In this segment you will create a pipeline to automate data cleaning using Airflow. For creating the pipeline follow the below mentioned instructions:

1. Go to the **Assignment/02\_training\_pipeline**/scripts folder.

2. First we will break down all the data cleaning tasks **into functions in the utils.py** file you have been provided with functions level breakdown of all the tasks. You can do that by simply **taking appropriate code from the model experimentation notebook named “lead\_scoring\_experimentation.ipynb”** that you created in the “Assignment/02\_training\_pipeline/notebooks” folder.

3. Once you have created the code into the respective functions. You will **modify the function to read their input data and write their output to the database present in file 'lead\_scoring\_data\_cleaning.db'** which was created in the data pipeline. The function will be executed in the following order so make sure to read the input for the **functions appropriately**

encode\_features -> get\_trained\_model

**Set the mlflow experiment name** to “Lead\_scoring\_mlflow\_production”.

NOTE: We need to clean our new data before we can use it to train our model. Thus we will run the data pipeline on the raw data and use the cleaned data for training. Thus **our encode\_feature will read the input from data pipeline’s output which is present in the ‘lead\_scoring\_data\_cleaning.db’ file**. Also note that since we are not creating any db here we do not require build\_db function

4.Once you have modified the functions, create a test notebook and import utils file in it and run all the functions in the proper order. This will help you **debug your code**

NOTE : **Before you run the get\_trained\_model function ensure that you have run the “mlflow server” command** from the terminal at the appropriate path. Use the mlflow db name as 'Lead\_scoring\_mlflow\_production.db' . Lead\_scoring\_mlflow\_production.db is the database that is used for experiment tracking in the training pipeline. This is different from the ‘lead\_scoring\_model\_experimentation.db’ which was used to track all the model experimentation. Both of them are different from 'lead\_scoring\_data\_cleaning.db’ which is used for data cleaning in the data pipeline.

5. Once you have ensured that the functions in utils.py file are working as intended, **store all the constant values in constants.py** and repeat step 4 to ensure that the code is working properly. Some of the variables’ name are mentioned in the constants file which you can use. You can also create your own additional variable if required.

6. Now **create a pipeline using airflow in the 'lead\_scoring\_training\_pipeline.py'**. Follow the instructions present in the file for creating the dag for airflow.

7.Now go to the '~/airflow/dags/' folder and create a folder named 'Lead\_scoring\_training\_pipeline' in it and copy paste the following files in it.The following file should be present in Lead\_scoring\_training\_pipeline

├── lead\_scoring\_training\_pipeline.py

├── constants.py

└── utils.py

8. After **copying all the necessary files and folder from 'scripts' folder to 'Lead\_scoring\_training\_pipeline' and modify your constants, utils and lead\_scoring\_training\_pipeline.py** as the paths have been changed when you have changed the directory. Make **changes in the file path names in constants.py file**.

9. With this you should be able to run the training pipeline. Follow the instructions mentioned in the Airflow module to **run the Training pipeline and take the screenshot of Airflow UI for submission. Trigger the training pipeline manually**

NOTE : Before you trigger the training pipeline ensure that you have run mlflow server command from the terminal at the appropriate path. Use the mlflow db name as 'Lead\_scoring\_mlflow\_production' and set the name of experiment as 'Lead\_scoring\_mlflow\_production'