Read Me

List of Files
Task 1. Dataengineering : "**1. Data Engineering solution.pdf**" code with instructions as pdf
"**1. Data Engineering.ipynb**" actual code with instructions
Task 2. Prediction Cancellation: "**2. Logistic_Regression Model. pdf** " code with instructions as pdf
"**2. Logistic_Regression Model.ipynb**" actual code with instructions

Result of Task-1 send all data to CSV: "**Results.csv**"

Result of Task-1 create DB and DB Schema: "**SQLQuery3.sql**" – this is just to show that a DB was created when code ran in python

Input file for Task-2 : "**dataframe.csv**"

Author: Wish MK Natarajan

Software's Required

- **1. Data Engineering**

- Python3.8 or higher
- Anaconda Navigator – Jupyter Notebook (uses .ipynb format)
- Microsoft SQL server
- Microsoft SQL server management studio

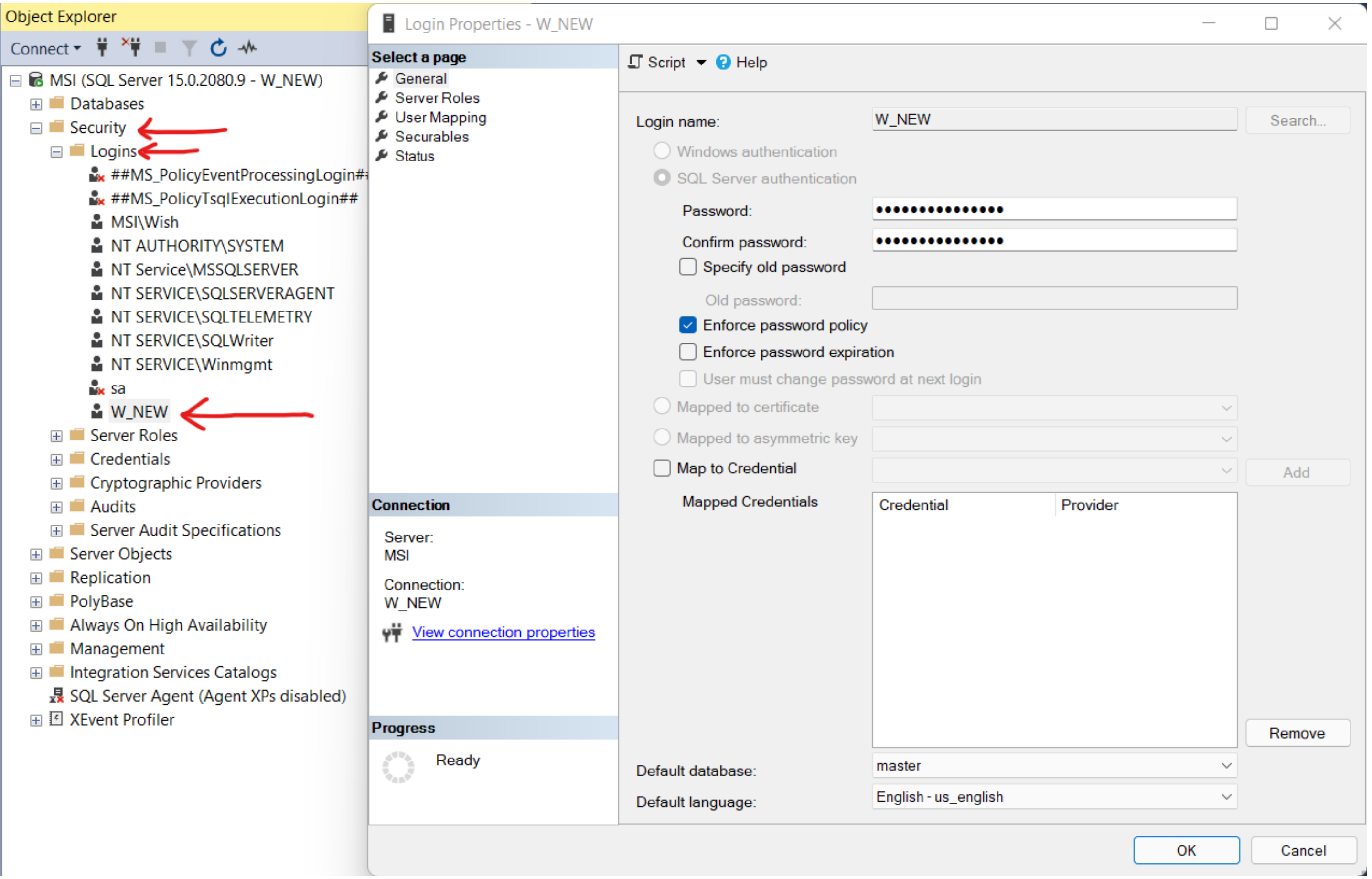
- **2. Predict Cancellations**

- Python
- Anaconda Navigator – Jupyter Notebook (uses .ipynb format)

Note: Install this to run the code

Prerequisites for Microsoft SQL Server and Server management Studio

1. Create new SQL server login in Server management studio give all permissions



Object Explorer

Connect ▾

- MSI (SQL Server 15.0.2080.9 - W_NEW)
  - Databases
  - Security
    - Logins
      - ##MS_PolicyEventProcessingLogin##
      - ##MS_PolicyTsqlExecutionLogin##
      - MSI\Wish
      - NT AUTHORITY\SYSTEM
      - NT Service\MSSQLSERVER
      - NT SERVICE\SQLSERVERAGENT
      - NT SERVICE\SQLTELEMETRY
      - NT SERVICE\SQLWriter
      - NT SERVICE\Winmgmt
      - sa
      - W_NEW
    - Server Roles
    - Credentials
    - Cryptographic Providers
    - Audits
    - Server Audit Specifications
  - Server Objects
  - Replication
  - PolyBase
  - Always On High Availability
  - Management
  - Integration Services Catalogs
  - SQL Server Agent (Agent XPs disabled)
  - XEvent Profiler

Login Properties - W_NEW

Select a page: General, Server Roles, User Mapping, Securables, Status

Script ▾ Help

Login name: W_NEW Search...

○ Windows authentication

● SQL Server authentication

Password: •••••••••••••••

Confirm password: •••••••••••••••

□ Specify old password

Old password:

☑ Enforce password policy

□ Enforce password expiration

□ User must change password at next login

○ Mapped to certificate

○ Mapped to asymmetric key

□ Map to Credential Add

Mapped Credentials

Credential	Provider

Remove

Connection

Server:
MSI

Connection:
W_NEW

View connection properties

Progress

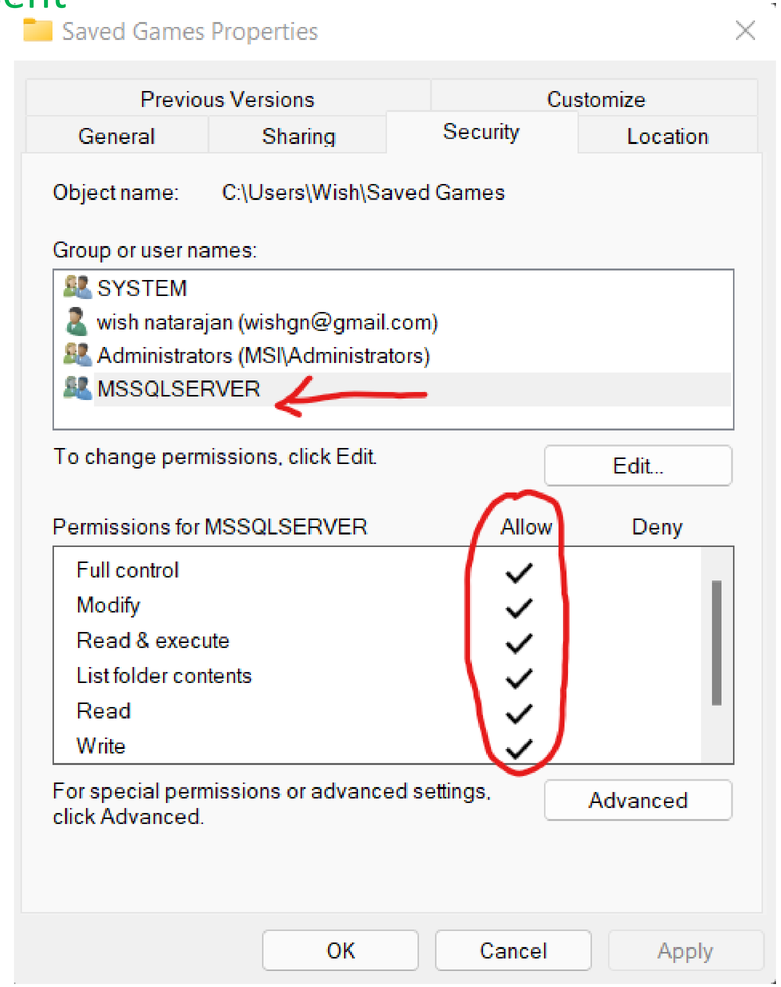
Ready

Default database: master

Default language: English - us_english

OK Cancel

2. Give access to sql server to access folder where csv is present



Saved Games Properties

Previous Versions | Customize | General | Sharing | Security | Location

Object name: C:\Users\Wish\Saved Games

Group or user names:

- SYSTEM
- wish natarajan (wishgn@gmail.com)
- Administrators (MSI\Administrators)
- MSSQLSERVER

To change permissions, click Edit. Edit...

Permissions for MSSQLSERVER	Allow	Deny
Full control	✓	
Modify	✓	
Read & execute	✓	
List folder contents	✓	
Read	✓	
Write	✓	

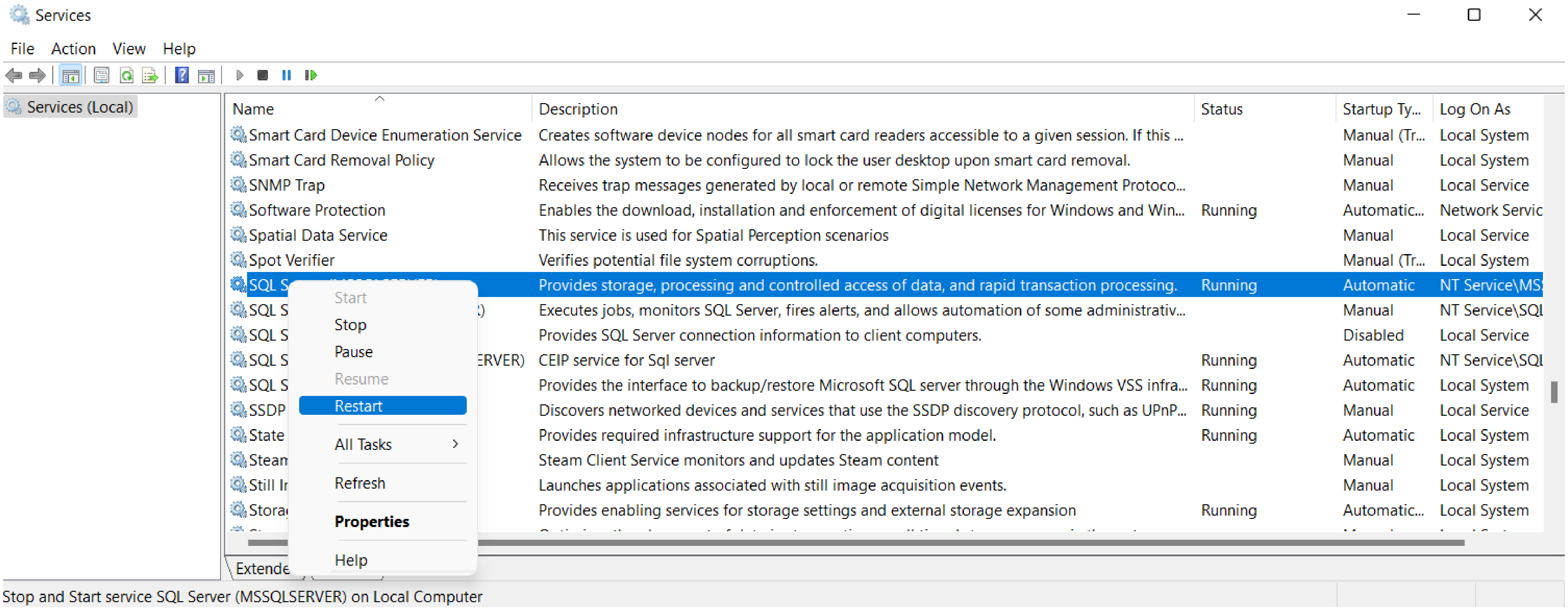
For special permissions or advanced settings, click Advanced. Advanced

OK Cancel Apply

Note: Required before run of SQL code in python

Prerequisites for Microsoft SQL Server and Server management Studio

- 3. Once the previous steps are finished restart the SQL server in services to run the code (Press: Windows key and type ‘services’ then press enter)



Name	Description	Status	Startup Ty...	Log On As
Smart Card Device Enumeration Service	Creates software device nodes for all smart card readers accessible to a given session. If this ...		Manual (Tr...	Local System
Smart Card Removal Policy	Allows the system to be configured to lock the user desktop upon smart card removal.		Manual	Local System
SNMP Trap	Receives trap messages generated by local or remote Simple Network Management Protoco...		Manual	Local Service
Software Protection	Enables the download, installation and enforcement of digital licenses for Windows and Win...	Running	Automatic...	Network Servic
Spatial Data Service	This service is used for Spatial Perception scenarios		Manual	Local Service
Spot Verifier	Verifies potential file system corruptions.		Manual (Tr...	Local System
SQL S	Provides storage, processing and controlled access of data, and rapid transaction processing.	Running	Automatic	NT Service\MS
SQL S)	Executes jobs, monitors SQL Server, fires alerts, and allows automation of some administrativ...		Manual	NT Service\SQL
SQL S	Provides SQL Server connection information to client computers.		Disabled	Local Service
SQL S ERVER)	CEIP service for Sql server	Running	Automatic	NT Service\SQL
SQL S	Provides the interface to backup/restore Microsoft SQL server through the Windows VSS infra...	Running	Automatic	Local System
SSDP	Discovers networked devices and services that use the SSDP discovery protocol, such as UPnP...	Running	Manual	Local Service
State	Provides required infrastructure support for the application model.	Running	Automatic	Local System
Steam	Steam Client Service monitors and updates Steam content		Manual	Local System
Still I	Launches applications associated with still image acquisition events.		Manual	Local System
Stora	Provides enabling services for storage settings and external storage expansion	Running	Automatic...	Local System

Stop and Start service SQL Server (MSSQLSERVER) on Local Computer

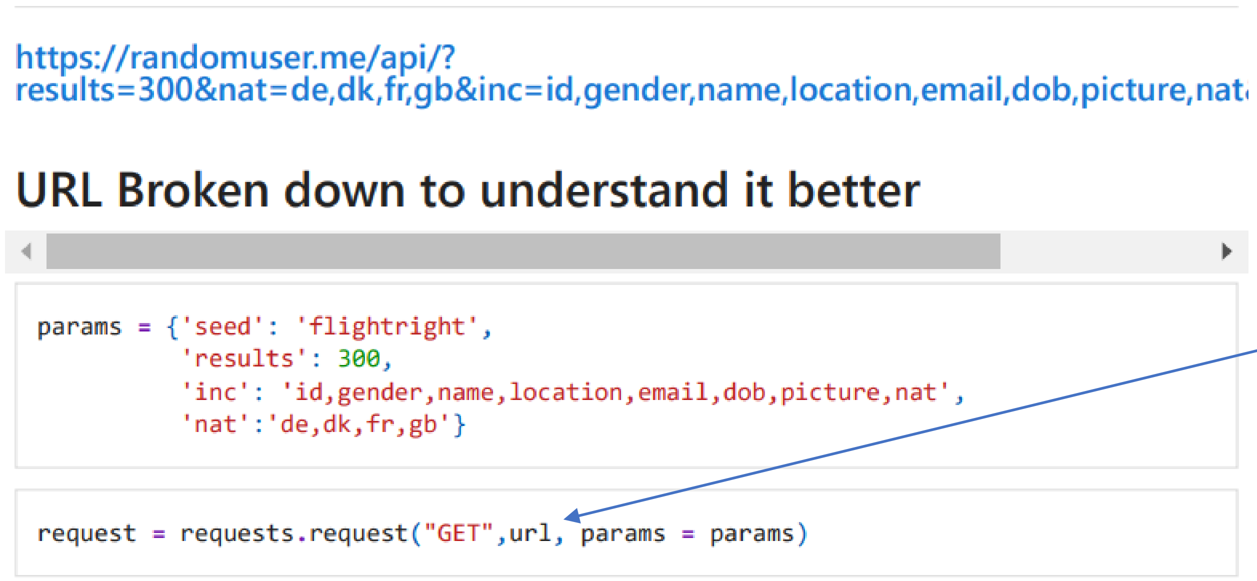
4. Once this is done run code for Database creation, table creation, running sql quering all of this on the Python Jupyter Notebook

Task 1 Data Engineering

1.1 Data pull from rest api

- Enhancement or feature added:

- Did not use direct url as to long and complex to understand, instead divided it into mini url and parameters



https://randomuser.me/api/?results=300&nat=de,dk,fr,gb&inc=id,gender,name,location,email,dob,picture,nat

URL Broken down to understand it better

```
params = {'seed': 'flightright',
          'results': 300,
          'inc': 'id,gender,name,location,email,dob,picture,nat',
          'nat':'de,dk,fr,gb'}
```

```
request = requests.request("GET",url, params = params)
```

url = "https://randomuser.me/api/"

- **1.2 All data to CSV**

- Normalized the .json file to get clean dataframe.
- Enhancements or feature added:
- Gave clean column names for ease of reading
- Data transfer to CSV file: requires header of dataframe

- Can do more here:
- Eg: date of birth column: DOB, has date+time+extra_text as wrong date format data. Can clean it to extract only date.
- Make composite primary keys, indexes etc to make data access faster in table

- Eg of DDL statements used:

- Simple CREATE and ALTER statements used. For database and table creations

Q. Why Designed Database?

A.

- To store historical data for further analysis or report creation in future.

Helps support and ensure the accuracy and integrity of your information

Divides your information into subject-based tables to reduce redundant data.

- 1.3 DDL to create hypothetical database schema

- Simple CREATE and ALTER statements used
- Ideal query for table would be: (if all data formats were clear and as required)
- CREATE TABLE flightright_data (Gender varchar(8), Email varchar(250), Nat varchar(4), Title varchar(7), First_Name varchar(250), Last_Name varchar(250), Str_Number int, Str_Name varchar(250), City varchar(250), State varchar(250), Country varchar(250), Postcode int, Latitude varchar(50), Longitude varchar(50), Time_Zone_Offset varchar(20), Time_Zone_Discription varchar(250), DOB date, Age int, ID_Name varchar(250), ID_Value varchar(250))

- Based on current extracted data below used : (Highly generalized)
- CREATE TABLE flightright_data (Gender varchar(10), Email varchar(250), Nat varchar(5), Title varchar(10), First_Name varchar(250), Last_Name varchar(250), Str_Number varchar(250), Str_Name varchar(250), City varchar(250), State varchar(250), Country varchar(250), Postcode varchar(250), Latitude varchar(250), Longitude varchar(250), Time_Zone_Offset varchar(250), Time_Zone_Discription varchar(250), DOB varchar(250), Age varchar(10), ID_Name varchar(250), ID_Value varchar(250))

As header in dataframe to csv gets extra row (Column Names) that needs to be deleted after table records insertion

See code for better understanding (with pictures)

Code understanding and uses

- **Open Connection to sql server management studio**
- conn = pyodbc.connect("driver_details;sql_server_details;Usr_Id_details;Passwrd_details", autocommit = True)

- **Cursor** **"This acts something like a pointer to database you are currently working in"**
- Cursor = conn.cursor()

- **Queries are named as**
- sq_cmd_<n> :
- Where <n> is 1,2,3,4,5…. Infinity

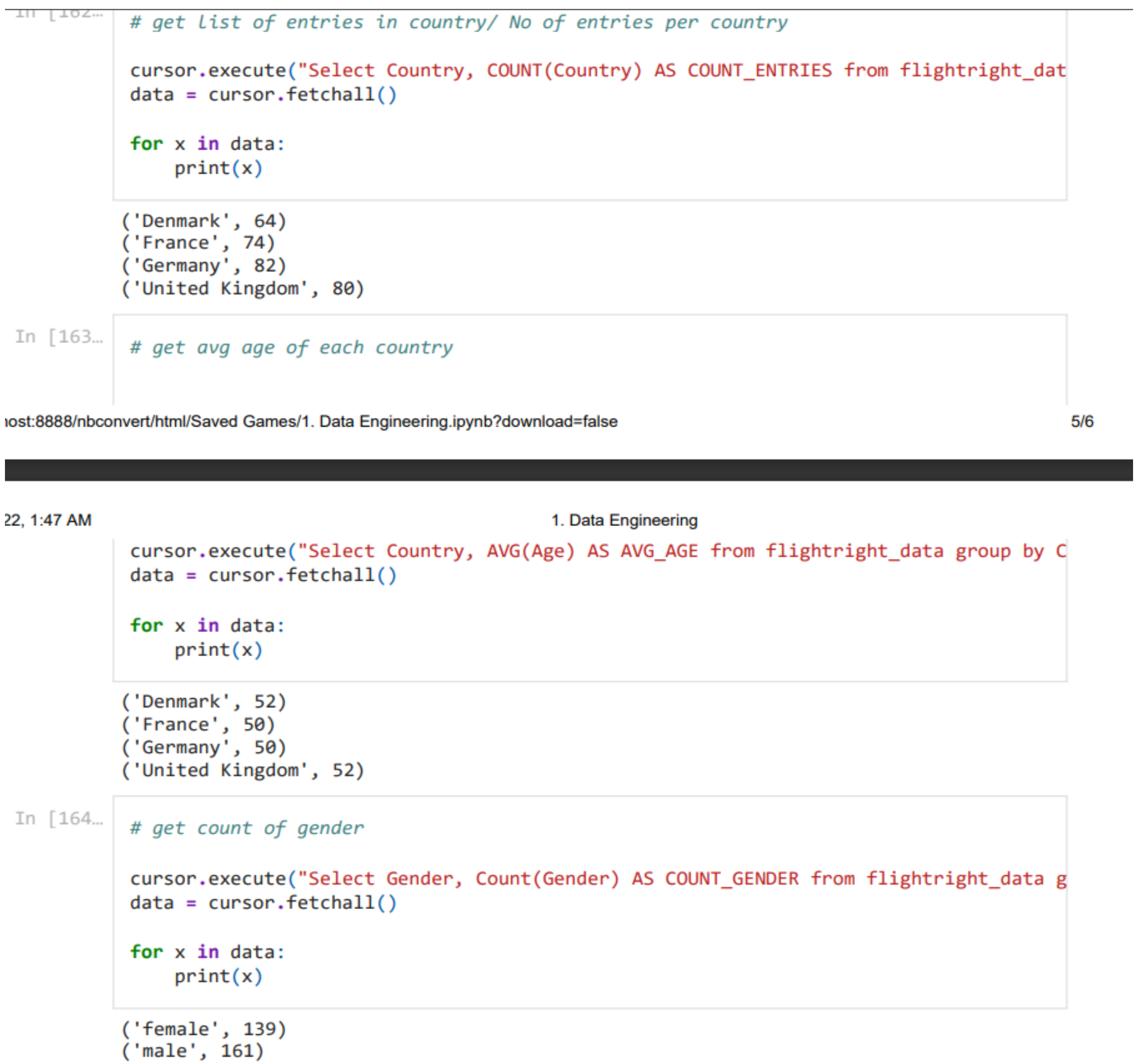
- **Query executed with the below command**:
- cursor.execute(sq_cmd_<n>)

- 1.4 Generate Readable statistics for data

Can improve here : Display data on PowerBi or Tableau Dashboard

Note: For now only showing simple text output

Output In Code



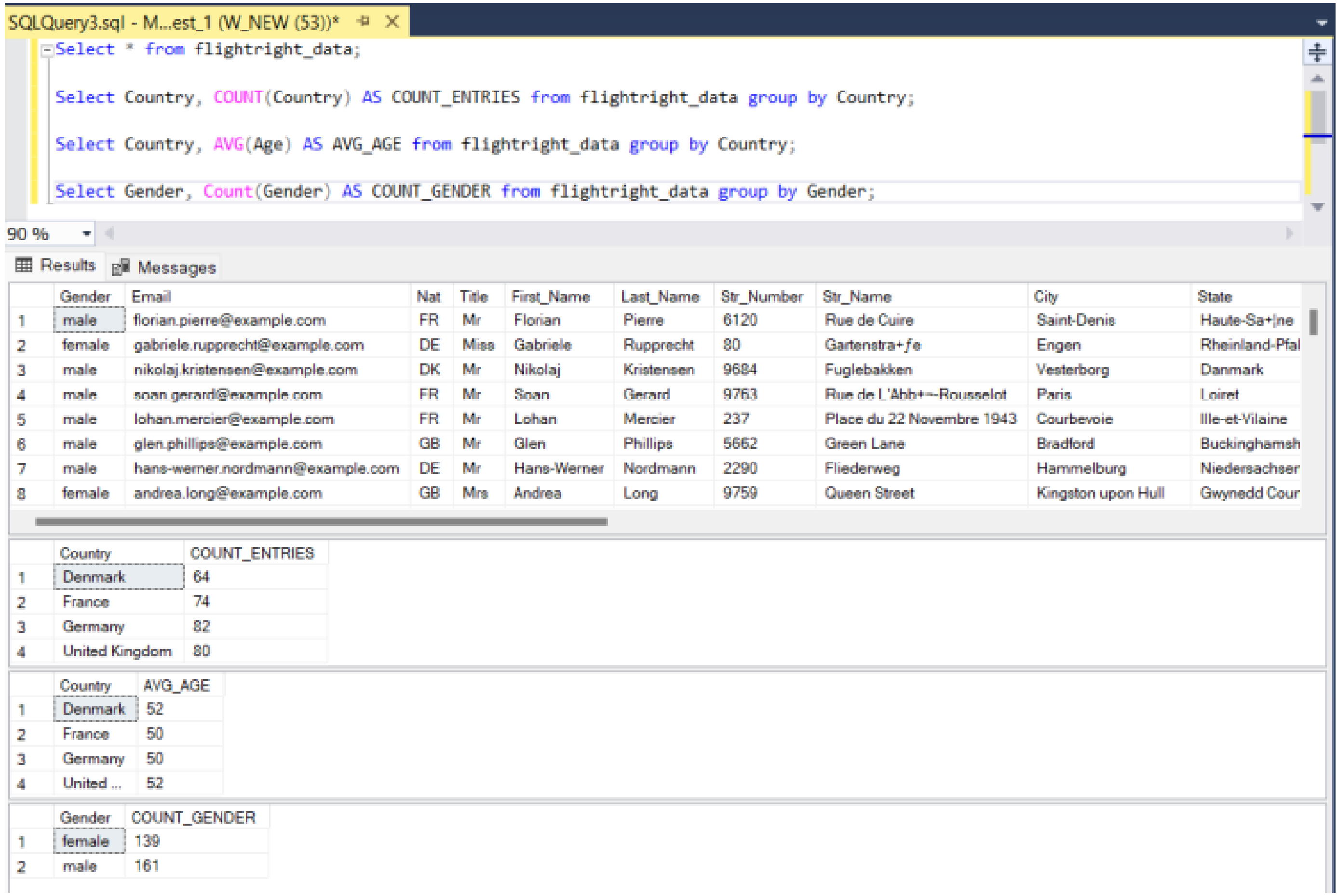
```
In [162…  # get list of entries in country/ No of entries per country

          cursor.execute("Select Country, COUNT(Country) AS COUNT_ENTRIES from flightright_dat
          data = cursor.fetchall()

          for x in data:
              print(x)

         ('Denmark', 64)
         ('France', 74)
         ('Germany', 82)
         ('United Kingdom', 80)

In [163…  # get avg age of each country
```

ost:8888/nbconvert/html/Saved Games/1. Data Engineering.ipynb?download=false 5/6

22, 1:47 AM 1. Data Engineering

```
          cursor.execute("Select Country, AVG(Age) AS AVG_AGE from flightright_data group by C
          data = cursor.fetchall()

          for x in data:
              print(x)

         ('Denmark', 52)
         ('France', 50)
         ('Germany', 50)
         ('United Kingdom', 52)

In [164…  # get count of gender

          cursor.execute("Select Gender, Count(Gender) AS COUNT_GENDER from flightright_data g
          data = cursor.fetchall()

          for x in data:
              print(x)

         ('female', 139)
         ('male', 161)
```

Output In SQL server Management Studio



SQLQuery3.sql - M...est_1 (W_NEW (53))*

```sql
Select * from flightright_data;

Select Country, COUNT(Country) AS COUNT_ENTRIES from flightright_data group by Country;

Select Country, AVG(Age) AS AVG_AGE from flightright_data group by Country;

Select Gender, Count(Gender) AS COUNT_GENDER from flightright_data group by Gender;
```

	Gender	Email	Nat	Title	First_Name	Last_Name	Str_Number	Str_Name	City	State
1	male	florian.pierre@example.com	FR	Mr	Florian	Pierre	6120	Rue de Cuire	Saint-Denis	Haute-Sa+|ne
2	female	gabriele.rupprecht@example.com	DE	Miss	Gabriele	Rupprecht	80	Gartenstra+ƒe	Engen	Rheinland-Pfal
3	male	nikolaj.kristensen@example.com	DK	Mr	Nikolaj	Kristensen	9684	Fuglebakken	Vesterborg	Danmark
4	male	soan.gerard@example.com	FR	Mr	Soan	Gerard	9763	Rue de L'Abb+--Rousselot	Paris	Loiret
5	male	lohan.mercier@example.com	FR	Mr	Lohan	Mercier	237	Place du 22 Novembre 1943	Courbevoie	Ille-et-Vilaine
6	male	glen.phillips@example.com	GB	Mr	Glen	Phillips	5662	Green Lane	Bradford	Buckinghamsh
7	male	hans-werner.nordmann@example.com	DE	Mr	Hans-Werner	Nordmann	2290	Fliederweg	Hammelburg	Niedersachser
8	female	andrea.long@example.com	GB	Mrs	Andrea	Long	9759	Queen Street	Kingston upon Hull	Gwynedd Cour

	Country	COUNT_ENTRIES
1	Denmark	64
2	France	74
3	Germany	82
4	United Kingdom	80

	Country	AVG_AGE
1	Denmark	52
2	France	50
3	Germany	50
4	United ...	52

	Gender	COUNT_GENDER
1	female	139
2	male	161

Note: Most of code description is in code file itself

2 Predict Cancellations

- **Flow**:
- Install pyspark **>**
- Create spark entry point **>**
- Import csv datafile **>**
- PrintSchema (see table properties) **>**
- VectorAssembler to create features and Labels **>**
- Divide dataset into 70% train and 30% Test **>**
- LogisticRegression Model and its parameters **>**
- BinaryClassificationEvaluator ‘0’ or ‘1’ **>**
- Evaluate **>**
- Get AUC value **>**
- Get new dataset to work on without label column and predict the label column using existing model

- Recommendations:
- Tableau or PowerBi or any dashboard tools for data visualization would be better.

Here in the task, I have only used text represent for simplicity

Thank You