BI

Mini Project   
Flight Price Prediction Analysis

**SUMBITTED BY -**

**SPRINT 4 -**

**Sandhya M**

**Kanchan Vinod Kumawat**

**Priyanshi Shukla**

**Sayali Anil Katolkar**

**Aditi Mahendra Sonawane**

**Bhumika Yadav**

**1. Table of Contents**

1. Introduction
   1. Setup Checklist for Mini Project
   2. Instructions
2. Problem Statement
   1. Objective
   2. Abstract of the project
   3. Technology used:
3. **Execution And Result**
4. **Conclusion**

**INTRODUCTION**

The objective of the study is to Analyse the flight booking dataset obtained from “Ease My Trip” website and to conduct various statistical hypothesis tests to get meaningful information from it.

The 'Linear Regression' statistical algorithm would be used to train the dataset and predict a continuous target variable. 'Easemytrip' is an internet platform for booking flight tickets, and hence a platform that potential passengers use to buy tickets.

A thorough study of the data will aid in the discovery of valuable insights that will be of enormous value to passengers

## SETUP CHECKLIST FOR MINI PROJECT

### Minimum System Requirements

* + - Intel Pentium 90 or higher (P166 recommended)
    - Microsoft Windows 2010 or above.
    - Memory: 4GB of RAM (4GB or more recommended)
    - Internet Explorer 10.0 or higher
    - Snowflake and Cloud(AWS/Azure) access

# 2.PROBLEM STATEMENT

## OBJECTIVE

Development of an Analysis to predict the price of the flight

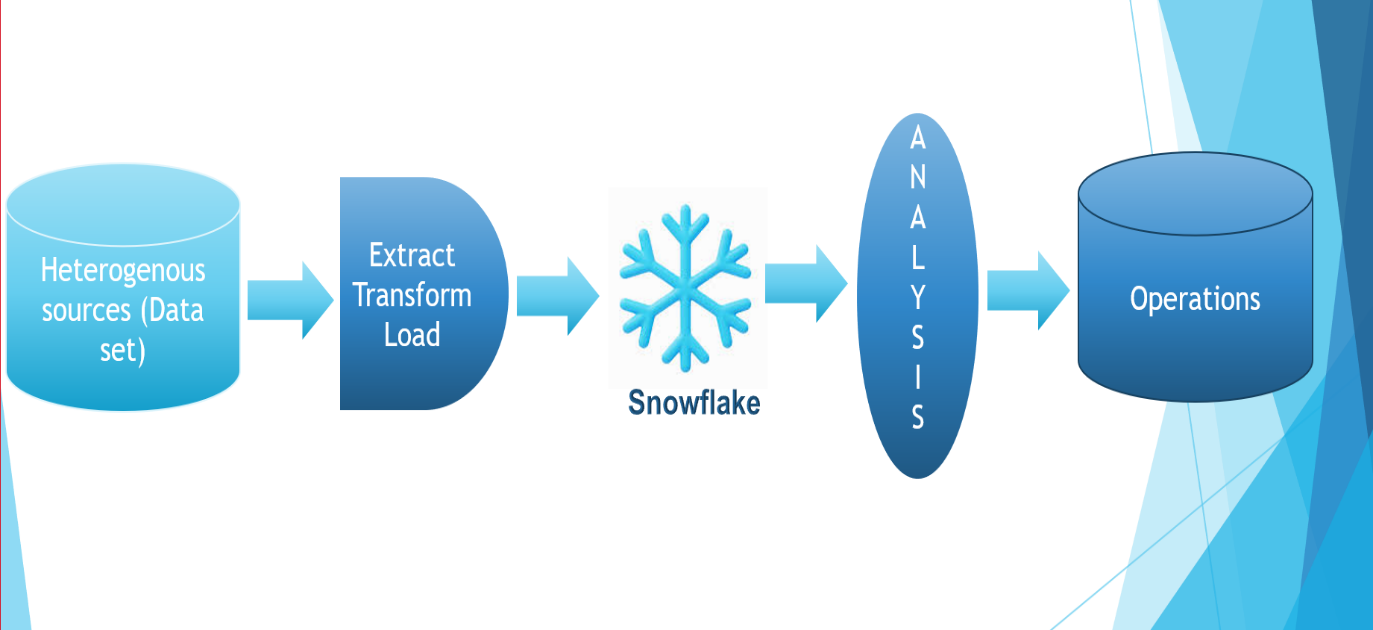
## ABSTRACT OF THE PROJECT

The project is to develop Analysis to predict the price of the flight.

## TECHNOLOGY USED:

* + - Snowflake
    - Cloud (AWS)

**Project Flow:**



## Execution :

**CODE : PYTHON SCRIPT :**

import pandas as pd

chunk\_size = 42879

batch\_no = 1

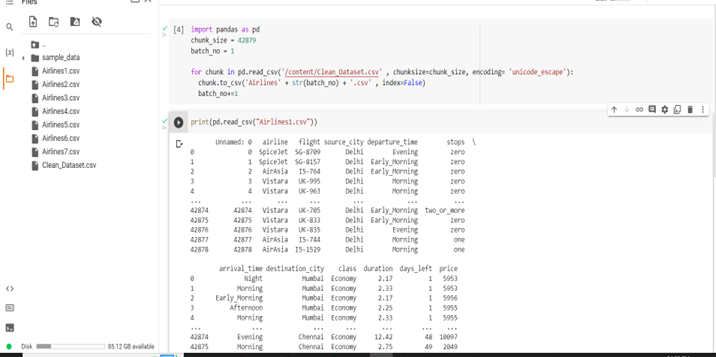
for chunk in pd.read\_csv('/content/Clean\_Dataset.csv' , chunksize=chunk\_size, encoding= 'unicode\_escape'):

  chunk.to\_csv('Airlines' + str(batch\_no) + '.csv' , index=False)

  batch\_no+=1

**Result:-**

Graphical user interface, text

Description automatically generated with medium confidence

**Fig : Splitting into multiple files using python script**

**SNOWFLAKE :**

**-------CREATING DATABASE---------**

create database sprint4;

**---------AWS INTEGRATION EXTERNAL STAGE----------**

create or replace storage integration s3\_int

type = external\_stage

storage\_provider = s3

enabled = true

storage\_aws\_role\_arn = 'arn:aws:iam::242650538488:role/sprint-4-role'

storage\_allowed\_locations = ('s3://sprint-4-bucket/');

DESC INTEGRATION s3\_int;

**-----CREATE FILE FORMAT------**

create or replace file format my\_csv\_format

type = csv field\_delimiter = ',' skip\_header = 1 null\_if = ('NULL', 'null') empty\_field\_as\_null = true;

desc file format my\_csv\_format;

**----STAGE CREATION--------**

create or replace stage my\_s3\_stage

storage\_integration = s3\_int

url = 's3://sprint-4-bucket/'

file\_format = my\_csv\_format;

List @my\_s3\_stage;

select t.$1 as row\_id , t.$2 as airline , t.$3 as flight , t.$4 as source\_city , t.$5 as departure\_time , t.$6 as stops ,

t.$7 as arrival\_time , t.$8 as destination\_city , t.$9 as class , t.$10 as duration , t.$11 as days\_left , t.$12 as price

from @my\_s3\_stage/ t;

**-----CREATING TABLE------**

create or replace table mytable2

(row\_id number , airline string , flight varchar , source\_city string ,

departure\_time string , stops string , arrival\_time string , destination\_city string , class string ,

duration varchar , days\_left number , price number);

copy into mytable2

from @my\_s3\_stage

file\_format = (type = csv field\_optionally\_enclosed\_by='"',skip\_header=1)

pattern = '.\*.csv'

on\_error = 'skip\_file';

select \* from mytable2;

**Result:-**

Graphical user interface, table

Description automatically generated

Graphical user interface, application, table

Description automatically generated

**Fig : Loading data into the table using External Stage in Snowflake**

**------CREATING TASK MYTASK---**

CREATE or replace TASK mytask

WAREHOUSE = my\_wh

SCHEDULE = 'Using CRON 0 0 \* \* THU Asia/Kolkata'

TIMESTAMP\_INPUT\_FORMAT = 'YYYY-MM-DD HH24'

AS

copy into mytable2

from (select t.$1,t.$2,t.$3,t.$4,t.$5,t.$6,t.$7,t.$8,t.$9,t.$10,t.$11,t.$12 from @my\_s3\_stage/ t)

file\_format = (type = csv field\_optionally\_enclosed\_by='"')

pattern = '.\*.csv'

on\_error = 'skip\_file';

SHOW TASKS

alter task mytask resume;

alter task mytask suspend;

**------CREATING SNOWPIPE--**

create or replace pipe snowpipe auto\_ingest=true as

copy into mytable2

from @my\_s3\_stage

file\_format = (type = csv field\_optionally\_enclosed\_by='"')

pattern = '.\*.csv'

on\_error = 'skip\_file';

show pipes;

alter pipe snowpipe refresh;

select \* from mytable2;

select SYSTEM$PIPE\_STATUS('snowpipe')

select \* from table(validate\_pipe\_load(

pipe\_name=>'sprint4.PUBLIC.snowpipe',

start\_time=>dateadd(minute, -4, current\_timestamp())));

select \*

from table(information\_schema.copy\_history(table\_name=>'mytable2', start\_time=> dateadd(minute, -4, current\_timestamp())));

create or replace stream sprint4 on table mytable2;

select \* from sprint4;

**-----CREATING TABLE TARGER\_T-----**

create or replace

table target\_t(row\_id number , airline string , flight varchar , source\_city string ,

departure\_time string , stops string , arrival\_time string , destination\_city string , class string ,

duration varchar , days\_left number , price number,stream\_type string default null, rec\_version number default 0, REC\_DATE TIMESTAMP\_LTZ);

--select \* from tgt\_merge;

**-----CREATING TASK TGT\_MERGE-----**

CREATE or replace TASK tgt\_merge

WAREHOUSE = my\_wh

SCHEDULE = '1 minute'

WHEN

SYSTEM$STREAM\_HAS\_DATA('sprint4')

AS

merge into target\_t t

using sprint4 s

on t.row\_id=s.row\_id and (metadata$action='DELETE')

when matched and metadata$isupdate='FALSE' then update set rec\_version=9999, stream\_type='DELETE'

when matched and metadata$isupdate='TRUE' then update set rec\_version=rec\_version-1, stream\_type='UPDATE'

when not matched then insert (row\_id , airline , flight , source\_city ,

departure\_time , stops , arrival\_time , destination\_city , class ,

duration , days\_left , price,stream\_type, rec\_version ,REC\_DATE)

values(s.row\_id , s.airline , s.flight , s.source\_city ,

s.departure\_time , s.stops , s.arrival\_time , s.destination\_city , s.class ,

s.duration , s.days\_left , s.price,metadata$action,0,CURRENT\_TIMESTAMP());

ALTER TASK tgt\_merge RESUME;

ALTER TASK mytask RESUME;

alter task tgt\_merge suspend;

show tasks

select \* from target\_t;

select \* from mytable2;

insert into mytable2 values (1, 'Indigo','SG-8709','Nagpur','Early\_Morning' , 'zero' , 'Night' , 'Mumbai' , 'Economy',

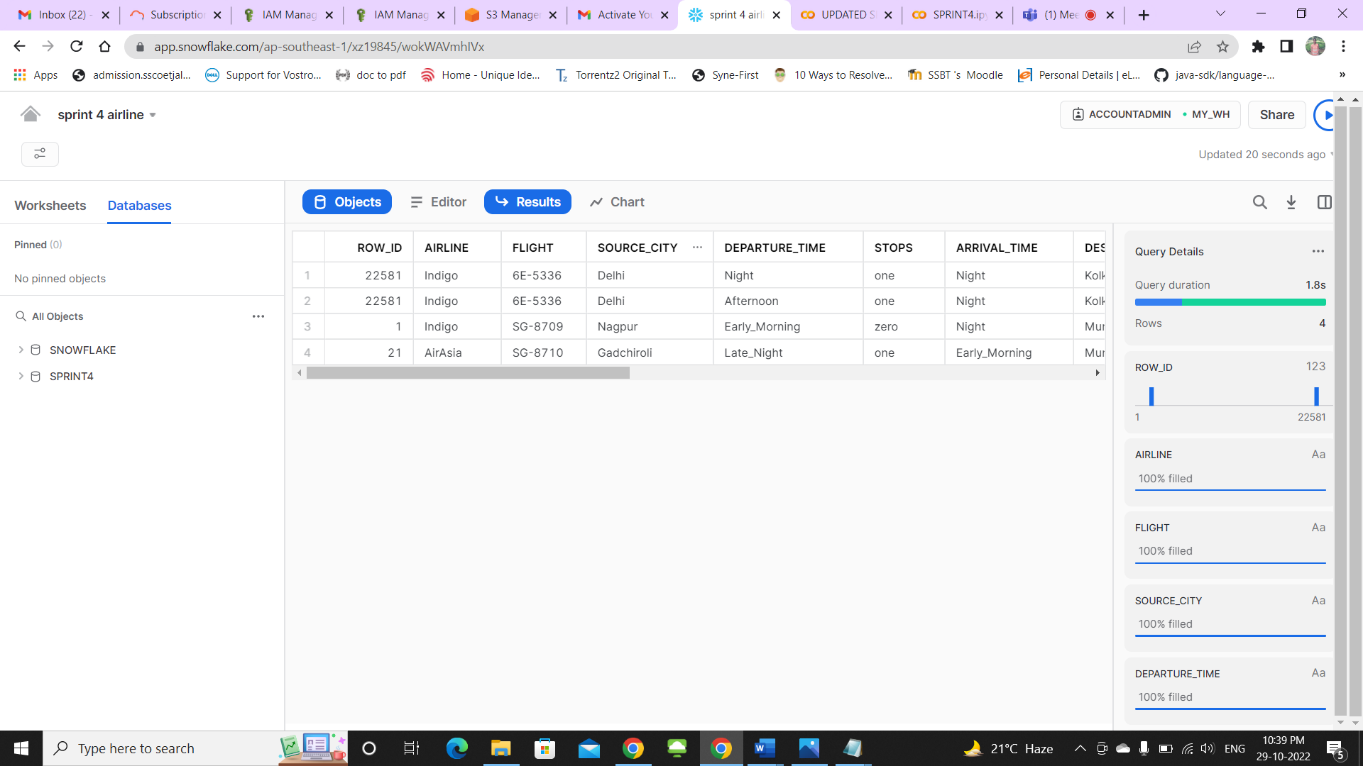
2.17 , 1 ,5953 );

delete from mytable2 where row\_id=1;

**Result:-**

A screenshot of a computer

Description automatically generatedText, table

Description automatically generated with medium confidence

**Fig : Slowly Changing Dimension (SCD 2)**

**----COLUMN LEVEL SECURITY---**

CREATE MASKING POLICY flight\_policy AS (VAL STRING) RETURNS STRING ->

CASE

WHEN CURRENT\_ROLE() IN ('FLIGHT') THEN VAL

ELSE '\*\*\*\*\*\*'

END;

create role flight;

CREATE OR REPLACE TABLE flight\_number( flight STRING MASKING POLICY flight\_policy,flight1 STRING)

INSERT INTO flight\_number(flight,flight1)

SELECT flight,flight FROM mytable2;

select current\_role();

select current\_user();

GRANT SELECT ON flight\_number TO ROLE flight;

grant usage on warehouse my\_wh to role flight;

grant usage on database sprint4 to role flight;

grant usage on schema public to role flight;

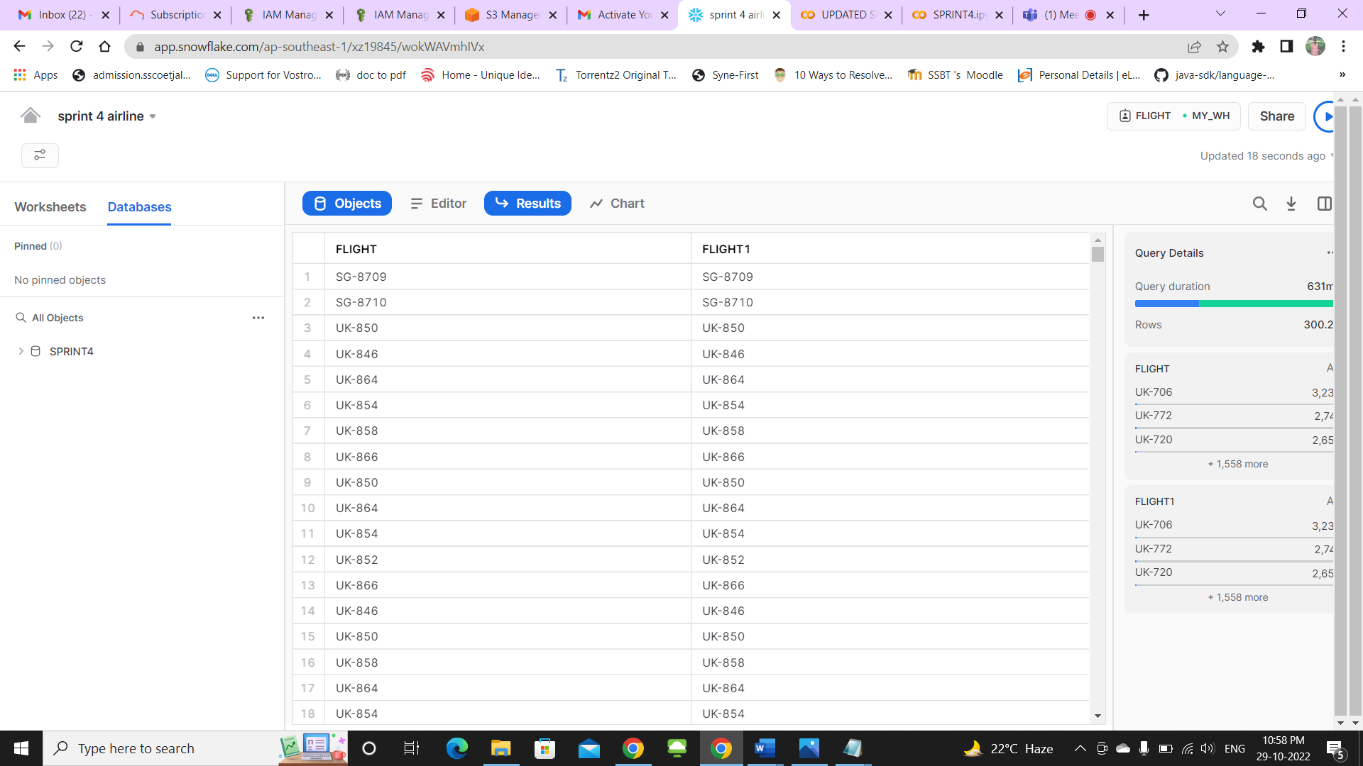
grant role flight to user ADITI0503;

select \* from flight\_number;

**Result:**

A screenshot of a computer

Description automatically generated



**Fig : Column Level Security**

**-------ROW LEVEL SECURITY----**

**-- Apply Row-level security using Secure Views**

**-- create a secure view**

**------STEP 1 : CREATION OF ROLE----**

create or replace role Spicejet;

create or replace role Vistara;

create or replace role Go\_First;

create or replace role Air\_India;

create or replace role AirAsia;

create or replace role Indigo;

create or replace user sayali password = 'temp123' default\_Role = 'Spicejet';

grant role Spicejet to user sayali;

create or replace user kanchan password = 'temp123' default\_Role = 'Vistara';

grant role Vistara to user kanchan;

create or replace user bhumika password = 'temp123' default\_Role = 'Go\_First';

grant role Go\_First to user bhumika;

create or replace user priyanshi password = 'temp123' default\_Role = 'Air\_India';

grant role Air\_India to user priyanshi;

create or replace user sandhya password = 'temp123' default\_Role = ' AirAsia';

grant role AirAsia to user sandhya;

create or replace user aditi password = 'temp123' default\_Role = 'Indigo';

grant role Indigo to user aditi;

**------STEP 2 : GRANT PRIVILEGES TO ROLES-----**

grant role SPICEJET to user ADITI0503;

grant role VISTARA to user ADITI0503;

grant role GO\_FIRST to user ADITI0503;

grant role AIR\_INDIA to user ADITI0503;

grant role AIRASIA to user ADITI0503;

grant role INDIGO to user ADITI0503;

grant usage on warehouse my\_wh to role AIRASIA;

grant usage on warehouse my\_wh to role AIR\_INDIA;

grant usage on warehouse my\_wh to role INDIGO;

grant usage on warehouse my\_wh to role VISTARA;

grant usage on warehouse my\_wh to role GO\_FIRST;

grant usage on warehouse my\_wh to role SPICEJET;

grant usage on database sprint4 to role AIRASIA;

grant usage on database sprint4 to role AIR\_INDIA;

grant usage on database sprint4 to role INDIGO;

grant usage on database sprint4 to role VISTARA;

grant usage on database sprint4 to role GO\_FIRST;

grant usage on database sprint4 to role SPICEJET;

grant usage on schema public to role AIRASIA;

grant usage on schema public to role AIR\_INDIA;

grant usage on schema public to role INDIGO;

grant usage on schema public to role VISTARA;

grant usage on schema public to role GO\_FIRST;

grant usage on schema public to role SPICEJET;

**----STEP 3 : CREATING SECURE VIEW----**

create or replace secure view vw\_airline as

select a.\*

from mytable2 a

where upper(a.airline) in (select upper(airline)

from airline\_rls r

where upper(airline) = upper(current\_role()));

select current\_role();

select \* from vw\_airline;

grant select on view vw\_airline to role AIRASIA;

grant select on view vw\_airline to role AIR\_INDIA;

grant select on view vw\_airline to role INDIGO;

grant select on view vw\_airline to role VISTARA;

grant select on view vw\_airline to role SPICEJET;

grant select on view vw\_airline to role GO\_FIRST;

CREATE OR REPLACE TABLE airline\_RLS( row\_id number, airline string);

INSERT INTO airline\_RLS(row\_id,airline)

SELECT row\_id,airline FROM mytable2;

select \* from airline\_rls;

**-----STEP 4 : VERIFY ROLES-----**

use role AIRASIA;

use database SPRINT4;

use schema PUBLIC;

select \* from vw\_airline;

use role AIR\_INDIA;

use database SPRINT4;

use schema PUBLIC;

select \* from vw\_airline;

use role INDIGO;

use database SPRINT4;

use schema PUBLIC;

select \* from vw\_airline;

use role VISTARA;

use database SPRINT4;

use schema PUBLIC;

select \* from vw\_airline;

use role GO\_FIRST;

use database SPRINT4;

use schema PUBLIC;

select \* from vw\_airline;

use role SPICEJET;

use database SPRINT4;

use schema PUBLIC;

select \* from vw\_airline;

**Result:**

Graphical user interface, application, table

Description automatically generated

Graphical user interface, application, table

Description automatically generated

**Fig : Row Level Security**

5. Conclusion

In this project, we learned to use and implement AWS S3 for creating Bucket, Policies, Role and Event notification.

we saw how to apply External staging , Integration , Column, Row Level Security and many more using Snowflake.