quantity	INT_8	Quantity	tinyint
Price	DECIMAL	Price	decimal
Precision: 38 Scale: 18		Precision: 9 Scale: 2	

Settings:

General Source Sink Mapping **Settings** User properties

You will be charged # of used DIUs * copy duration * \$0.25/DIU-hour. Local currency and separate discounting

Data integration unit	<input type="button" value="Auto"/>
	<input type="checkbox"/> Edit
Degree of copy parallelism	<input type="text"/>
	<input checked="" type="checkbox"/> Edit
Data consistency verification	<input type="checkbox"/>
Fault tolerance	<input type="button" value=""/>
Enable logging	<input type="checkbox"/>
Enable staging	<input type="checkbox"/>

Now we will run and debug the pipeline.

loadSalesData

Runs

Refresh Update pipeline

This is a recent debug run. The local pipeline configuration is shown.

Activity runs

Pipeline run ID: 36c6cd23-fed0-424d-ba1c-0135c34f74f6

All status ▾

Showing 1 - 3 of 3 items

Activity name	Activity type	Run start ↑	Duration	Status	Error	Log	Integration runtime	User
Copy data1	Copy data	4/20/22, 9:16:53 AM	00:14:43	In progress				
ForEach1	ForEach	4/20/22, 9:16:52 AM	00:14:43	In progress				
Get Metadata for parquet	Get Metadata	4/20/22, 9:16:47 AM	00:00:04	Succeeded			AutoResolveIntegrationRuntime (North Europe)	

The execution of the pipeline for loading parquet file takes time and the progress till now is as below:

Details Refresh

Learn more on copy performance details from here.

Activity run id: c5f57980-4786-4a93-a0d4-65a95f2d38a0

Azure Data Lake Storage Gen2 → In progress → Azure SQL Database

Azure Data Lake Storage Gen2 Region: North Europe **Azure SQL Database** Region: West Europe

Data read: 1.102 GB Data written: 1.055 GB

Files read: 22 Rows written: 12,444,570

Rows read: 15,887,969 Peak connections: 2

Peak connections: 6

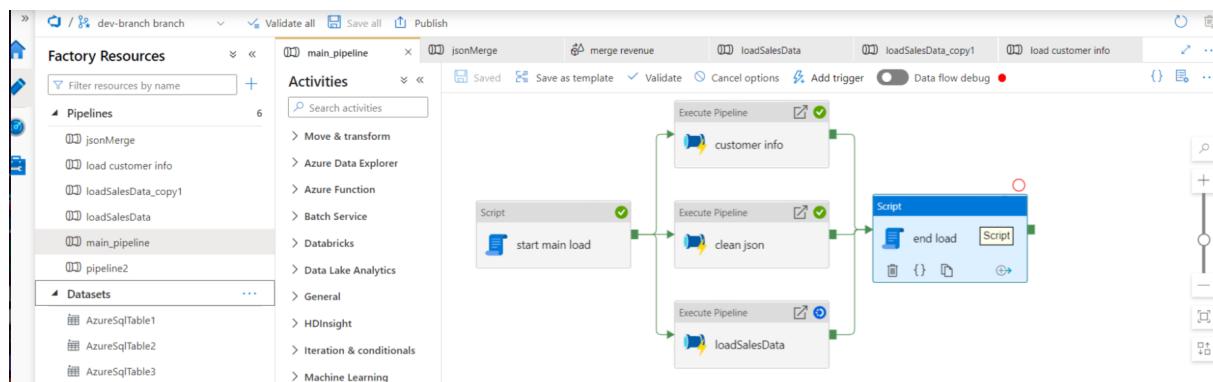
Copy duration: 01:04:16

Throughput: 299.652 KB/s

✓ Azure Data Lake Storage Gen2 → Azure SQL Database

Details	Working duration	Total duration
Queue	00:00:16	00:00:02

We have also added a main pipeline that starts all the pipelines which will run in parallel and then it records in a table the starting and the ending time for the tasks.



Bonus task:

- (Bonus points) Connect Streamlit to data source (3pt)
- (Bonus points) Present Dashboard with the data (3pt)

For this we will need to install the Streamlit library. For installing this we use the command:

```
pip install streamlit
```

We might also need these two libraries:

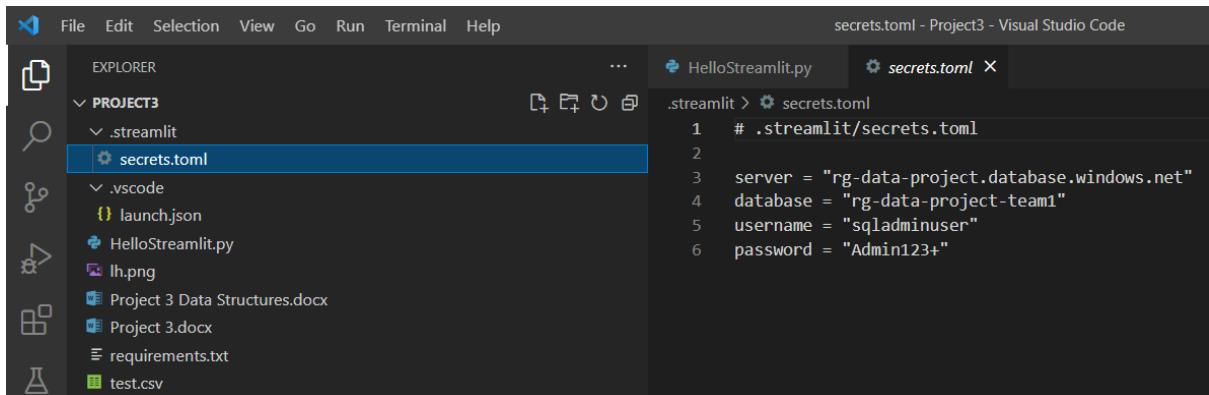
```
pandas
```

```
plotly
```

For this project we have decided to visualise the data related to **Campaign Analytics**.

For connecting streamlit to SQL Server we have done the following:

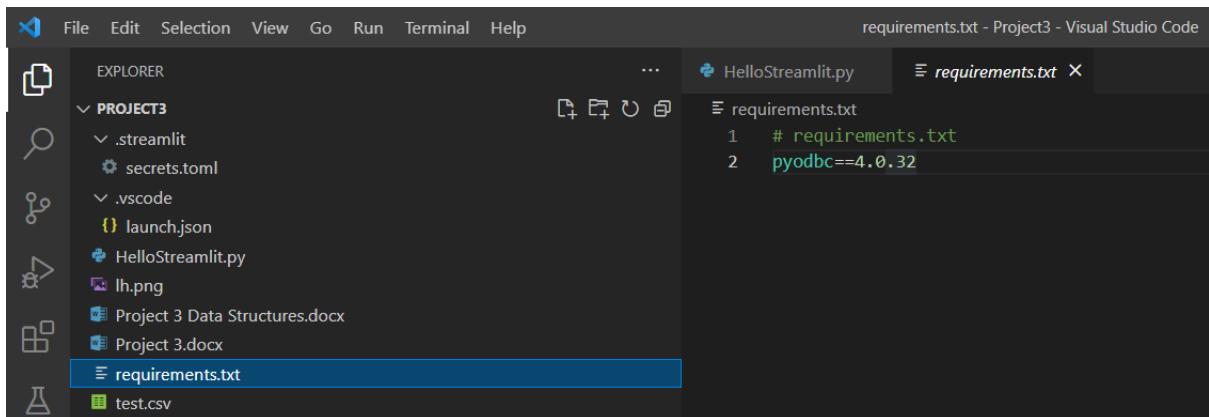
1. Added the credentials to the local apps secret. Our local Streamlit app will read secrets from a file `.streamlit/secrets.toml` in our app's root directory.



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists files and folders: .PROJECT3, .streamlit (which contains secrets.toml), .vscode (with launch.json), and other files like Ih.png, Project 3 Data Structures.docx, Project 3.docx, requirements.txt, and test.csv. The main editor window on the right displays the contents of secrets.toml:

```
.streamlit > secrets.toml
1  # .streamlit/secrets.toml
2
3  server = "rg-data-project.database.windows.net"
4  database = "rg-data-project-team1"
5  username = "sqladminuser"
6  password = "Admin123+"
```

2. Added pyodbc to our requirements file



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists files and folders: .PROJECT3, .streamlit (with secrets.toml), .vscode (with launch.json), and other files like Ih.png, Project 3 Data Structures.docx, Project 3.docx, requirements.txt, and test.csv. The main editor window on the right displays the contents of requirements.txt:

```
requirements.txt
1  # requirements.txt
2  pyodbc==4.0.32
```

3. Modified our code below to establish the connection.

```
def init_connection():

    return pyodbc.connect("DRIVER={ODBC Driver 17 for SQL
Server};SERVER=" + st.secrets["server"] + ";DATABASE=" +
st.secrets["database"] + ";UID="

    + st.secrets["username"]

    + ";PWD="

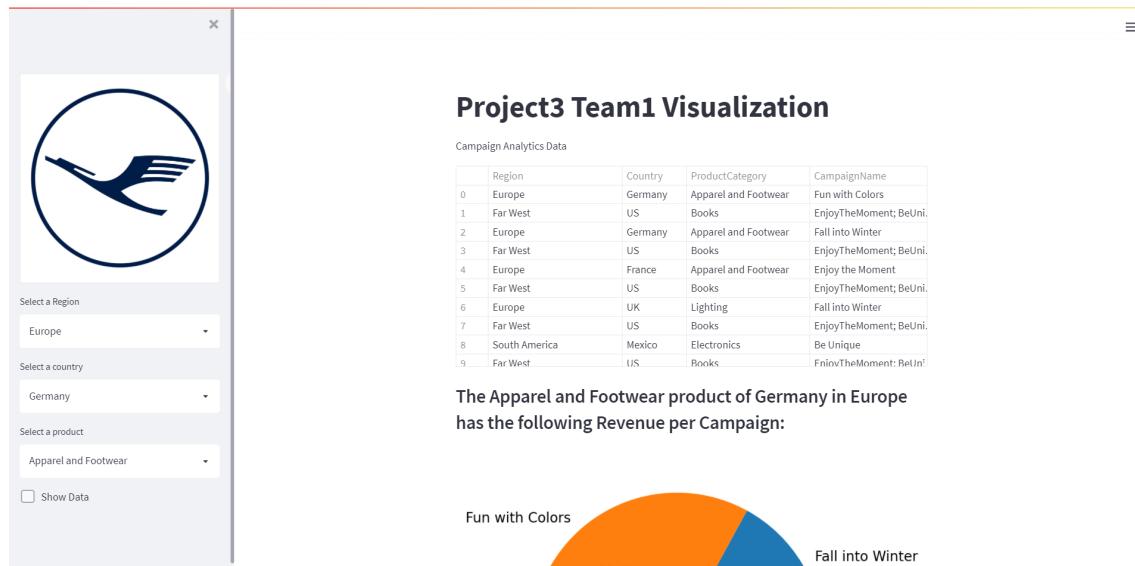
    + st.secrets["password"]

)

conn = init_connection()

chunk_test = pd.read_sql('SELECT * from CampaignAnalytics', conn)
```

The output:



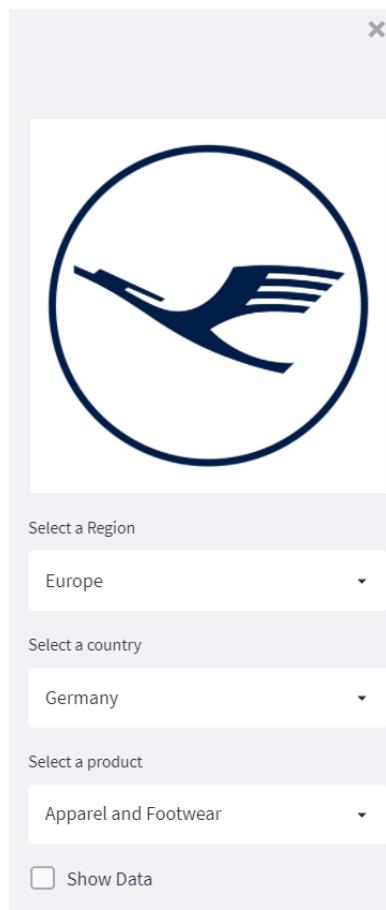
In the beginning of the page, we have displayed the dataset, which we transformed into Azure Data Factory.

Project3 Team1 Visualization

Campaign Analytics Data

	Region	Country	ProductCategory	CampaignName
20	South America	Mexico	Apparel and Footwear	Tailored for You
21	Far West	US	Books	EnjoyTheMoment; BeUni.
22	Europe	Spain	Furniture	Enjoy the Moment
23	Far West	US	Books	EnjoyTheMoment; BeUni.
24	Asia Pacific	Japan	Décor	Tailored for You
25	Far West	US	Books	EnjoyTheMoment; BeUni.
26	Europe	Germany	Electronics	Fall into Winter
27	Far West	US	Books	EnjoyTheMoment; BeUni.
28	Europe	Italy	Lighting	Get Sporty
29	Far West	US	Books	EnjoyTheMoment; BeUni.

Below is given the side menu, which helps you to filter the data according to your preferences. You are able to select the region, country and the product.

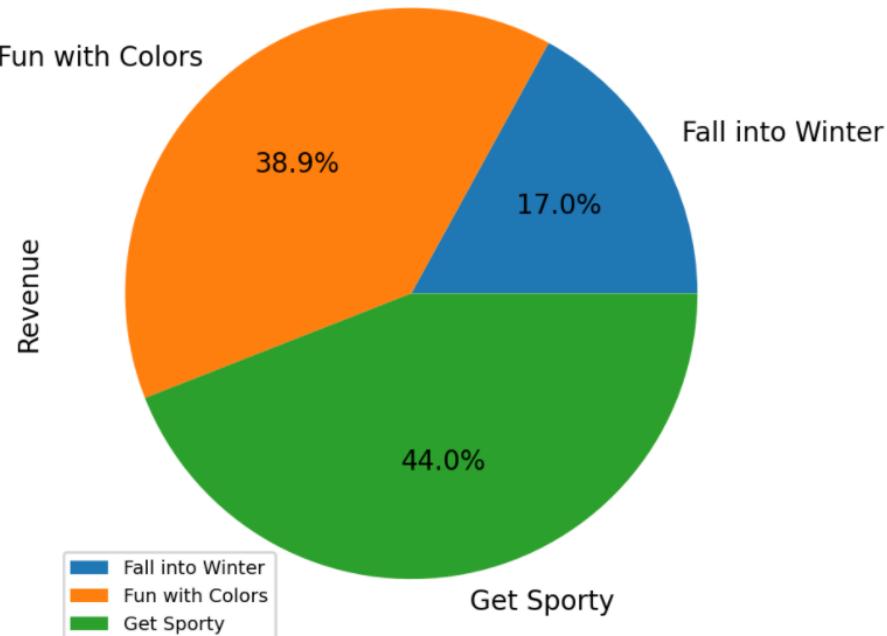


Clicking on the **Show Data** option, the data will be displayed in a tabular format for the selected filters.

The Apparel and Footwear product of Germany in Europe has the following Revenue per Campaign:

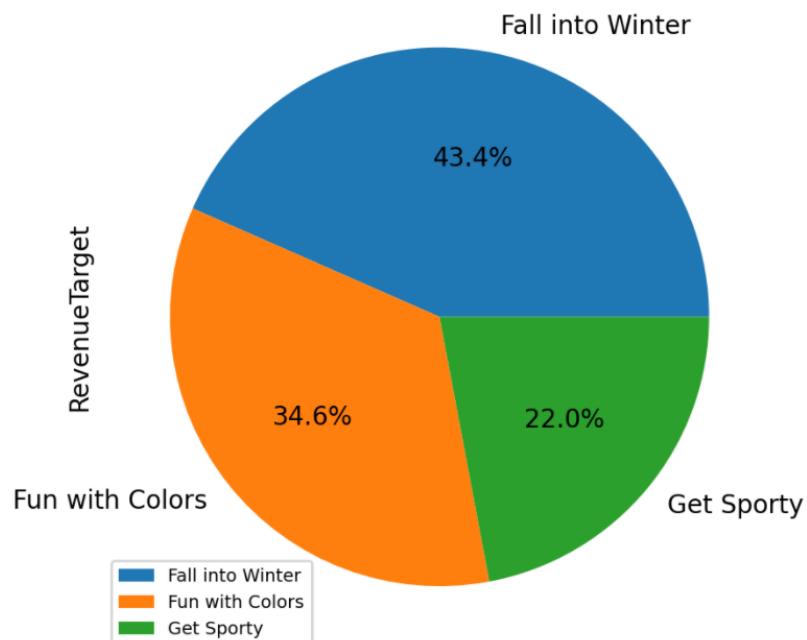
	Region	Country	ProductCategory	CampaignName	Revenue	Reve
0	Europe	Germany	Apparel and Footwear	Fun with Colors	14,865.0000	15
2	Europe	Germany	Apparel and Footwear	Fall into Winter	5,117.0000	8
38	Europe	Germany	Apparel and Footwear	Get Sporty	16,810.0000	10
57	Europe	Germany	Apparel and Footwear	Fall into Winter	1,383.0000	11
125	Europe	Germany	Apparel and Footwear	Fun with Colors	14,865.0000	15
127	Europe	Germany	Apparel and Footwear	Fall into Winter	5,117.0000	8
163	Europe	Germany	Apparel and Footwear	Get Sporty	16,810.0000	10
180	Europe	Germany	Apparel and Footwear	Fall into Winter	1,383.0000	11

Below we will see the data displayed with a pie chart.

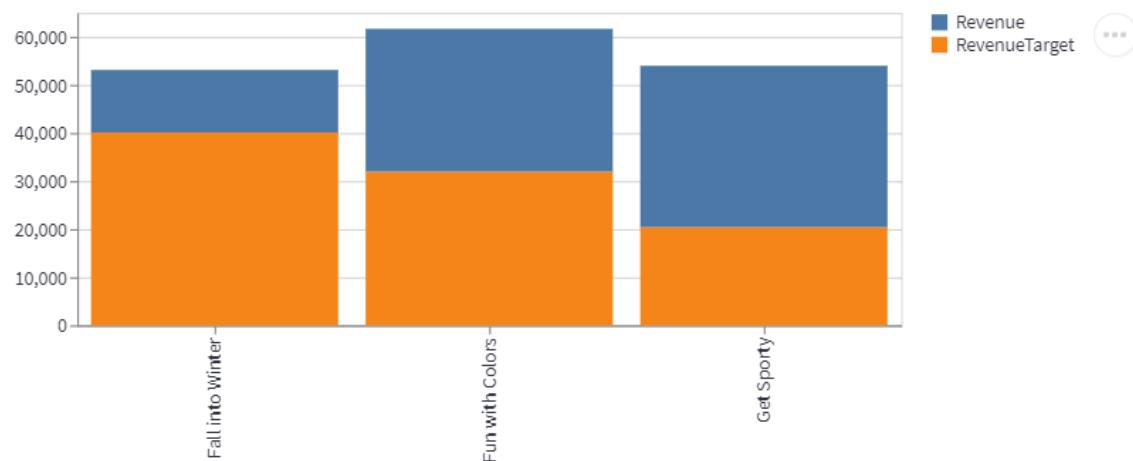


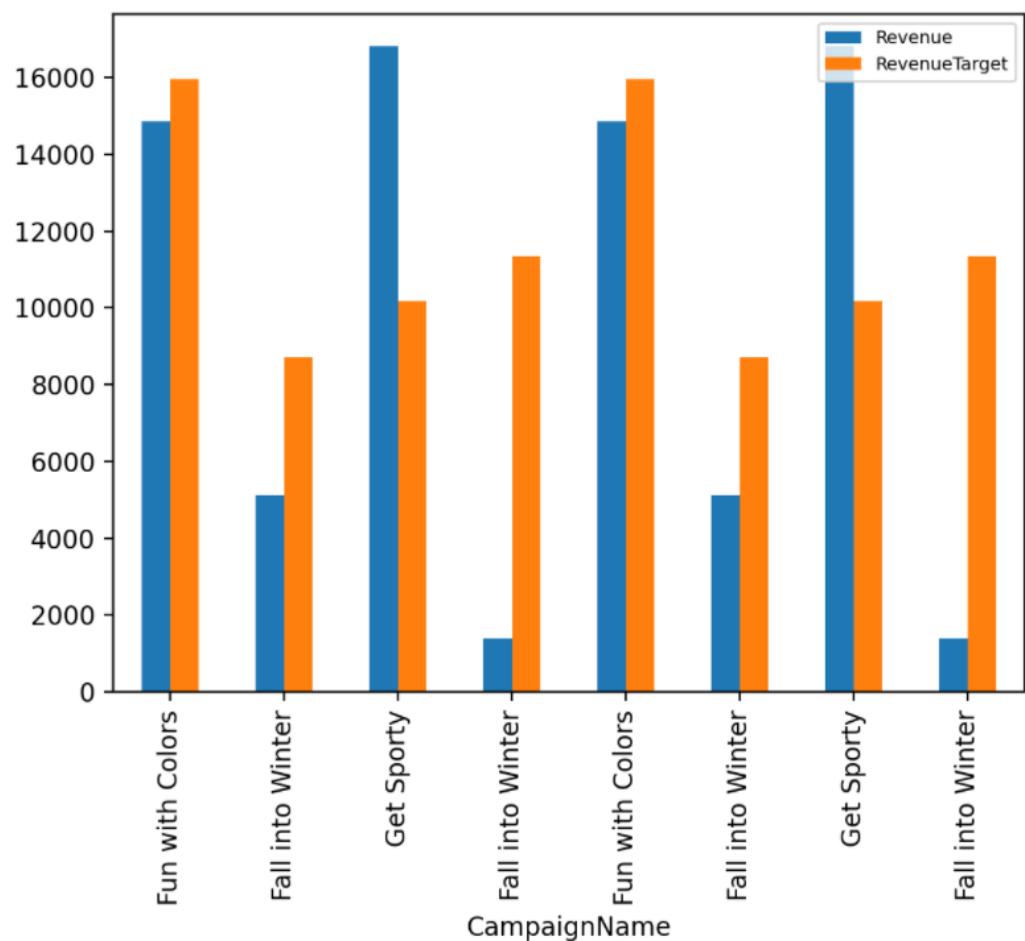
Below is displayed the Revenue Target per campaign based on the filters.

The Apparel and Footwear product of Germany in Europe has the following RevenueTarget per Campaign:



Another way we used to display the data:



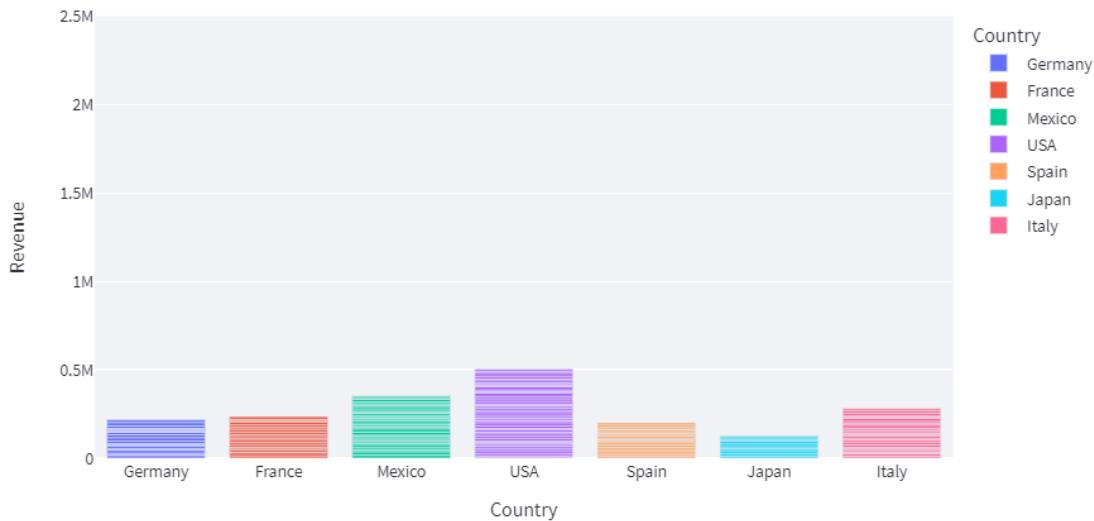


We have displayed the revenue based on the country. This is a multiselect option which allows us to select many countries and display the revenue for the selected countries.

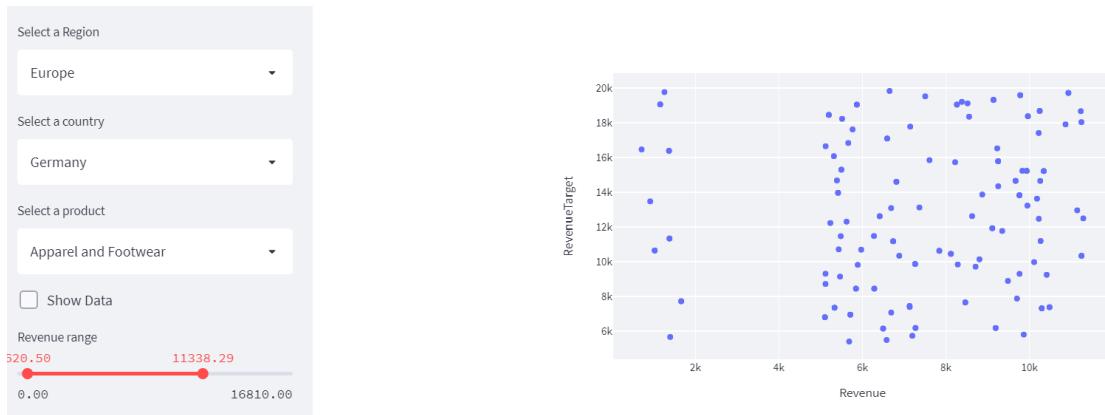
MultiSelect Option - Revenue based on Country

Which country you would like to see?

Germany × France × USA × Spain × Japan × Italy ×
Mexico ×



We also have added one more scatter plot for the revenue. On the left side of the page we have the filters and we can select the range for the revenue and display the results in a scatterplot.

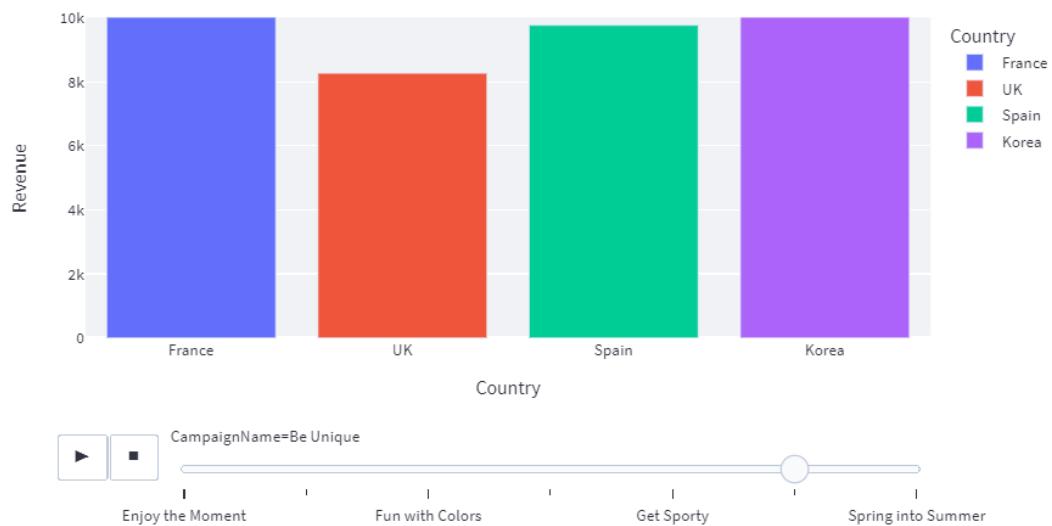


An extra thing we tried to experiment for this project is the animation. As you can see below the user is able to select the country and we have made it to group by campaign Name. Based on the country and the campaign name the user will see how the revenue changes.

> Animation test section

Which country you would like to see?

UK X France X Spain X Korea X



If the user slides through the other campaign this is how the graph will change:

Animation test section

Which country you would like to see?

UK X France X Spain X Korea X

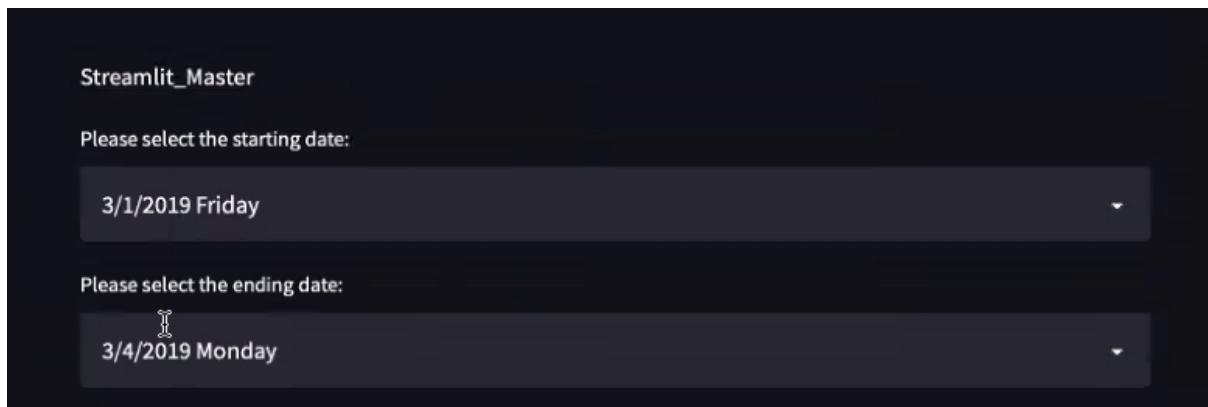
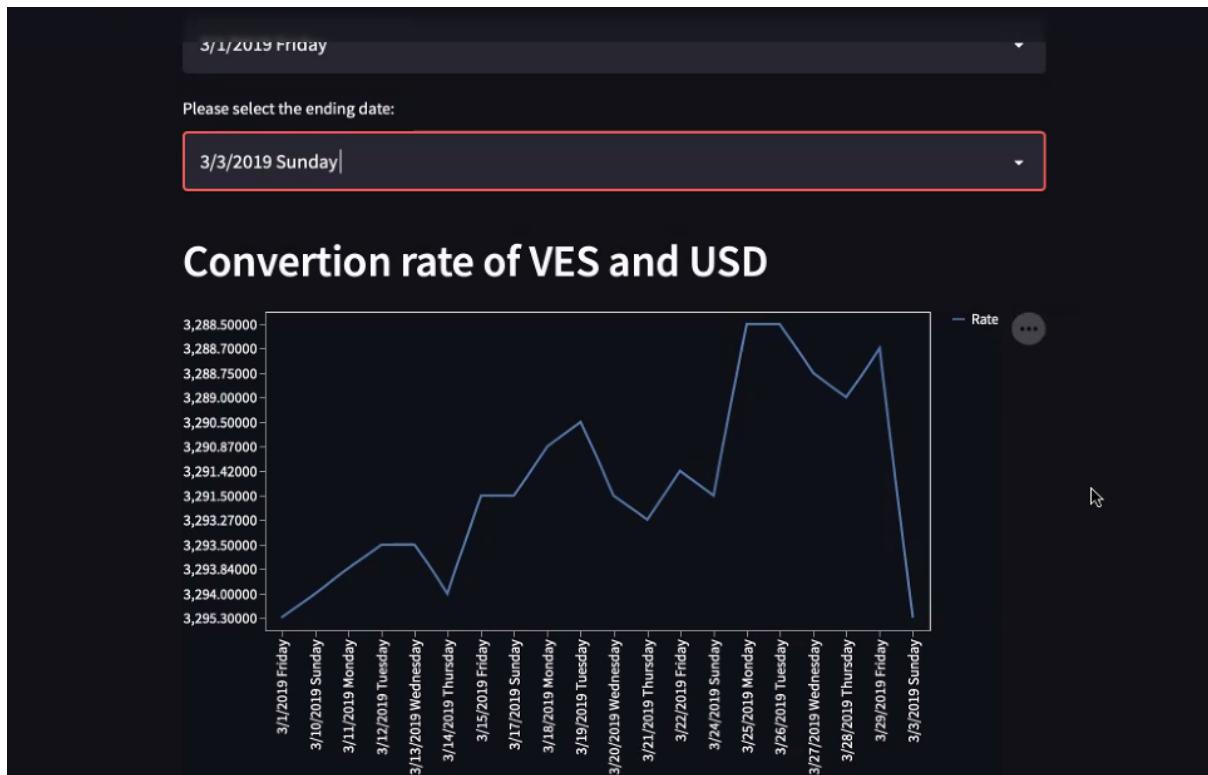


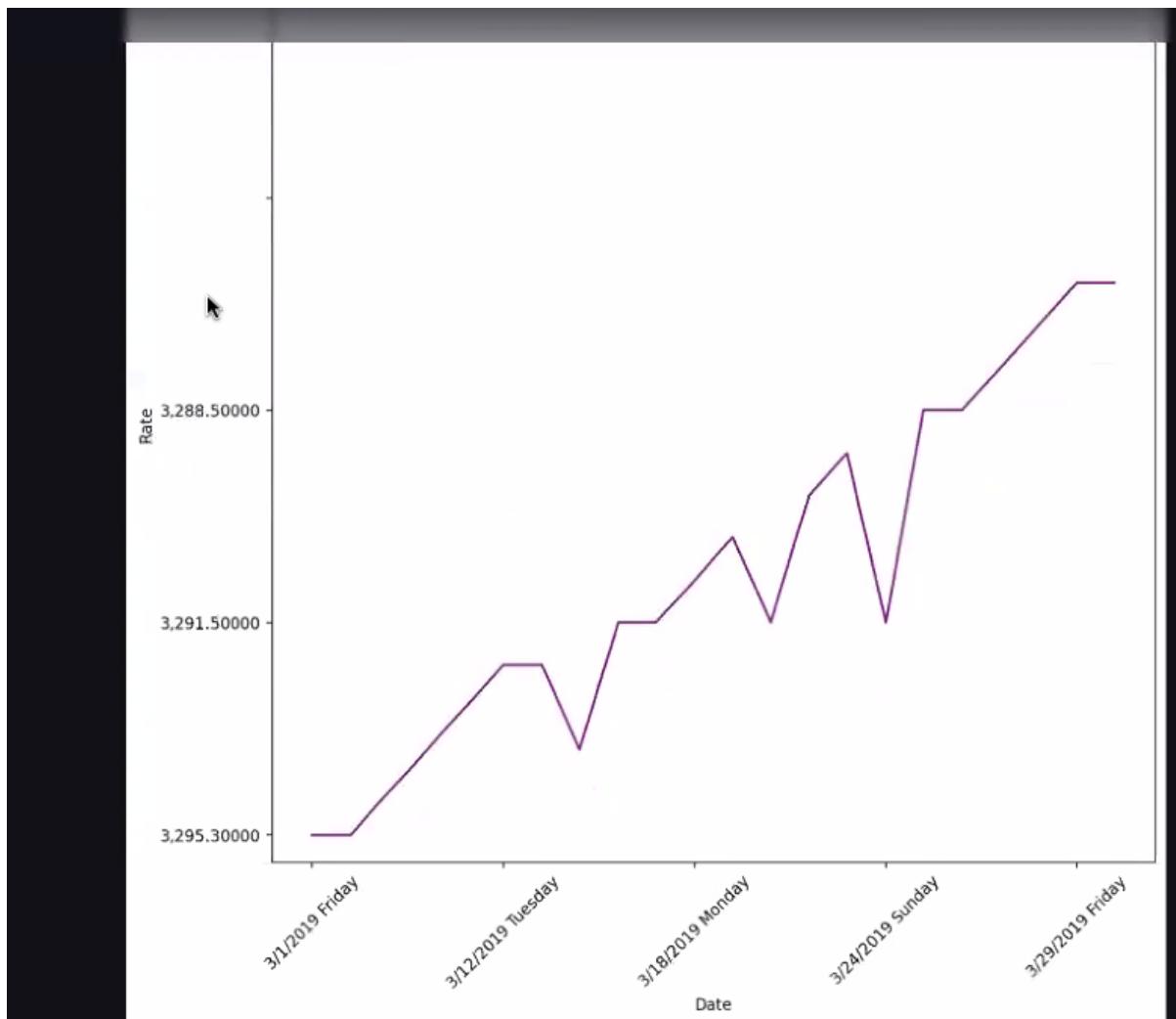
Extra bonus tasks(requirements taken in the last day of the project) :

Task 1

Load the file ves-usd.csv from the ML-datasets repository:

<https://github.com/matzim95/ML-datasets> and plot a linear graph to show the dependence between bolivar and US dollar.



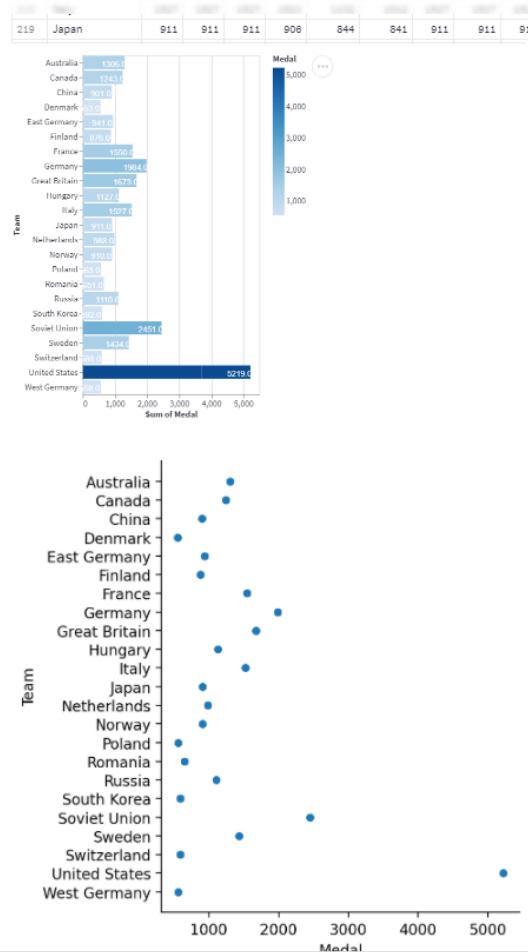


Task 3

Analyze the data from Olympic Games (look into the file `olympic.csv` <https://github.com/matzim95/ML-datasets>) and make a bar chart showing the number of medals the country has won. Focus only on these countries that got more than 500 medals in total. Use the colors to indicate whether the medal was gold, silver or bronze.

Task 4

Using the set from the previous task, create a scatter plot showing the relationship between the height, the weight, and the number of medals that were won by players at the Olympics. Focus only on these players that got more than 5 medals in total.



Task 5:

Load a co2 table from the repository and display the scatter plot showing average change of CO2 emission over time using seaborn. Think about odd values that could be mistakes and how we can filter them out. Ignore Trend and Interpolated columns.

Task 6

Make a boxplot, a violinplot and swarmplot plots for Age, Height and Weight of players from one kind of sports from the Olympians dataset using the seaborn library.

The Olympic Dataset

	ID	Name	Sex	Age	Height	Weight	Team
0	1	A Dijiang	M	24.0000	180.0000	80.0000	China
1	2	A Lamusi	M	23.0000	170.0000	60.0000	China
2	3	Gunnar Nielsen Aaby	M	24.0000	<NA>	<NA>	Denma
3	4	Edgar Lindenau Aabye	M	34.0000	<NA>	<NA>	Denma
4	5	Christine Jacoba Aafink	F	21.0000	185.0000	82.0000	Netherl
5	5	Christine Jacoba Aafink	F	21.0000	185.0000	82.0000	Netherl
6	5	Christine Jacoba Aafink	F	25.0000	185.0000	82.0000	Netherl
7	5	Christine Jacoba Aafink	F	25.0000	185.0000	82.0000	Netherl
8	5	Christine Jacoba Aafink	F	27.0000	185.0000	82.0000	Netherl
9	5	Christine Jacoba Aafink	F	27.0000	185.0000	82.0000	Netherl

Boxplot for a sport based on Age, Height, Weight

Please select the sport:

▼

	Year	Season	City	Sport	Event	Medal
0	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	<NA>
167	2008	Summer	Beijing	Basketball	Basketball Women's Bask...	<NA>
250	1952	Summer	Helsinki	Basketball	Basketball Men's Basketball	<NA>
264	2000	Summer	Sydney	Basketball	Basketball Men's Basketball	<NA>
346	1972	Summer	Munich	Basketball	Basketball Men's Basketball	<NA>
359	1984	Summer	Los Angeles	Basketball	Basketball Men's Basketball	<NA>
360	1988	Summer	Seoul	Basketball	Basketball Men's Basketball	<NA>
363	1972	Summer	Munich	Basketball	Basketball Men's Basketball	<NA>
364	1976	Summer	Montreal	Basketball	Basketball Men's Basketball	<NA>
490	1988	Summer	Seoul	Basketball	Basketball Men's Basketball	<NA>

