Final Project Report

Big Data Analysis and Cloud Computing (CS714)

Project Title

Telecom Customer Profiling for Personalized Marketing

Submitted By: A H M Gulam Azam

Student ID: 200487553

Submission Date- August 17, 2023

Table of Contents

1. Introduction

It's not enough to merely sell a product or service in today's fast-paced digital world. Businesses must use individualized marketing if they want to successfully win the hearts (and wallets) of today's consumers. Each consumer may have a customized experience, which boosts customer satisfaction, revenue, and loyalty. One of the most important advantages of personalized marketing is that it raises the likelihood of client interaction. Businesses that personalize their marketing messages to specific clients are more likely to notice and engage with the material. This can lead to improved conversions and income.

Another significant benefit of personalized marketing is that it allows firms to develop relationships with their clients. Businesses can develop a sense of connection with their clients by providing a personalized experience. This connection results in improved client loyalty and repeat business.

But how exactly can businesses implement personalized marketing? There are many ways businesses can tailor their marketing messages and offers to specific customers. For instance, one is segmentation: dividing customer base into segments based on common characteristics like demographics, usage behavior, interests, and engagement. This helps to create targeted marketing campaigns for specific groups. Another way is to build customer profiles: developing detailed customer profiles for each segment. Include information like demographics, preferences, browsing history, purchase frequency, and more. The more comprehensive the profile, the better you can tailor your marketing efforts.

Telecom operators became among the richest companies in terms of data volume. The bandwidths are more and more saturated, and the data generated is growing exponentially every minute. Large amounts of data need greater storage space and much more computational power to analyze. The Challenge is to understand how they can leverage this data to lower operating costs, deliver a personalized customer experience, reduce the churn rate, and develop new sources of revenue. The answer to this question turns out not to be so obvious: Cloud Computing.

Big data analysis by using cloud computing makes it flexible, scalable, and secure to identify target groups in real-time. Analyzing data is really important for so many marketing areas, such as - research and development(R&D) before launch, making a profit, being the leader in a competitive market, reaching targeted profit, ensuring profit margin, and even sometimes just being in the market. For Telecom companies, to identify effective solutions, there is no alternative to analyze the transactional data pattern of their customers, it is one of the most powerful entities of a customer.

There are many cloud-based service providers are already established and performing big roles by providing multiple services in the market, some are given below:

1. Google Cloud Services
2. Microsoft Azure
3. Amazon Web Services (AWS)
4. IBM Cloud

This report will describe the project overview- which means, the process of using customers usage data of a Telecom Company by using **Azure Cloud Service**- and oracle database engine. Also, the process is identifying usage pattern of an individuals directly from the hosted databases in real time. In a nutshell, this real-time data process can help a telecom company to identify the nature of a targeted customer to push more relatable and suitable offers or promotions to each customer from cloud storage.

2. Statement of Purpose

Activating telecom customers' core usage data to drive targeted marketing. The purpose is to Create a framework to manage large volumes of data in a highly scalable ecosystem. Which will reduce campaign managers dependency on internal IT infrastructure. After completing the process anyone with proper access can fetch data from Azure Cloud by accessing web platform any time. It will ease the hurdle to get appropriate segmented data from the internally managed data ware house.

Another aspect is to find cost efficient way of data archiving and analyzing customer data for more accurate personalized marketing. Telecom companies tend to discard data frequently due to the space constraint of their data centers. To accommodate this large volume of data they need more data centers which needs time and money to built.

This project is focused on archiving data in Azure Cloud service directly from company's oracle data ware house. Using Azure fast computing and data processing the data will be transformed and customer profiling will be done. The processed data will be hosted in Azure SQL database

3. Methodological Approach

3.1. Data Type

In this project, synthetic data of a Telcom- customer's usage are being created using real-life data and synthetic data creation algorithm. The Data pattern is Structured Data, that has been organized into a formatted pattern, usually, this dataset's elements are addressable for more effective processing and analysis purposes. The reasons behind using Structured Dataset in this project are given below-

1. This is highly specific – as it is already decided, which types of data are going to be used in this project, this is a perfect option
2. This is user-defined type data- that contains one or more named attributes; this project column includes- MSISDN, prp_pst_tag, Activation Status, Voice Revenue, Data Revenue, Voice Out Min, MB_USAGE and 13 more.
3. This can be stored in a predefined format- this project considers fixed fields and columns.

3.2. Data Modeling

In this project, telecom customers' usage data is considered for profiling to identify which offers are best suited according to their usage pattern. Here is the sample of a data instance. This way, 190,000 number of individual rows are created using synthetic data created. At first I have created 200K rows but some data needed to be discarded due to data type mismatch.

Table 1: Usage Table Format

Number of Row	189,960	
Number of Columns	21	
Usage Report Database	**Column Name**	**Column Details**
	MONTH	- Base Month of the data
	MSISDN	- Customer Unique Identification Number
	BSCode	- Business Code
	prp_pst_tag	- Customer Billing type. Prepaid/Postpaid
	FIRST_CONNECTION_DATE	- Onboarding Date with the Network
	Activation Status	- Customer's Connection status
	PRODUCT	- Type of Connection
	Voice Revenue	- Revenue of Outgoing Call
	Data Revenue	- Revenue of Internet Use
	SMS Revenue	- Revenue of Text Messaging
	MMS Revenue	- Multimedia Messaging Revenue
	VAS Revenue	- Revenue for value added services
	Voice Out Min	- Total Minutes of usage
	MB_USAGE	- Total Internet usage
	HS_CONN	- Handset Configuration
	USIM_TAG	- SIM Card status 3G/4G
	Voice On Min	- Total Minutes of Usage within same Operator
	Voice OFF Min	- Total Minutes of Usage with other Operator
	PP_GROUP	- Group of the price plan
	NOT_USING_TILL_DAYS	- Inactive status in the network
	Mix Bundle Revenue	- Bundle product (Voice+Data offer) revenue

3.3. Data Flow Diagram

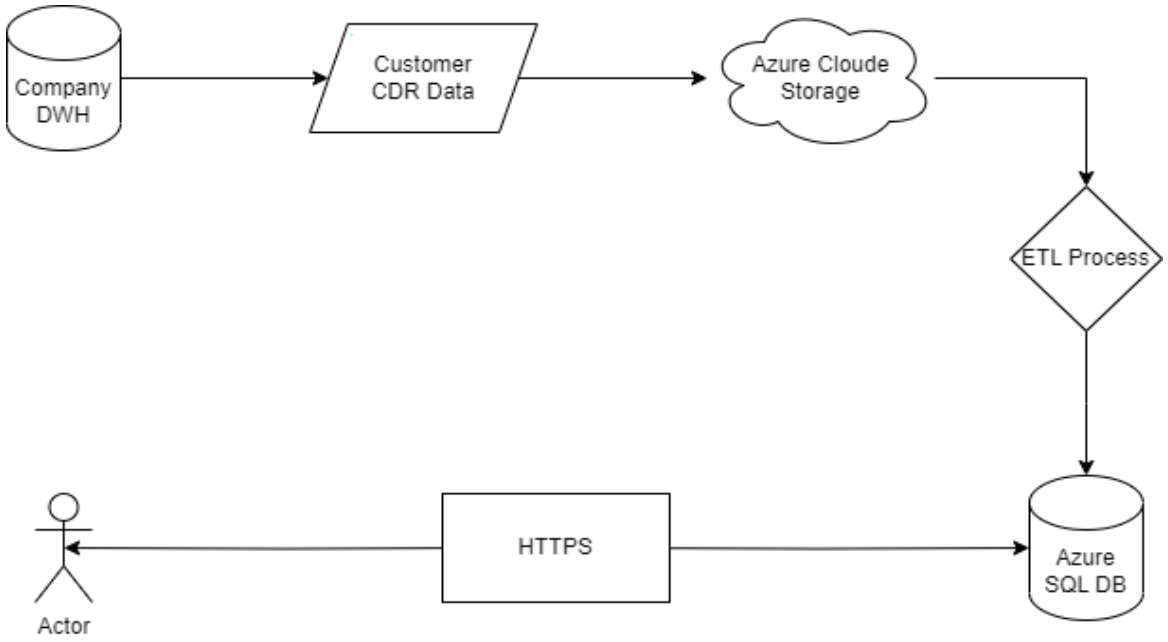


Figure 1: Data Flow Diagram

As shown in the Figure 1, Customer usage data will be fetched from company data wiare house directly to the Azure Cloud. In Azure cloud this data will be preprocessed and cleaned. After that the data will be processed for customer profiling according to their demographic and usage pattern. Profiled data will be pushed to Azure SQL DB. User with appropriate access can search customer according to the campaign need and data can be extracted in a CSV file. User doesn't need any IT knowledge for data fetching.

3.4. Tools/Software

- Google Colab- Google Colab, short for Google Colaboratory, is a free cloud-based platform provided by Google that allows users to write and execute Python code in a collaborative and interactive environment. Colab provides a Jupyter Notebook interface that enables data scientists, researchers, and developers to create and share documents that contain live code, equations, visualizations, and explanatory text.
  In this project I have used google colab for running jupyter notebook and CTGEN algorithm to produce synthetic data.

- Oracle Database- Oracle Database is a powerful and widely used relational database management system (RDBMS) developed by Oracle Corporation. It provides a structured and efficient way to store, manage, and retrieve data, making it a core technology for businesses and organizations of all sizes. Oracle Database offers a range of features to ensure data integrity, security, scalability, and performance.

In this project I have created a demo database where telecom customer database was hosted to demonstrate the connection between company DWH and Azure Cloud service.

- PySpark:- PySpark is the Python library for Apache Spark, an open-source big data processing framework. PySpark allows Python developers to work with Spark's powerful distributed computing capabilities to process and analyze large datasets. Spark is designed to handle various data processing tasks, including batch processing, interactive querying, machine learning, and more, with a focus on speed, ease of use, and scalability.
  In this project PySpark was used to process the data in the cloud. .

3.5. Cloud Platform

- Microsoft Azure - Microsoft Azure, is a comprehensive cloud computing platform and set of services provided by Microsoft. It offers a wide range of cloud-based solutions that empower businesses and organizations to build, deploy, and manage applications and services through Microsoft's global network of data centers. Azure provides resources such as virtual machines, storage, databases, networking, analytics, artificial intelligence, and more, allowing users to innovate and scale without the need to invest in physical infrastructure.

  Key Features of Microsoft Azure:

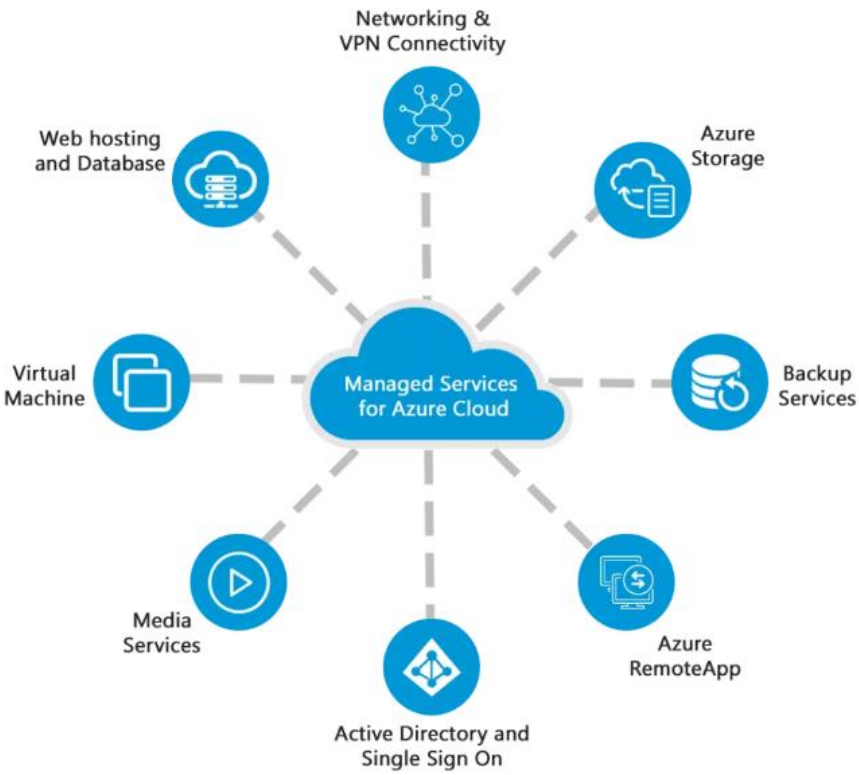


Figure 2: Key Features of Microsoft Azure

- Azure Databricks: - Azure Databricks is used for a wide range of data-related tasks, such as data preparation, data exploration, feature engineering, data visualization, machine learning model development, and more. It helps data professionals accelerate their data analysis workflows and leverage the power of cloud computing for processing and analyzing big data.
  In this project I have used Azure Databricks and pyspark to load the data from cloud storage, process those and save transformed data back to azure cloud storage.

- Azure SQL: - Azure offers a comprehensive relational database service called "Azure SQL Database." It's a fully managed cloud database service provided by Microsoft that allows you to create, operate, and scale relational databases without the need to manage the underlying infrastructure. Azure SQL Database is based on Microsoft SQL Server and provides a wide range of features tailored for cloud deployment.
  Reasons of using Azure SQL:

  - It is a managed SQL database service in the cloud.
  - It offers built-in scalability options, allowing you to scale your database up or down based on your performance requirements.
  - Manage Relational Database Management tasks- such as data migration, backup, recovery, and patching.
  - Azure SQL Database supports built-in machine learning and analytics features, allowing you to perform advanced analytics and predictive modeling on your data.

3.6. Process of Work Distribution

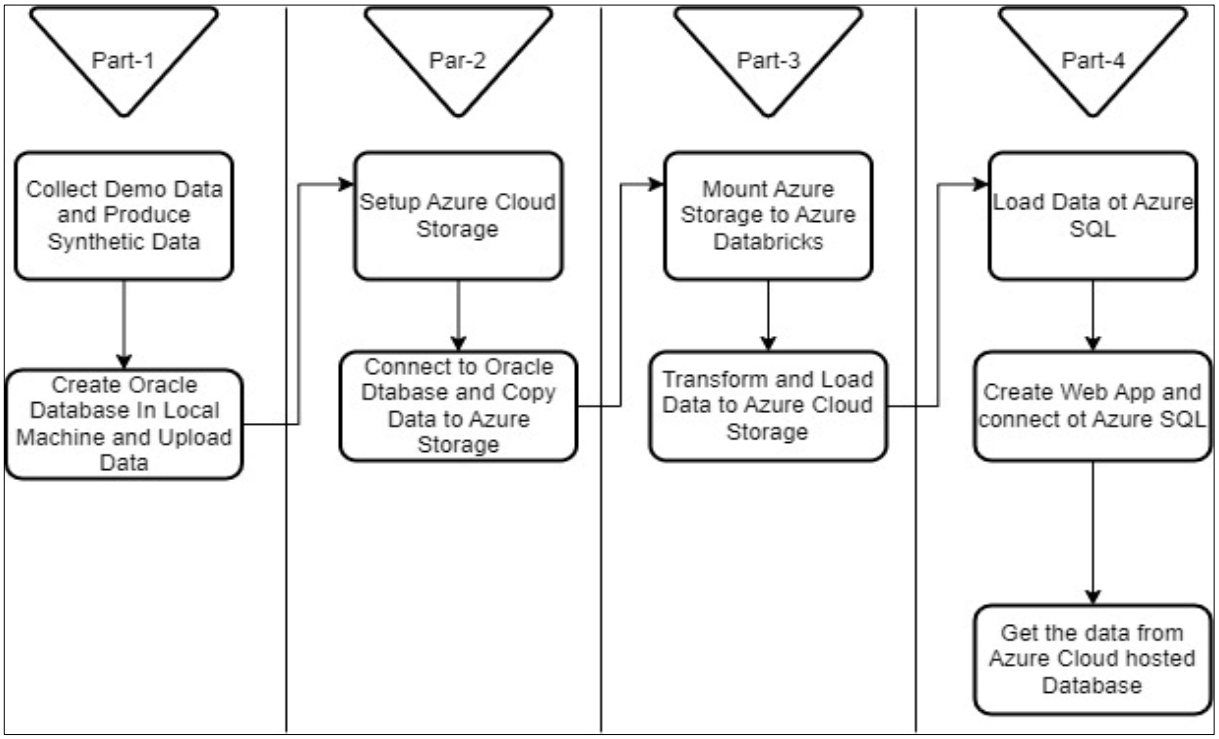


Figure 3: Parts of Process of Project

As shown in Figure 3, the whole project is based on 4 parts of work to make this possible to provide viewable and usable.

Details of process of project are provided described below:

Part 1: Create Customer usage Database

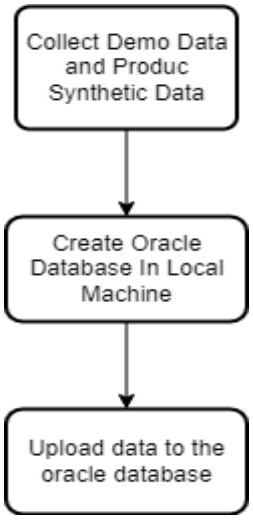


Figure 4- Process of creating Customer usage Database

As articulated in Figure 4, there has been used a python programing code (full code is provided at Appendices part) to prepare synthetic data from real data. From this file 190K data has been used to create customer usage database in the local machine.

Part 2 (a): Setup Azure Cloud Storage:

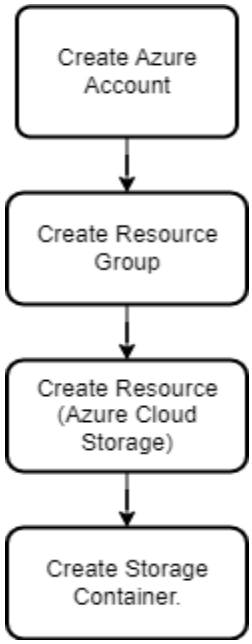


Figure 5- Process of creating Azure Data Lake Storage

Above mentioned Figure 5 is shown the steps of creating Azure Data Lake storage, where data from oracle database will be stored:

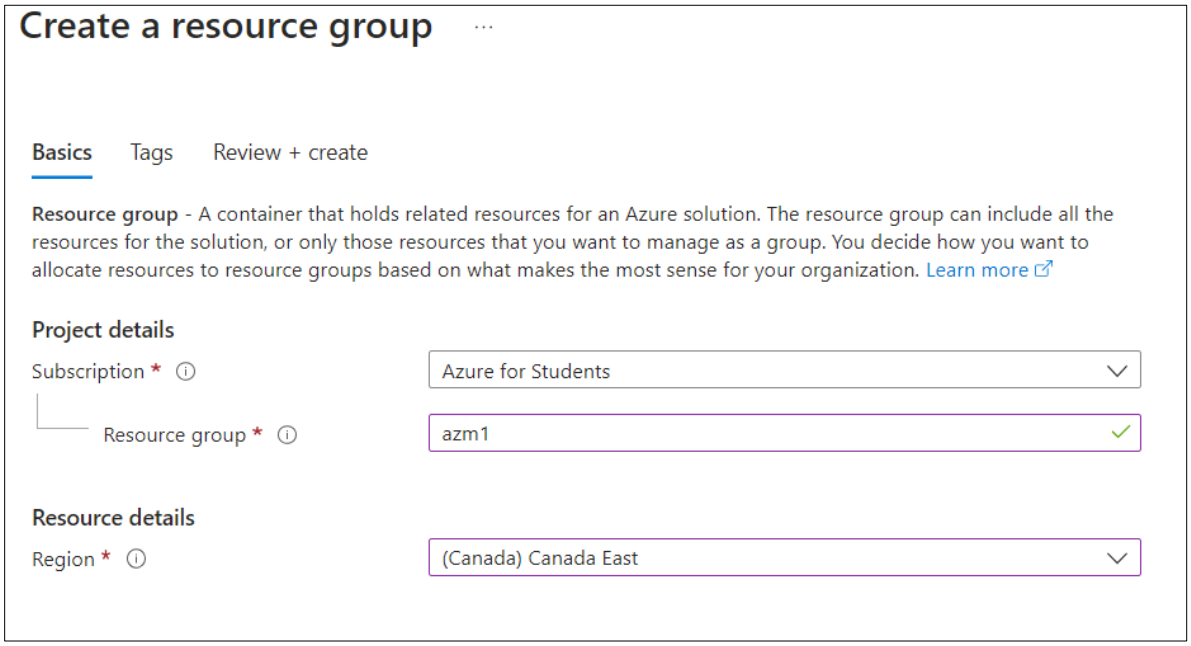


Figure 5 (a)- Process of creating resource group under Azure Subscription

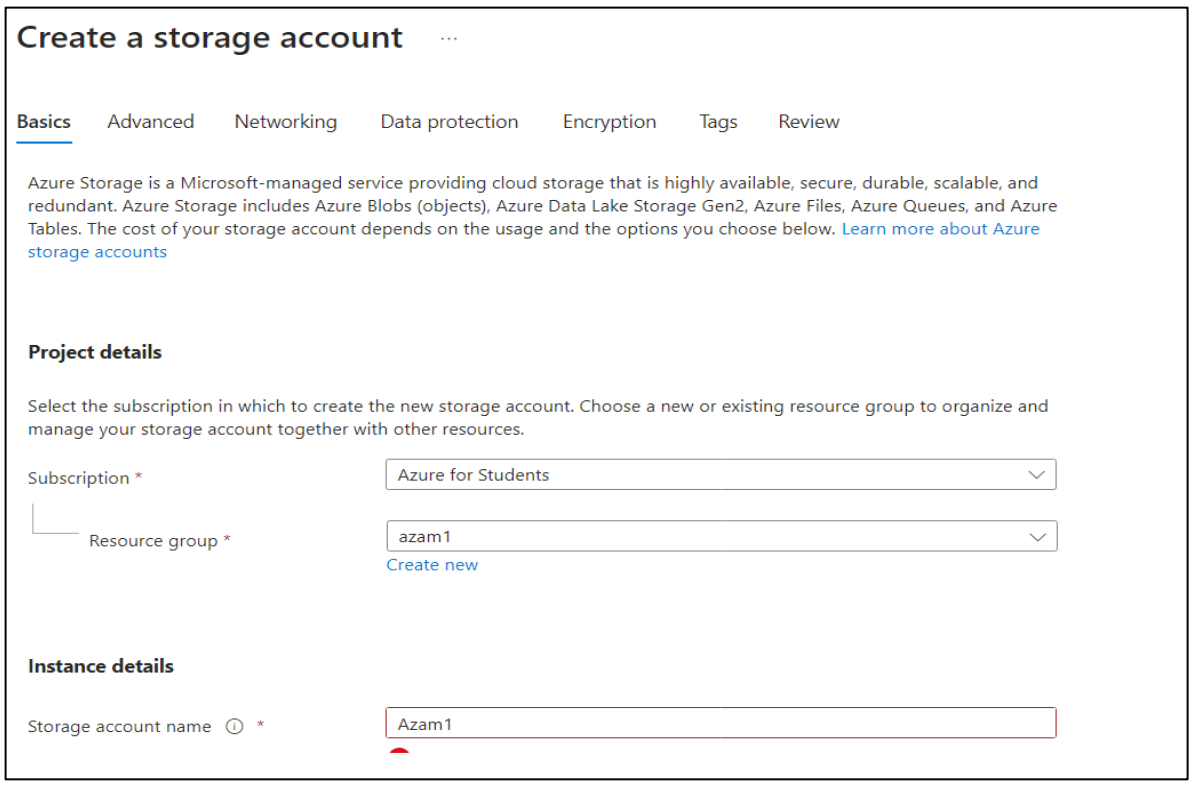


Figure 5 (b)- Process of creating Azure Storage Account

As displayed in the Figure 5(b): Select Create resource under resource group then select storage account

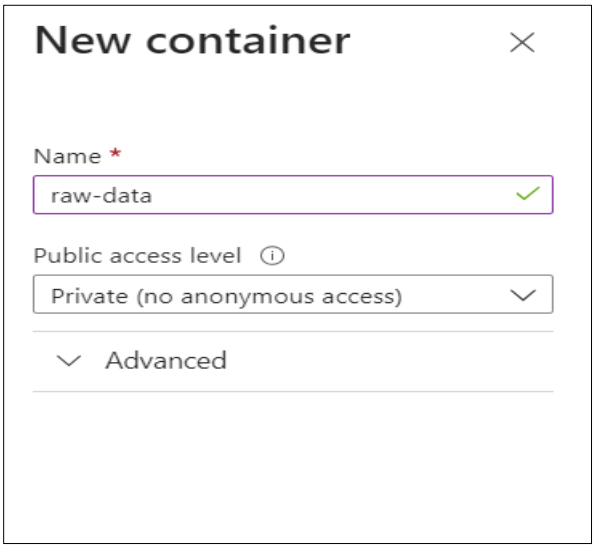


Figure 5 (c)- Process of creating Azure Storage Container

As displayed in the Figure 5(c): Select container→ +Container to create new container in Azure Storage account

Part 2 (b): Connect to Oracle Database and copy data to Azure Storage Container

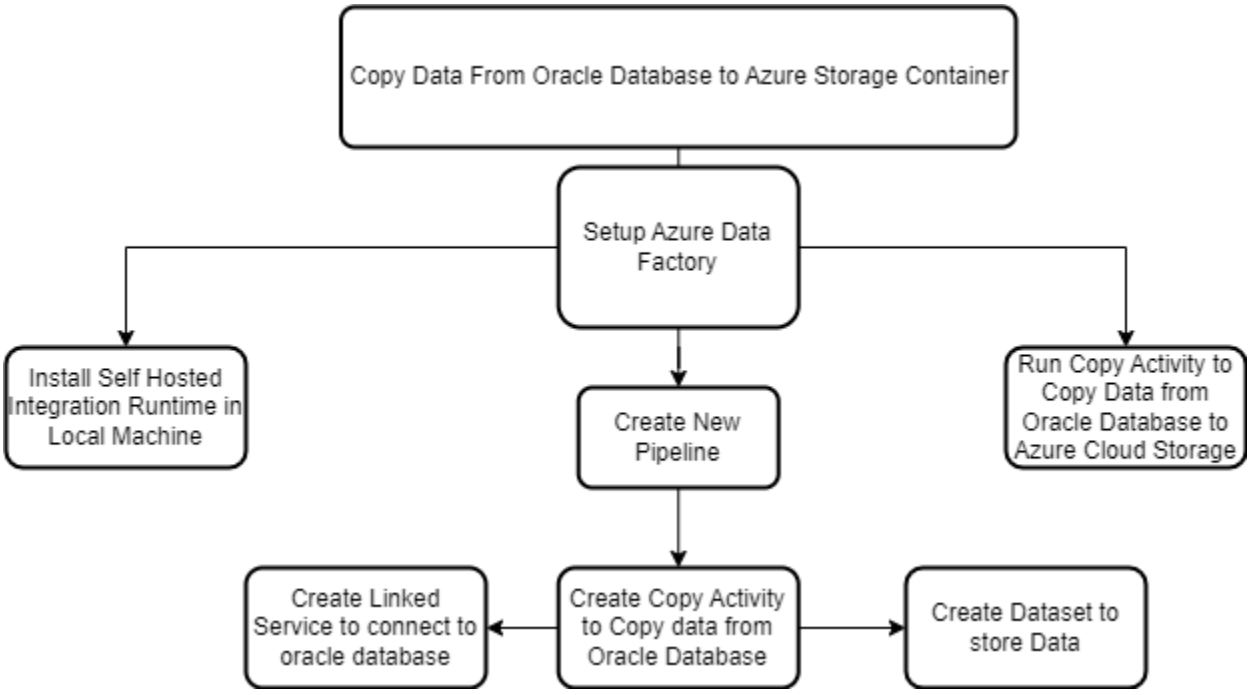


Figure 6- Process of Copying Data from Oracle Database to Azure Cloud

Above shown Figure 6 is shown, the process of copying data from oracle database to azure storage container. The below screenshots are provided to support the steps described in mentioned figure 6:

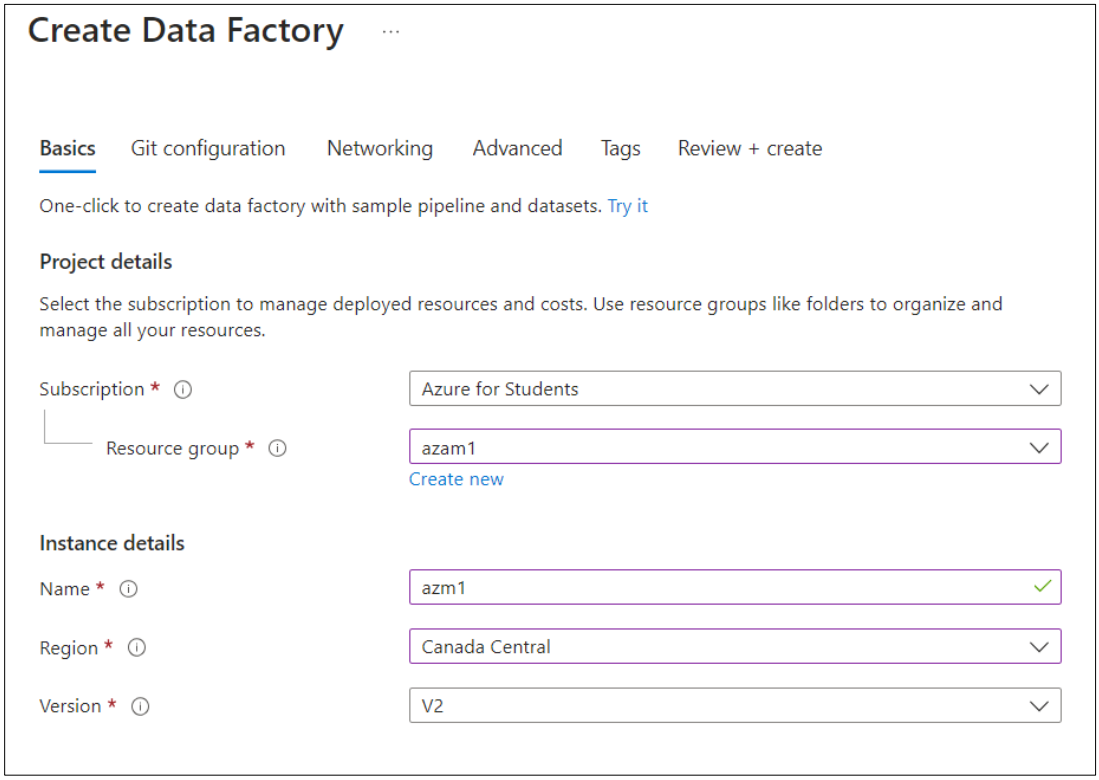


Figure 6 (a)- Process of Creating Data Factory

As presented in Figure 6(a), Step-1 – Select create resource in resource group→ select data factory

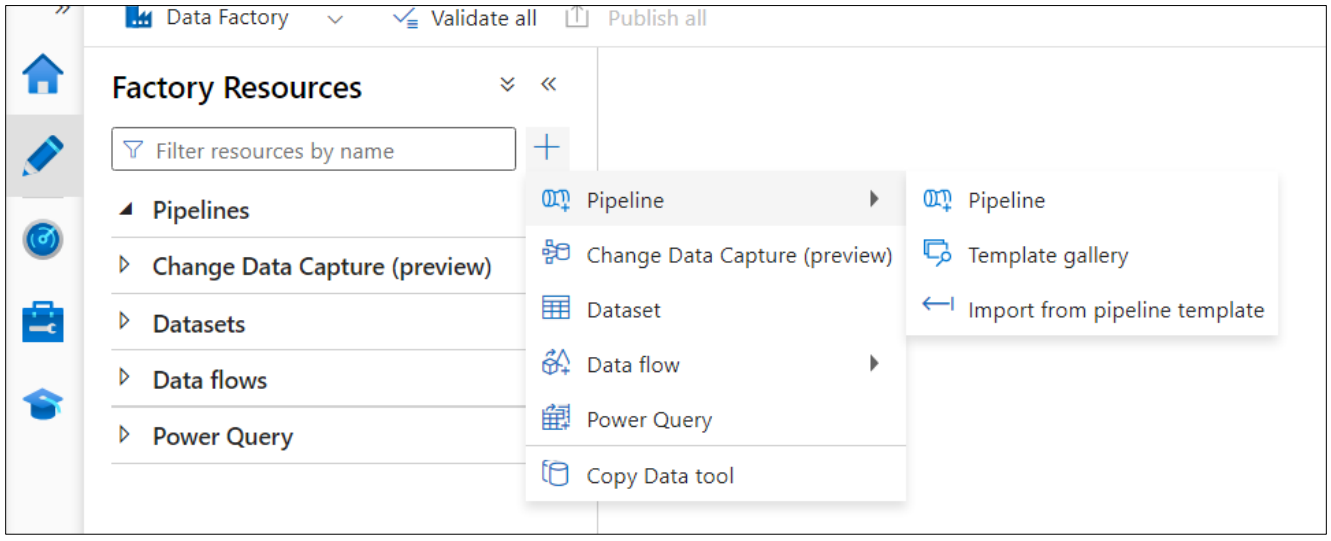


Figure 6 (b)- Process of Creating Pipeline

As presented in Figure 6(b), Step-2 – Select Author in Data Factory → Create New → select pipeline-> Name it

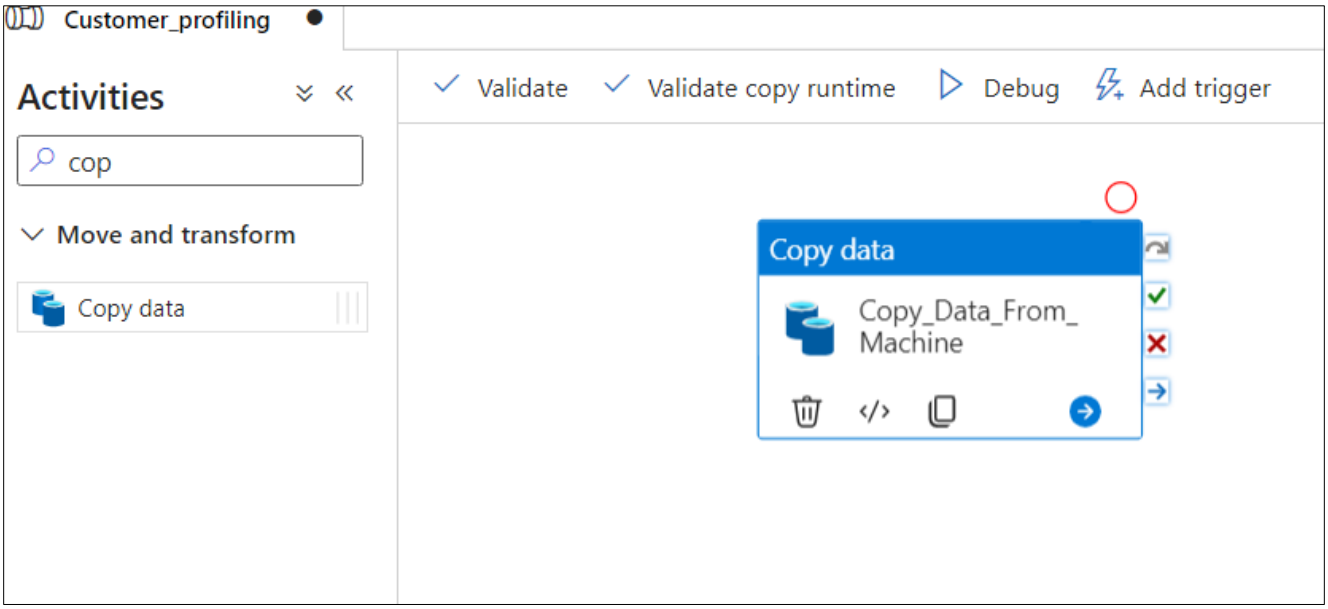


Figure 6 (c)- Process of Creating Copy Activity

As presented in Figure 6(b) and Figure 6(c), Step-3 – Search Activity in pipeline-> Select Copy Activity->Change Name

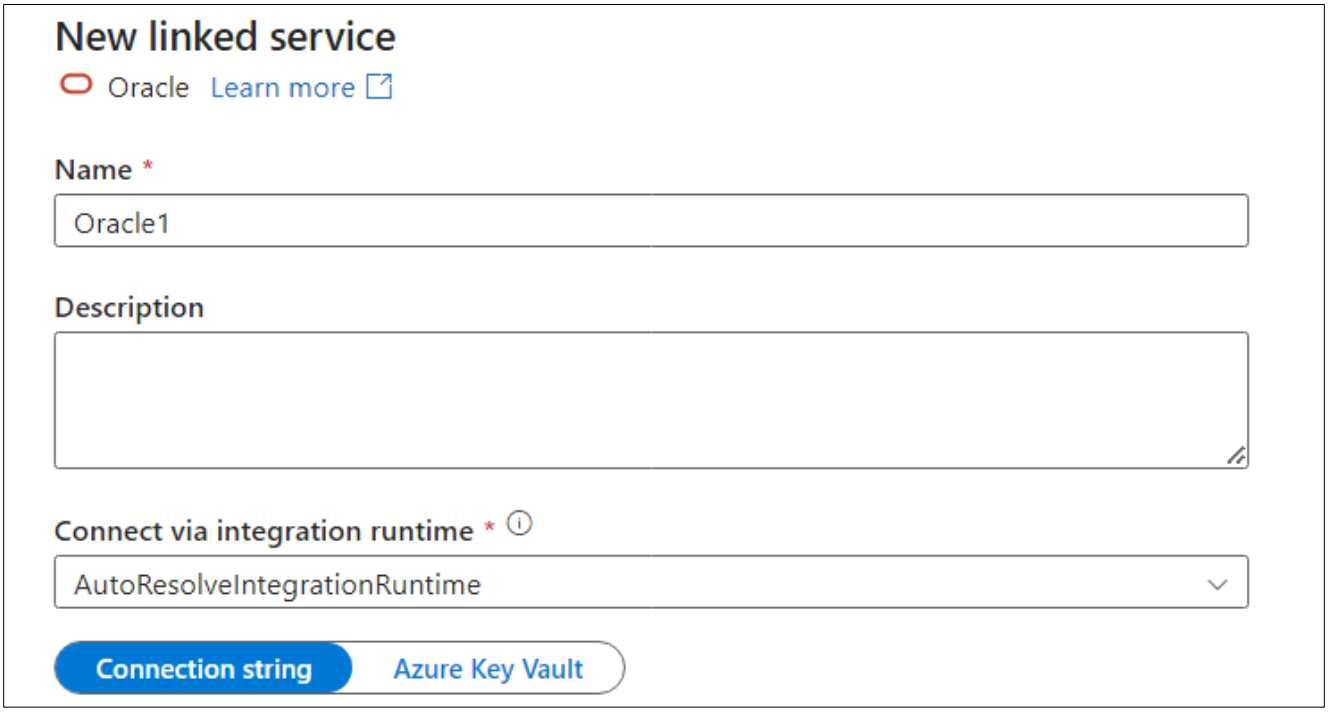


Figure 6 (d)- Process of Creating New Linked Service

As presented in Figure 6(d), Step-4 – Select dataset -> new Dataset->search and select oracle-> linked service-> new

Click on Connect via Integration runtime-> select new-> select self hosted-> select new -> select express setup.

This will install and run self hosted runtime in the local machine.

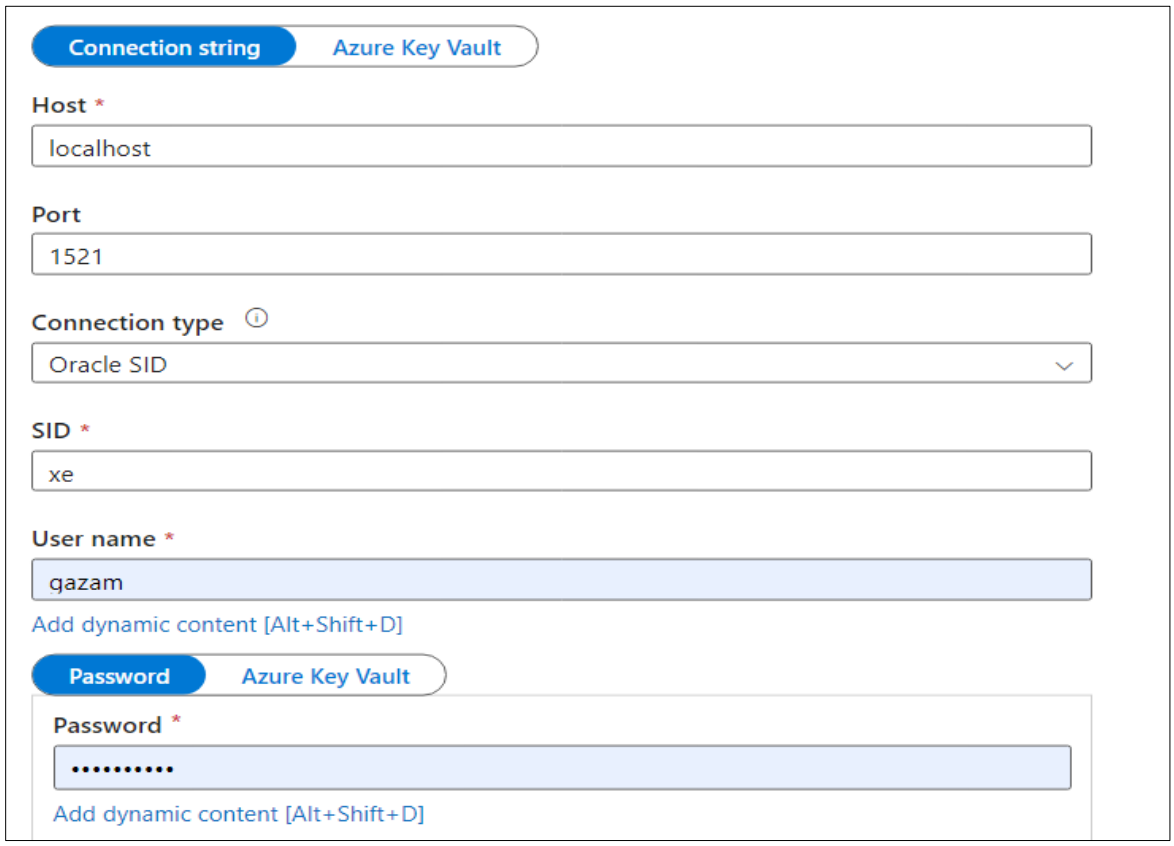


Figure 6 (e)- Process of creating new linked service

As presented in 6(e), Step-5 – Put the information about oracle database in local machine and select create to create linked service.

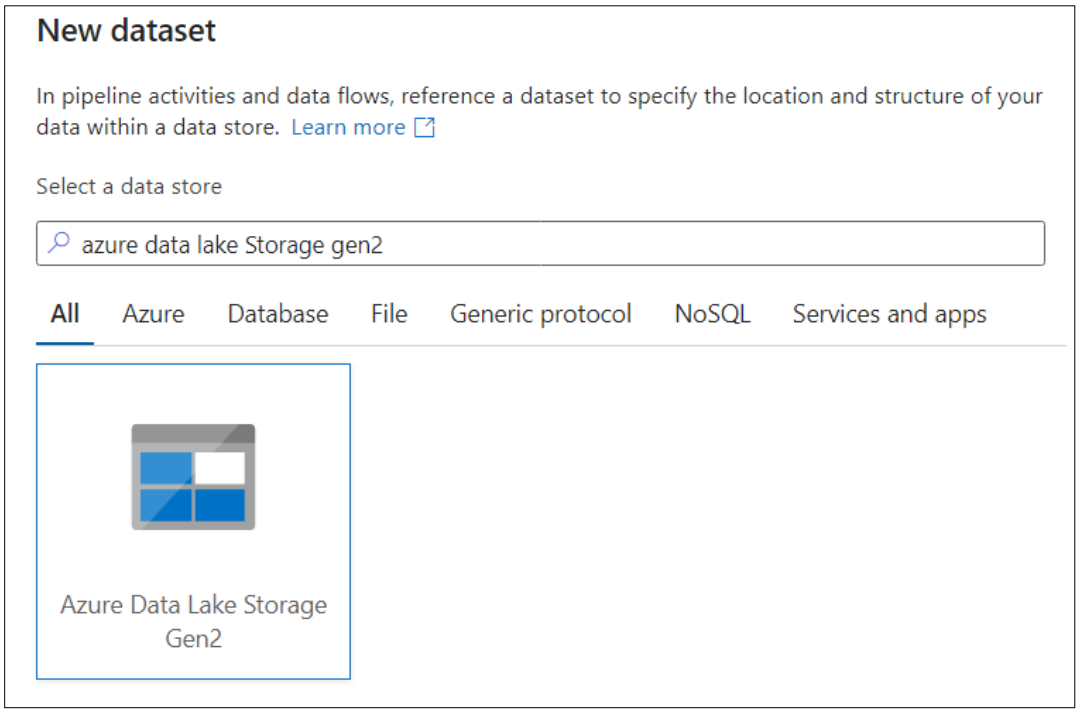


Figure 6 (f)- Process of creating new Dataset

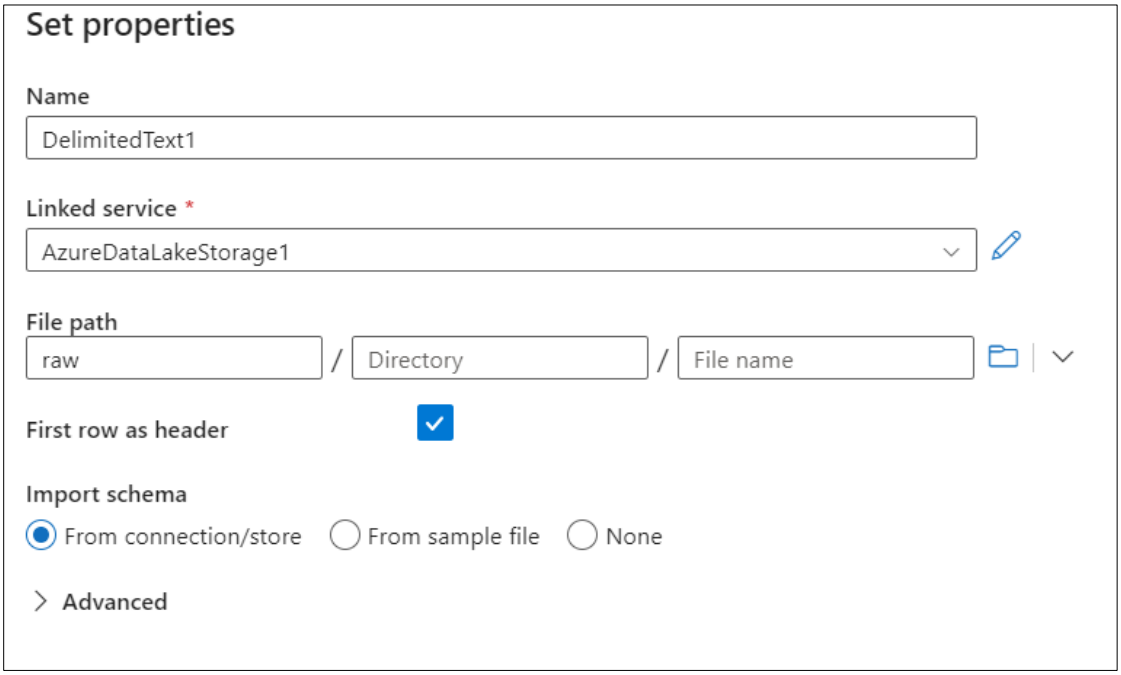


Figure 6 (g)- Process of selecting the container for data storage

Part 3: Process Data in Azure Data bricks

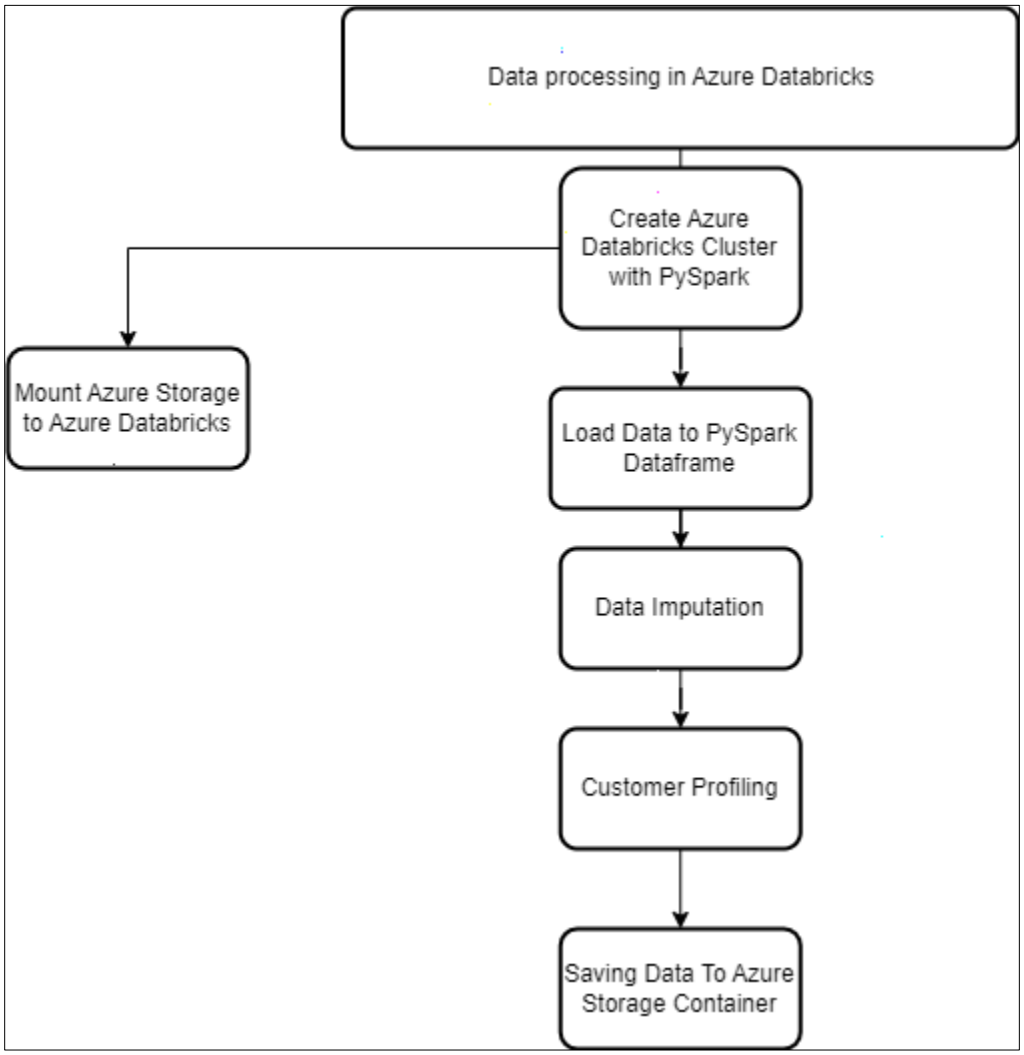


Figure 7- Data Processing in Azure Data Bricks

As per the above presented Figure 7, in order to process data, we need to create an Azure databricks cluster.

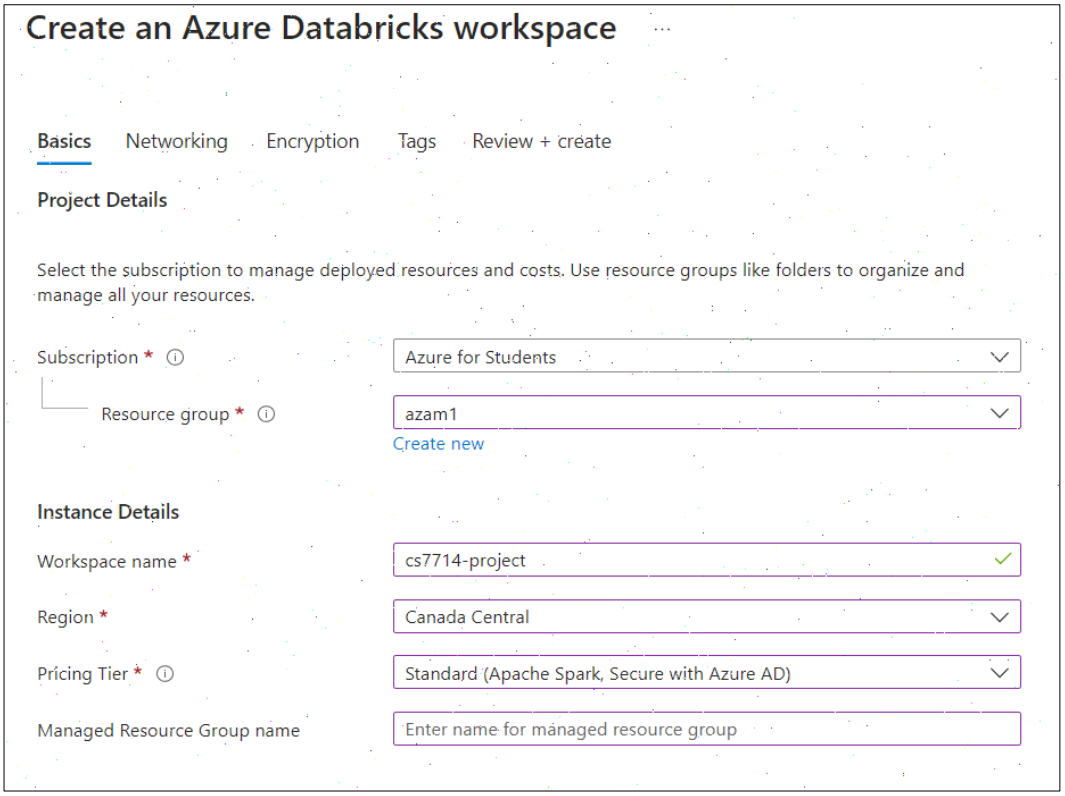


Figure 7(a)- Create Databricks cluster with PySpark

As presented in 7(a), Step-1 – under resource group select new resource->select Azure Databricks

Create Workspace in Azure Databricks: select workspace in azure databricks-> select workpspace -> select create from the menu-> select notebook.

This will create a PySpark notebook in databricks workspace. Then run below code to mount azure storage to databricks

```
Cmd 1

Python

configs = {
  "fs.azure.account.auth.type": "CustomAccessToken",
  "fs.azure.account.custom.token.provider.class": spark.conf.get("spark.databricks.passthrough.
  adls.gen2.tokenProviderClassName")
}

dbutils.fs.mount(
  source = "abfss://raw-data@azam1.dfs.core.windows.net/",
  mount_point = "/mnt/raw-data",
  extra_configs = configs)

Out[2]: True

Command took 11.29 seconds -- by aad693@uregina.ca at 8/14/2023, 7:04:24 PM on Forhad Azam's Cluster

Cmd 2
```

Figure 7(b)- Mount Storage in Azure Databricks

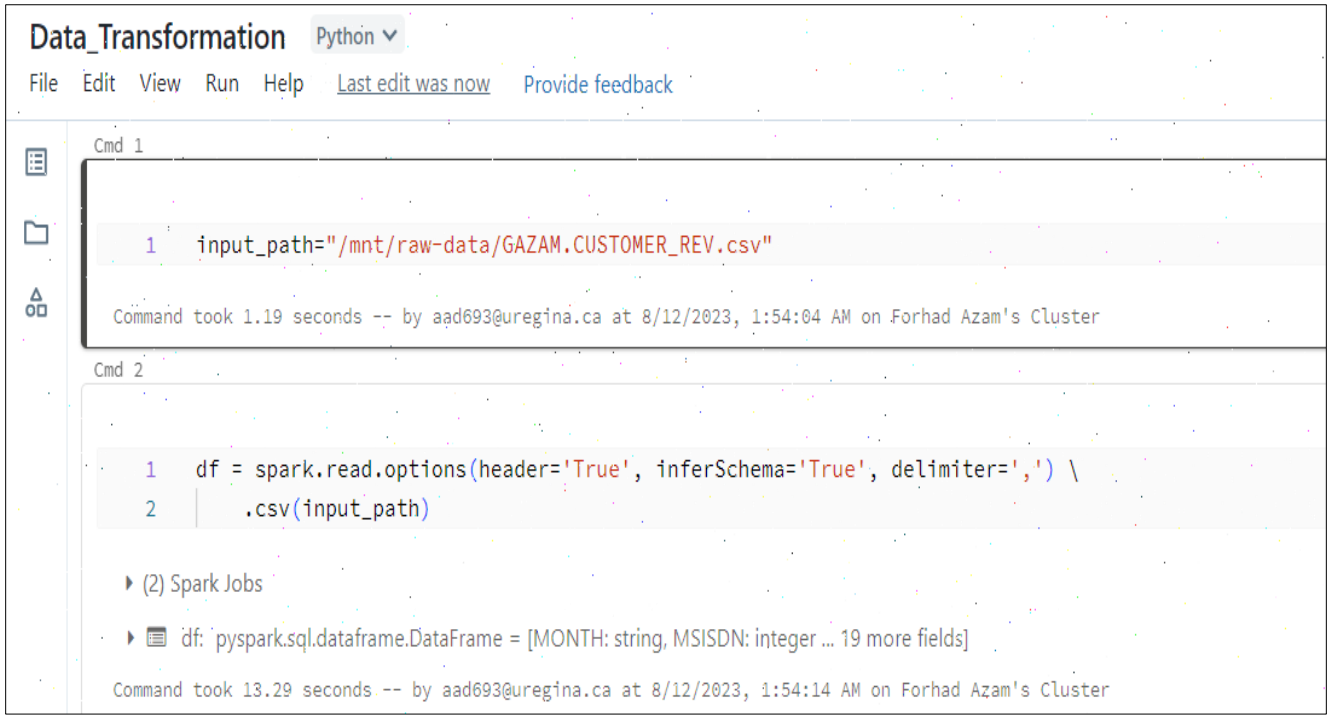


Figure 7 (c)- Load data from Azure Storage to databricks for processing

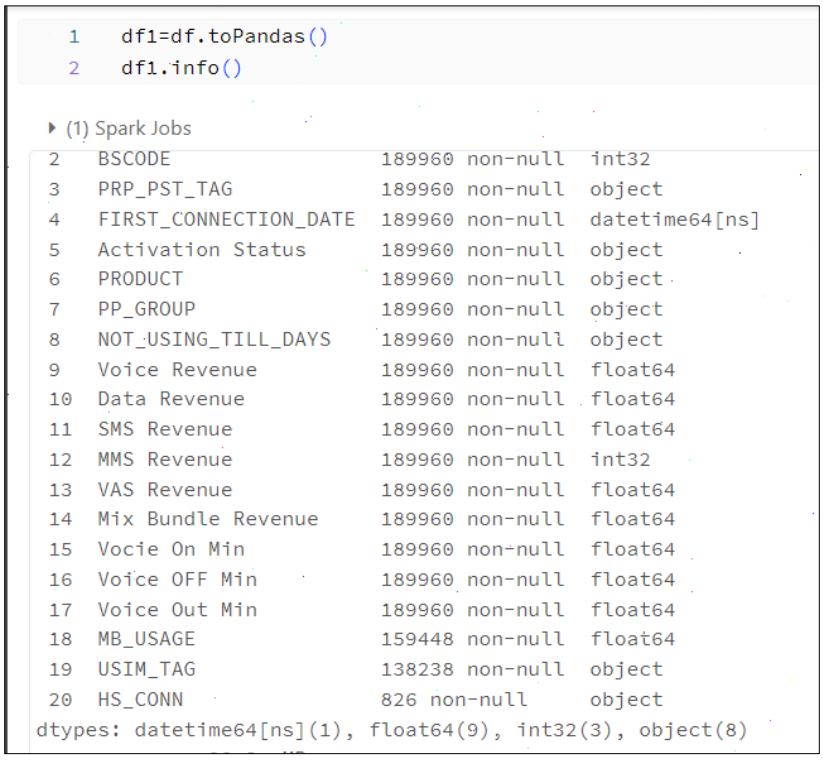


Figure 7 (d)- Finding Data anomaly

```
1  df1["HS_CONN"].fillna("UNKNOWN",inplace=True)
2  df1["USIM_TAG"].fillna("N",inplace=True)

Command took 0.07 seconds -- by aad693@uregina.ca at 8/14/2023, 7:35:04 PM on Forhad Azam's Cluster
```

Figure 7 (e)- Data Imputation

```
1  df=df.withColumn("Total_Rev",col("Voice Revenue")+col("Data Revenue"))

▸ df: pyspark.sql.dataframe.DataFrame = [MONTH: string, MSISDN: integer ... 20 more fields]
Command took 0.18 seconds -- by aad693@uregina.ca at 8/14/2023, 9:08:33 PM on Forhad Azam's Cluster

Cmd 9

1  df=df.withColumn("Rev_Segment",expr("CASE when Total_Rev=0 then '0'\
2                                       WHEN Total_Rev>0 AND Total_Rev <=100 then '>1-<=100'\
3                                       WHEN Total_Rev>100 AND Total_Rev <=200 then '>100-<=200'\
4                                       WHEN Total_Rev>200 AND Total_Rev <=350 then '>200-<=350'\
5                                       WHEN Total_Rev>350 AND Total_Rev <=500 then '>350-<=500' else '>500' END"))

▸ df: pyspark.sql.dataframe.DataFrame = [MONTH: string, MSISDN: integer ... 21 more fields]
Command took 0.24 seconds -- by aad693@uregina.ca at 8/14/2023, 9:08:38 PM on Forhad Azam's Cluster

Cmd 10

1  df=df.withColumn("RG_Customer",expr("CASE when Total_Rev=0 then 'N'\
2                                       else 'Y' END"))

▸ df: pyspark.sql.dataframe.DataFrame = [MONTH: string, MSISDN: integer ... 22 more fields]
Command took 0.16 seconds -- by aad693@uregina.ca at 8/14/2023, 9:10:30 PM on Forhad Azam's Cluster
```

Figure 7 (e)- Profiling

```
1  output_path="/mnt/transformed/temp/"
2
3  # Write CSV file with column header (column names)
4  df.coalesce(1).write.option("header",True).format("csv").mode("overwrite").save(output_path)
5

Cmd 14

1  files=dbutils.fs.ls(output_path)
2  name=""
3  for f in files:
4      if f.name.endswith(".csv"):
5          name=f.name
6
7  dbutils.fs.cp(output_path+name,"/mnt/transformed/customer_rev_tr.csv")
8
```

Figure 7 (e)- Saving Data to Azure Cloud Storage

Part 4: Load Data to Azure SQL and connect to web App

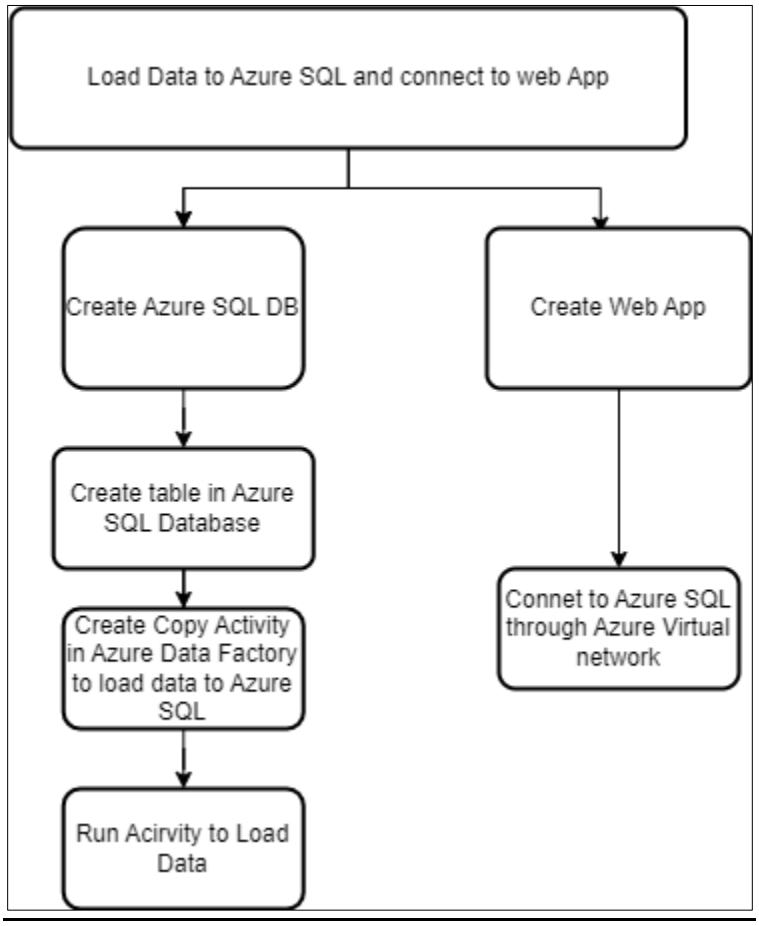


Figure 8- Loading Data to Azure SQL and connect to web app

As per the above presented Figure 8, in order to load data to Azure SL, we need to create an Azure SQL database.

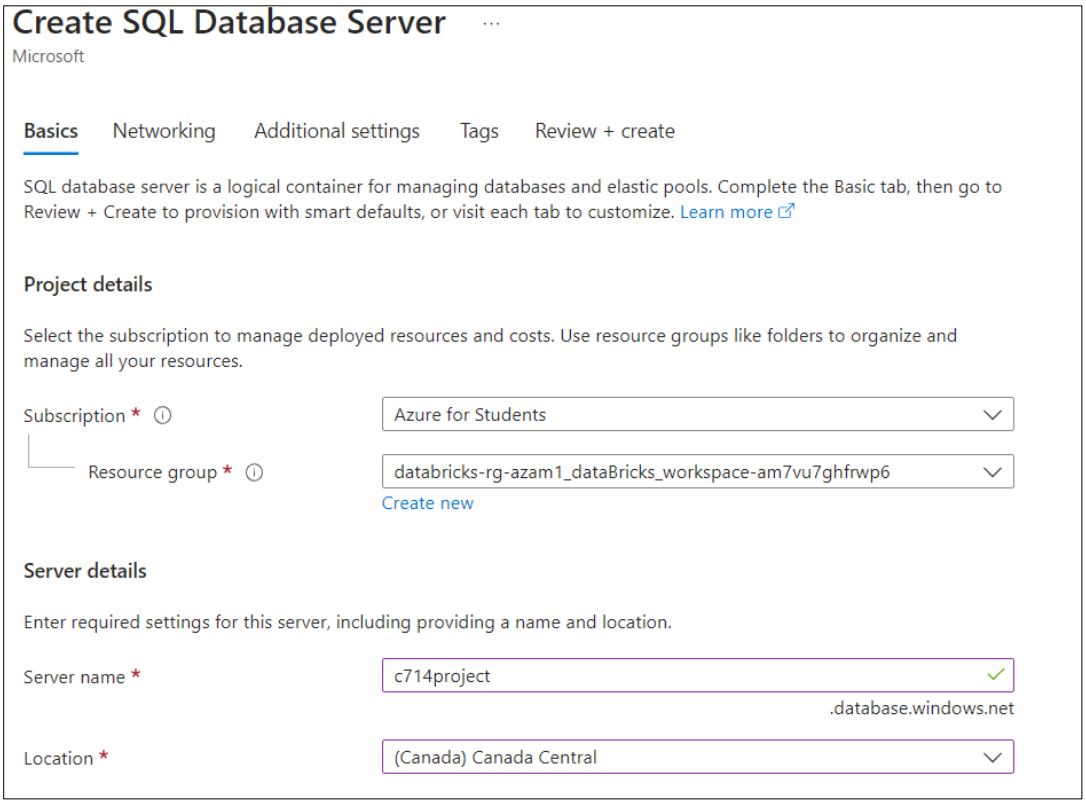


Figure 8(a)- Creating Azure SQL Server

As presented in 8(a), – under resource group select new resource->select Azure SQL->select SQL

Under the server option select new -> give database name -> give username-> give password-> Select create

From resource group select Azure SQL db-> select Query editor-> enter credentials to login

In the query editor past below query and hit Run to create table named “Customer_Rev”

```
CREATE TABLE CUSTOMER_REV (
    [MONTH] NVARCHAR(50),
    [MSISDN] INT,
    [BSCODE] INT,
    [PRP_PST_TAG] NVARCHAR(50),
    [FIRST_CONNECTION_DATE] DATE NOT NULL,
    [Activation Status] NVARCHAR(50),
```

```
    [PRODUCT] NVARCHAR(50),
    [PP_GROUP] NVARCHAR(50),
    [NOT_USING_TILL_DAYS] NVARCHAR(50),
    [Voice Revenue] FLOAT,
    [Data Revenue] FLOAT,
    [SMS Revenue] FLOAT,
    [MMS Revenue] INT,
    [VAS Revenue] FLOAT,
    [Mix Bundle Revenue] FLOAT,
    [Vocie On Min] FLOAT,
    [Voice OFF Min] FLOAT,
    [Voice Out Min] FLOAT,
    [MB_USAGE] FLOAT,
    [USIM_TAG] NVARCHAR(50),
    [HS_CONN] NVARCHAR(50)
);
```

Go to the Azure Data factory and create a Copy activity under pipeline to load data to Azure SQL DB table

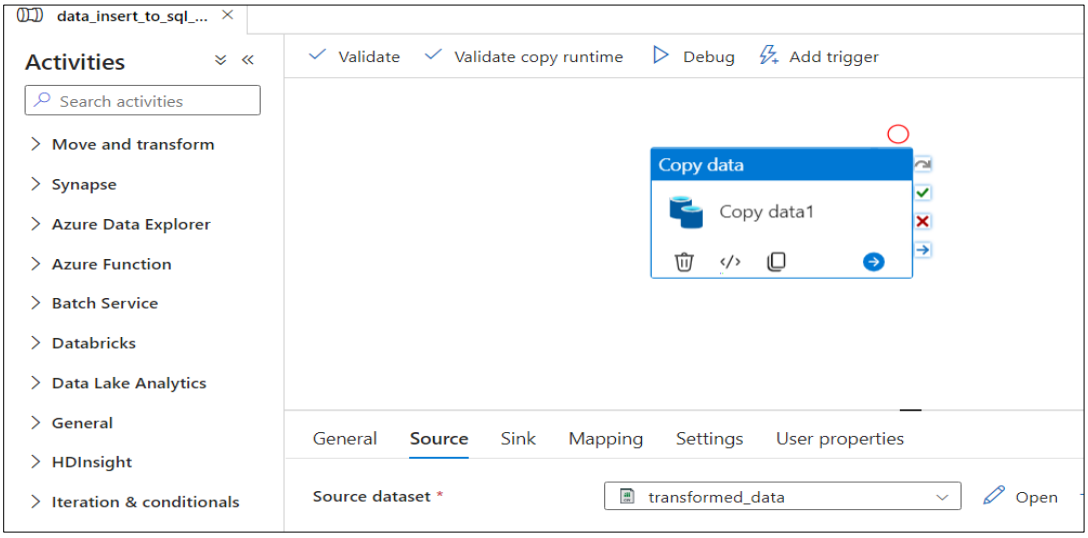


Figure 8(b)- Creating Copy Activity to Load data to Azure SQL

As show in Figure 8(b)- Select pipeline-> search and select copy activity-> select data source as transformed data container of azure storage and sink as azure sql db table then run this activity to load data to azure sql db

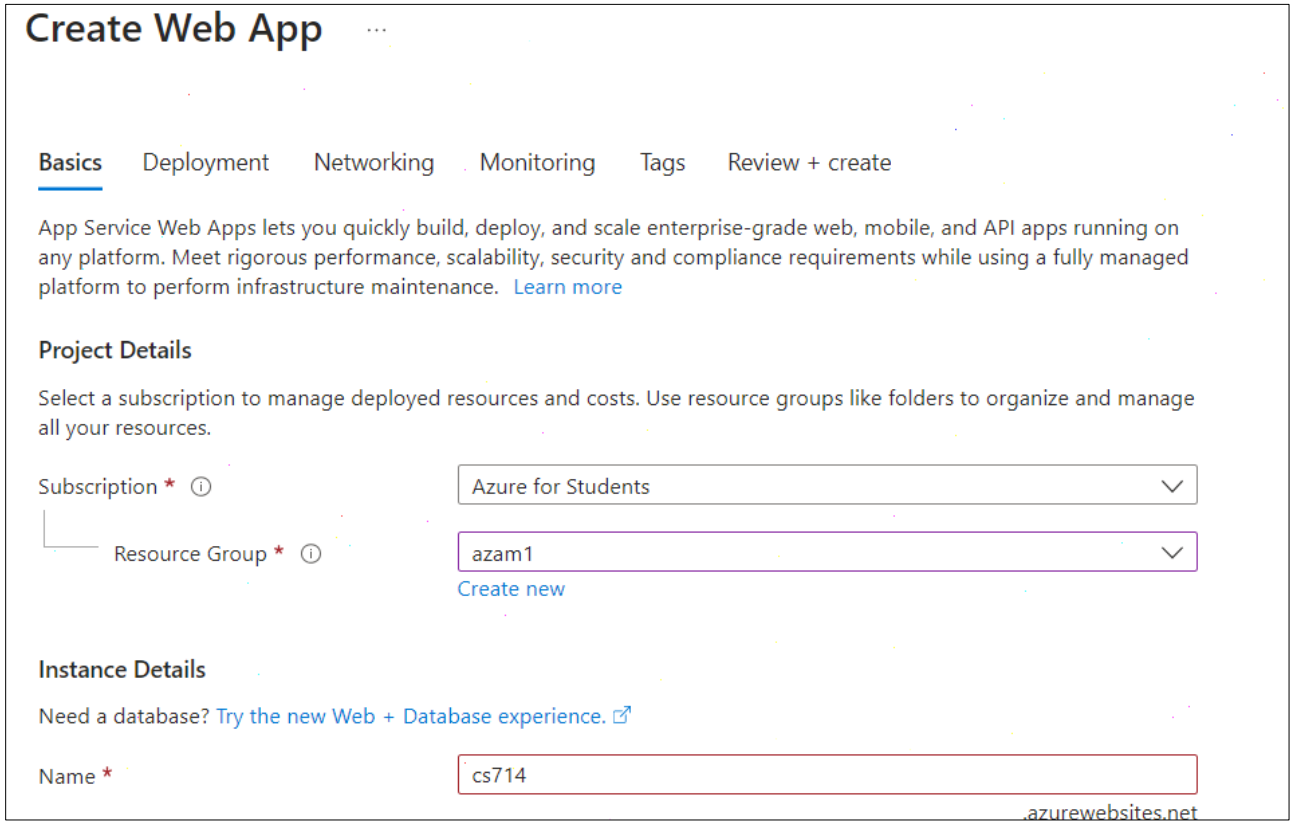


Figure 8(c)- Creating web app

As shown in figure 8(c)- we need to create a web app to for fetching data from azure sql database. For this select create new resource under resource group-> select web app-> select costing-> create

We need to create a Vnet to connect web app to the sql server

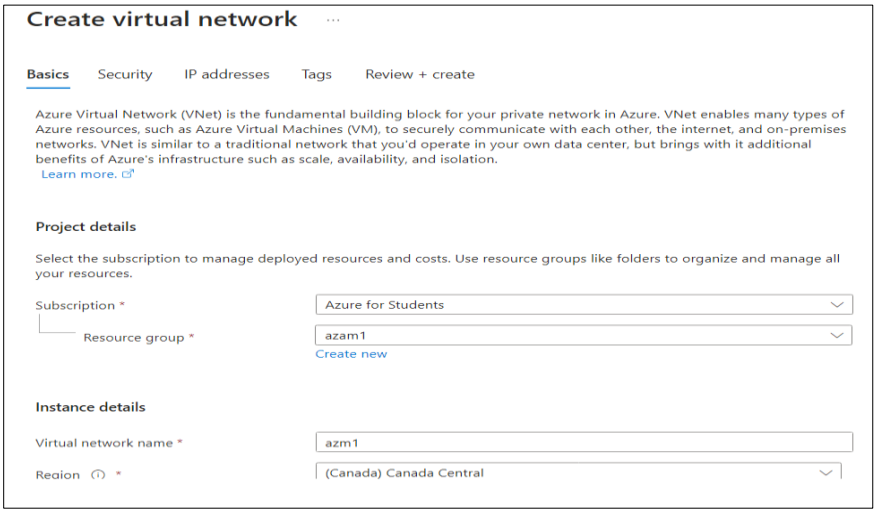
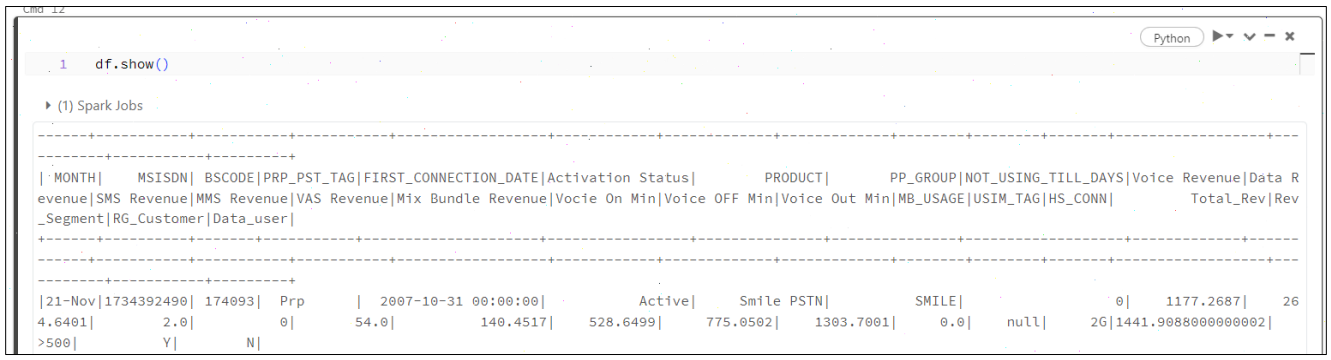


Figure 8(d)- Creating Virtual Network

4. Result

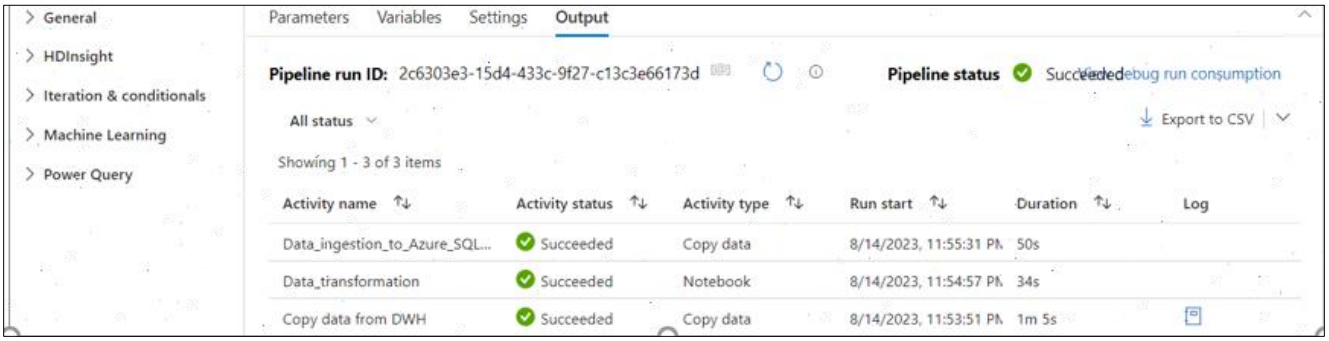
4.1. Sample Results and Discussion of Test Result

- Result on Customer data profiling:

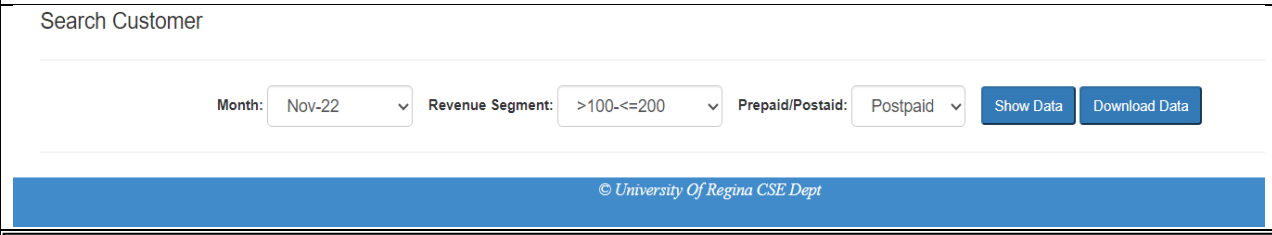
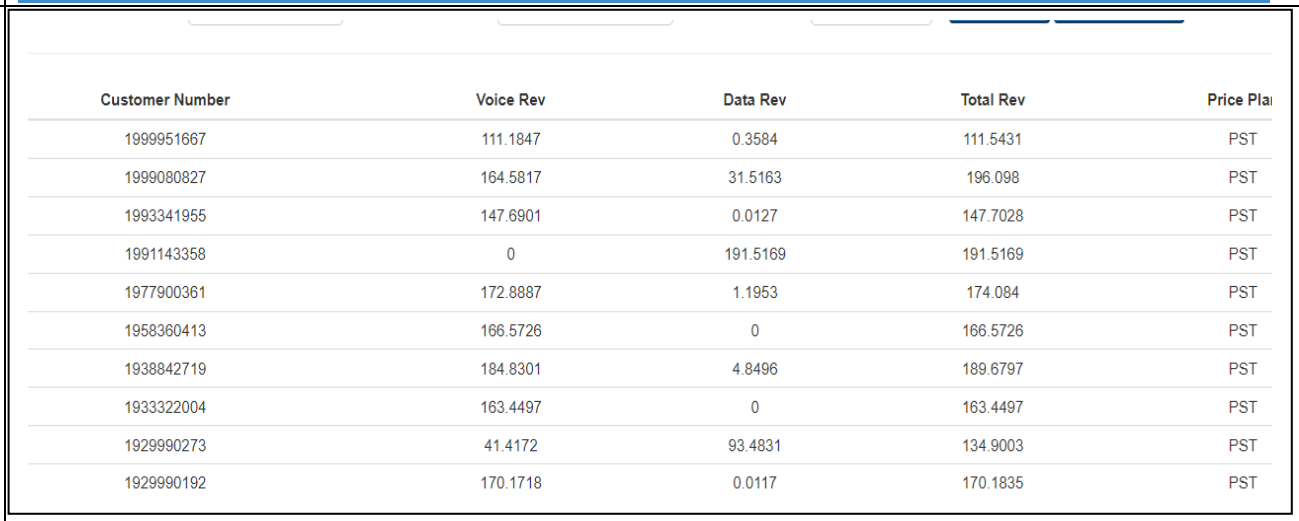


The first one is the Raw Data and the second one is after the profiling. There are several profiling parameters have been added like in which revenue segment the customer fall, if the customer is revenue generating customer or not, what is the total revenue generated by the customer and if the customer is using internet or not.

Visible result of Azure computing power:



It took only 2 Mins and 55 seconds to complete the whole process.

Data Fetching by User	With appropriate access any personnel from the company can search customer by using this link: University Of Regina,Data Science (cs714.azurewebsites.net)
Search Criteria	
Search Result	
Result Details	In the above mentioned table is showing In the month Of Nov-22 which customers have generated revenue between 100 and 200 and their connection type is postpaid

Customer Number	Voice Rev	Data Rev	Total Rev	Price Pla
1999951667	111.1847	0.3584	111.5431	PST
1999080827	164.5817	31.5163	196.098	PST
1993341955	147.6901	0.0127	147.7028	PST
1991143358	0	191.5169	191.5169	PST
1977900361	172.8887	1.1953	174.084	PST
1958360413	166.5726	0	166.5726	PST
1938842719	184.8301	4.8496	189.6797	PST
1933322004	163.4497	0	163.4497	PST
1929990273	41.4172	93.4831	134.9003	PST
1929990192	170.1718	0.0117	170.1835	PST

4.2. Purpose of this result:

Any product manager or Campaign manager can fetch data from this web app and make campaign offer or send personalized marketing communication to the target customer group. Anyone can fetch data according to their choice of customer segment.

5. Limitations

- Lack of Real Time Database- Obhiously telecome companies cannot share their data due to privacy and legal issue. Most of the time country's regulatory body doesn't allow telecom companies to share their data. I have produced the data with the help of some peer working in telecom industry and collected data from Kaggle data repository. Based on the idea I have produced synthetic data for the project.
- Less of Relevant Work Paper- The concept of Cloud computing and the concern of Big Data is identified by social media platform and social screening platform at very long time ago. Telecom companies are mostly new adopters, behind the security purpose of data storage. But this is been an aggressive movement for the last 10 years approximately, that they slowly adopting. So, the research area is still blurred.
- Time Constraints- The total project time is approximately less that any other project. There are many analytics can be done with this amount of data.

6. Possible Extension

- Work on real life dataset- Future plan is to implement same project using the concept work on real life data set from telecom industry.
- Integration of Machine learning algorithms for real-time personalizing
- Cross Industry Collaboration: Many other industries can use telecom customer data for their business purpose like credit scoring and to have idea about customers geographical concentration.
- Integrate the Power-BI tool for more statistical analysis: Power BI is and statistical analytical tool developed by Microsoft. These data can be pushed to Power BI from where a holistic view of customers can be monitored.

7. Conclusion

The whole idea of this project is to demonstrate how Cloud Computing Can easily manage large volumes of data and the scalable computing power can analyze the data to extract valuable insight from it. Automating the process of Data Management and profiling system. The right use of technology which can save time and money

8. Reference:

1. https://www.kaggle.com/datasets/jpacse/datasets-for-churn-telecom?resource=download-- Kaggle Data Repository
2. "How to create synthetic data with CTGen machine learnig library" https://towardsdatascience.com/how-to-generate-real-world-synthetic-data-with-ctgan-af41b4d60fde
3. "What is Data ware house" https://www.oracle.com/ca-en/database/what-is-a-data-warehouse/
4. "Big data analytics in telecommunications:" https://www.sciencedirect.com/science/article/pii/S131915782030553X
5. "What is personalized marketing and how it is used" https://martech.org/what-is-personalized-marketing-and-how-is-it-used-today/

9. Appendices:

Part 1: Producing Synthetic data:

```
pip install ctgan
```

```
from ctgan import CTGAN
from ctgan import load_demo
import pandas as pd
```

```
#real_data = load_demo()
real_data =pd.read_csv('sub_base_nov.csv')
```

```
real_data.shape
```

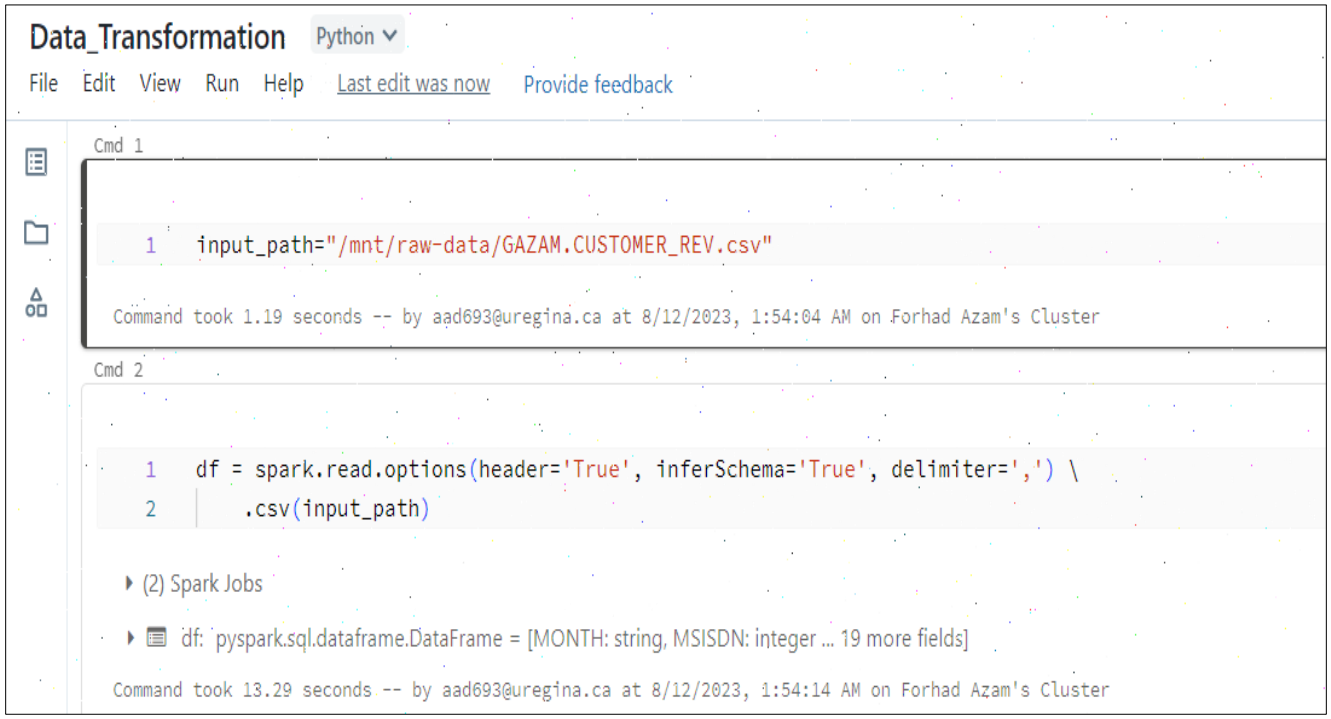
```
real_data.columns
```

```
Index(['MSISDN', 'BSCODE', 'PRP_PST_TAG', 'ACTIVATION_STATUS', 'PRODUCT',
       'FIRST_CONNECTION_DATE', 'NOT_USING_TILL_DAYS', 'MOBILITY_STR',
       'VOICE_REV', 'DATA_REV', 'MOU', 'MB_USAGE', 'USIM_TAG', 'HS_CONN',
       'MYGP', 'STAR', 'STAR_STATUS', 'IS_BULK'],
      dtype='object')
```

```
#real_data = pd.read_csv('sub_base_feb.csv')


# Names of the columns that are discrete
discrete_columns = [
    'PRP_PST_TAG',
    'ACTIVATION_STATUS',
    'PRODUCT',
    'FIRST_CONNECTION_DATE',
    'NOT_USING_TILL_DAYS',
    'USIM_TAG',
    'HS_CONN',
    'MYGP',
    'STAR',
    'STAR_STATUS',
    'IS_BULK'
]

ctgan = CTGAN(epochs=10)
ctgan.fit(real_data, discrete_columns)

# Create synthetic data
synthetic_data = ctgan.sample(200000)
```

Part2 Data processing in Azure data Bricks:

Mounting Azure Storage to Azure Data bricks



Finding Data Anomaly:

```
df1=df.toPandas()
df1.info()
```

(1) Spark Jobs

```
 2   BSCODE                  189960 non-null  int32
 3   PRP_PST_TAG             189960 non-null  object
 4   FIRST_CONNECTION_DATE   189960 non-null  datetime64[ns]
 5   Activation Status       189960 non-null  object
 6   PRODUCT                 189960 non-null  object
 7   PP_GROUP                189960 non-null  object
 8   NOT_USING_TILL_DAYS     189960 non-null  object
 9   Voice Revenue           189960 non-null  float64
 10  Data Revenue            189960 non-null  float64
 11  SMS Revenue             189960 non-null  float64
 12  MMS Revenue             189960 non-null  int32
 13  VAS Revenue             189960 non-null  float64
 14  Mix Bundle Revenue      189960 non-null  float64
 15  Vocie On Min            189960 non-null  float64
 16  Voice OFF Min           189960 non-null  float64
 17  Voice Out Min           189960 non-null  float64
 18  MB_USAGE                159448 non-null  float64
 19  USIM_TAG                138238 non-null  object
 20  HS_CONN                 826 non-null     object
dtypes: datetime64[ns](1), float64(9), int32(3), object(8)
```

Data Imputation:

```
df1["HS_CONN"].fillna("UNKNOWN",inplace=True)
df1["USIM_TAG"].fillna("N",inplace=True)
```

```
Command took 0.07 seconds -- by aad693@uregina.ca at 8/14/2023, 7:35:04 PM on Forhad Azam's Cluster
```

Customer Profiling:

```
df=df.withColumn("Total_Rev",col("Voice Revenue")+col("Data Revenue"))
```

```
df: pyspark.sql.dataframe.DataFrame = [MONTH: string, MSISDN: integer ... 20 more fields]
Command took 0.18 seconds -- by aad693@uregina.ca at 8/14/2023, 9:08:33 PM on Forhad Azam's Cluster
```

```
Cmd 9
```

```
df=df.withColumn("Rev_Segment",expr("CASE when Total_Rev=0 then '0'\
                                     WHEN Total_Rev>0 AND Total_Rev <=100 then '>1-<=100'\
                                    WHEN Total_Rev>100 AND Total_Rev <=200 then '>100-<=200'\
                                    WHEN Total_Rev>200 AND Total_Rev <=350 then '>200-<=350'\
                                    WHEN Total_Rev>350 AND Total_Rev <=500 then '>350-<=500' else '>500' END"))
```

```
df: pyspark.sql.dataframe.DataFrame = [MONTH: string, MSISDN: integer ... 21 more fields]
Command took 0.24 seconds -- by aad693@uregina.ca at 8/14/2023, 9:08:38 PM on Forhad Azam's Cluster
```

```
Cmd 10
```

```
df=df.withColumn("RG_Customer",expr("CASE when Total_Rev=0 then 'N'\
                                    else 'Y' END"))
```

```
df: pyspark.sql.dataframe.DataFrame = [MONTH: string, MSISDN: integer ... 22 more fields]
Command took 0.16 seconds -- by aad693@uregina.ca at 8/14/2023, 9:10:30 PM on Forhad Azam's Cluster
```

Saving Data Frame to Azure Data Lake:

```
output_path="/mnt/transformed/temp/"

# Write CSV file with column header (column names)
df.coalesce(1).write.option("header",True).format("csv").mode("overwrite").save(output_path)
```

```
Cmd 14
```

```
files=dbutils.fs.ls(output_path)
name=""
for f in files:
    if f.name.endswith(".csv"):
        name=f.name

dbutils.fs.cp(output_path+name,"/mnt/transformed/customer_rev_tr.csv")
```

Query For Table Creation in Azure SQL:

```
CREATE TABLE CUSTOMER_REV (
    [MONTH] NVARCHAR(50),
    [MSISDN] INT,
    [BSCODE] INT,
    [PRP_PST_TAG] NVARCHAR(50),
    [FIRST_CONNECTION_DATE] DATE NOT NULL,
    [Activation Status] NVARCHAR(50),
    [PRODUCT] NVARCHAR(50),
    [PP_GROUP] NVARCHAR(50),
    [NOT_USING_TILL_DAYS] NVARCHAR(50),
    [Voice Revenue] FLOAT,
    [Data Revenue] FLOAT,
    [SMS Revenue] FLOAT,
    [MMS Revenue] INT,
    [VAS Revenue] FLOAT,
    [Mix Bundle Revenue] FLOAT,
    [Vocie On Min] FLOAT,
    [Voice OFF Min] FLOAT,
    [Voice Out Min] FLOAT,
    [MB_USAGE] FLOAT,
    [USIM_TAG] NVARCHAR(50),
    [HS_CONN] NVARCHAR(50)
);
```