**DMML Assignment - Group 3**

**Group Member Names:**

1. A Rohit Sai - 2024aa05183

2. Boddupalli Venkata Naga Lokesh - 2024aa05360

3. Lingam Vinay Naga Pavan Kumar - 2024aa05364

Drive link : <https://drive.google.com/drive/folders/1uPqLmOFKMpDCZVdodDFt3XKwSEt4fSb-?usp=sharing>

**1. Problem Formulation**

**1.1 Business Problem Statement for Customer Churn Prediction**

Our business seeks to proactively identify customers at high risk of churning—those who may stop using our products or services in the near future—even though interventions could retain them. By analyzing customer data on demographics, behavior, and engagement, we aim to predict which customers are likely to leave and develop targeted retention strategies (e.g., personalized offers, improved support) to decrease churn rate, increase customer lifetime value, and maintain a profitable, loyal customer base.

* 1. **Key Business Objectives**

Reduce the addressable churn rate through timely interventions.

Increase customer retention and loyalty.

Boost revenue and overall business health by minimizing loss of customers.

Enable data-driven and targeted retention campaigns for at-risk customer segments.

**1.3 Key data sources and their attributes**

HuggingFace API - <https://datasets-server.huggingface.co/rows/scikit-learn/churn-prediction>

Kaggle API - <https://www.kaggle.com/api/v1/datasets/download/blastchar/telco-customer-churn>

**1.4 Expected Outputs from the Pipeline**

1. Machine learning-ready, transformed feature dataset.
2. Churn prediction model performance report (including metrics such as accuracy, recall, precision, ROC-AUC).
3. Well-organized and documented source code split into stages (ingestion, cleaning, engineering, modeling, evaluation, orchestration, reporting).

**1.5 measurable evaluation metrics**

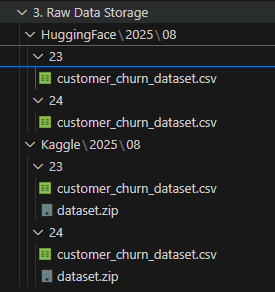
a. models are saved on each day when new training data is run and each version is saved so that we can revert anytime we want.

**2. Data Ingestion**

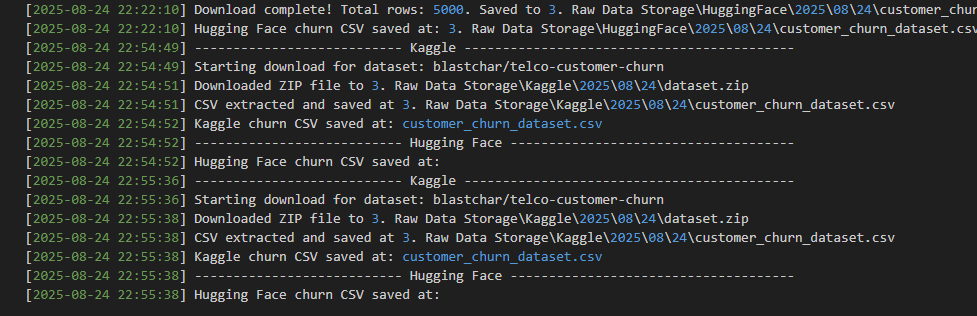
2 sources are identified – HuggingFace API and Kaggle API

Data from each source is downloaded everyday in respective folders in Raw Storage

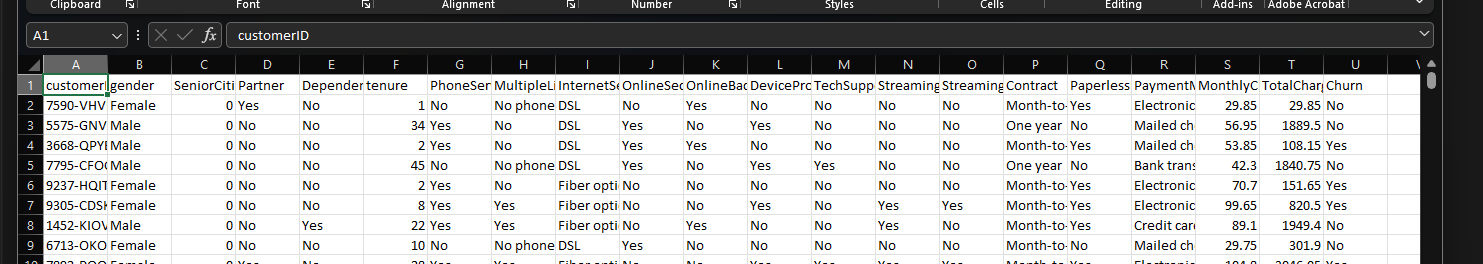
Screenshot:



Logs:



Data stored:



3. **Raw Data Storage**

Earlier ingested code is stored in local server in respective folders of APIs with time stamps.

Folder / bucket Structure

HuggingFace/{year}/{month}/{date}/churn\_customer\_csv

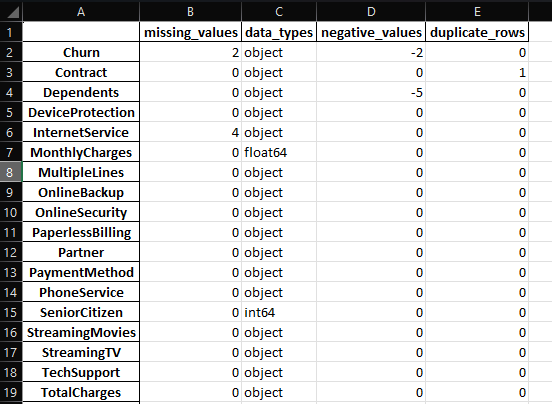
Kaggle/{year}/{month}/{date}/churn\_customer\_csv

1. **Data Validation**

Downloaded csv’s are validated against

* Duplicated rows
* Negative values
* Empty values

And respective validation\_csv are generated



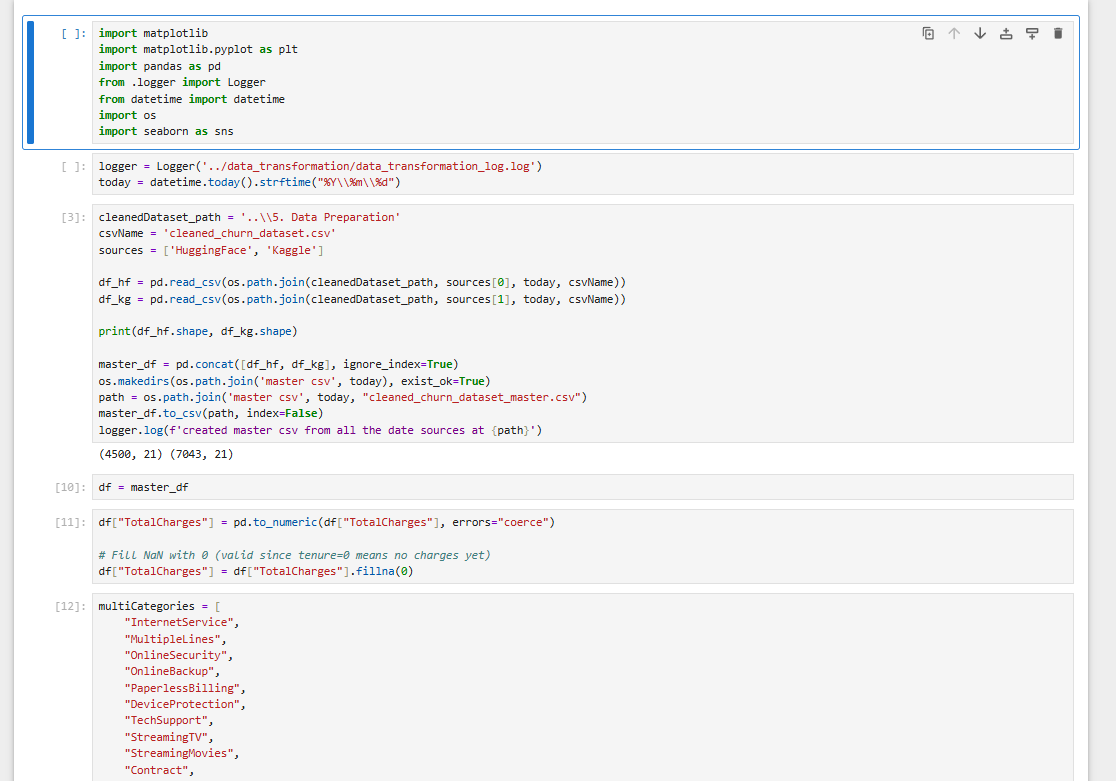
1. **Data Cleaning**

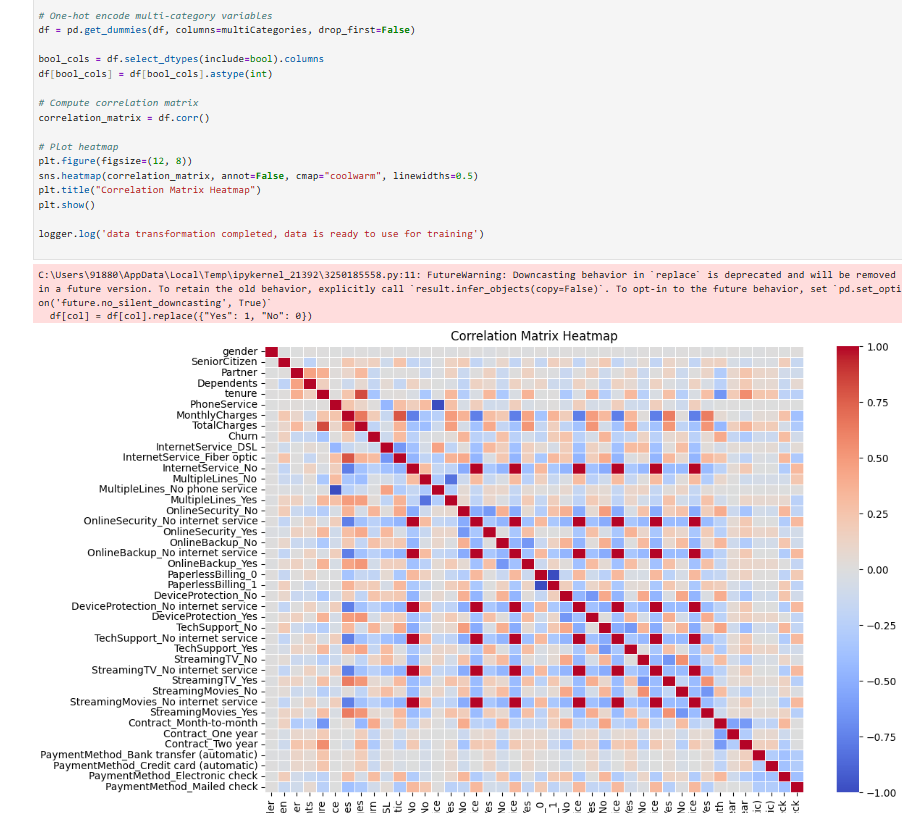
After the validation, same cleaned csv’s are stored in different place

1. **Data Transformation and Storage**

In Transformation, data is prepared by encoding binary and categorical values and printing correlation values

Output of python notebook:



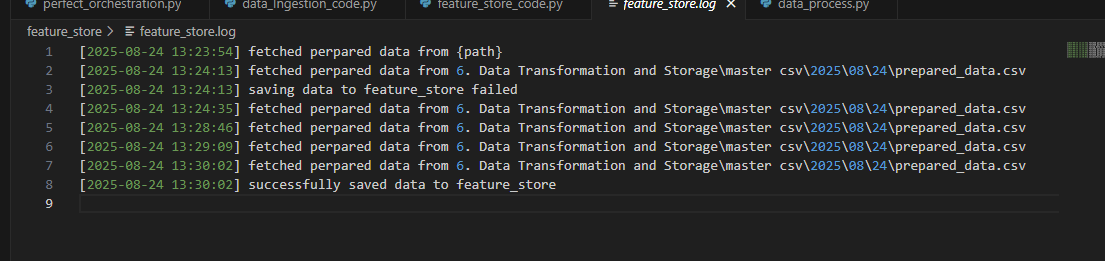


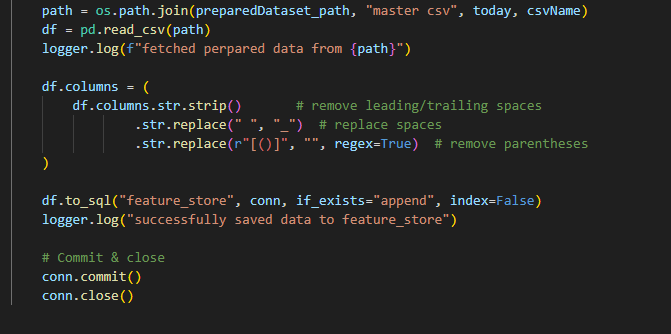
These data are stored in prepared\_data.csv

1. **Feature Store**

Finally data in prepared\_data.csv is stored in sqlite database which will be used in model training







Logs are also added for every process.

1. **Data Versioning**

we are using github to save each and every process ingestion, validation, cleaning, transformation, feature\_stores, models.

Github link:

<https://github.com/vinaylingam/DMML-Assignment>

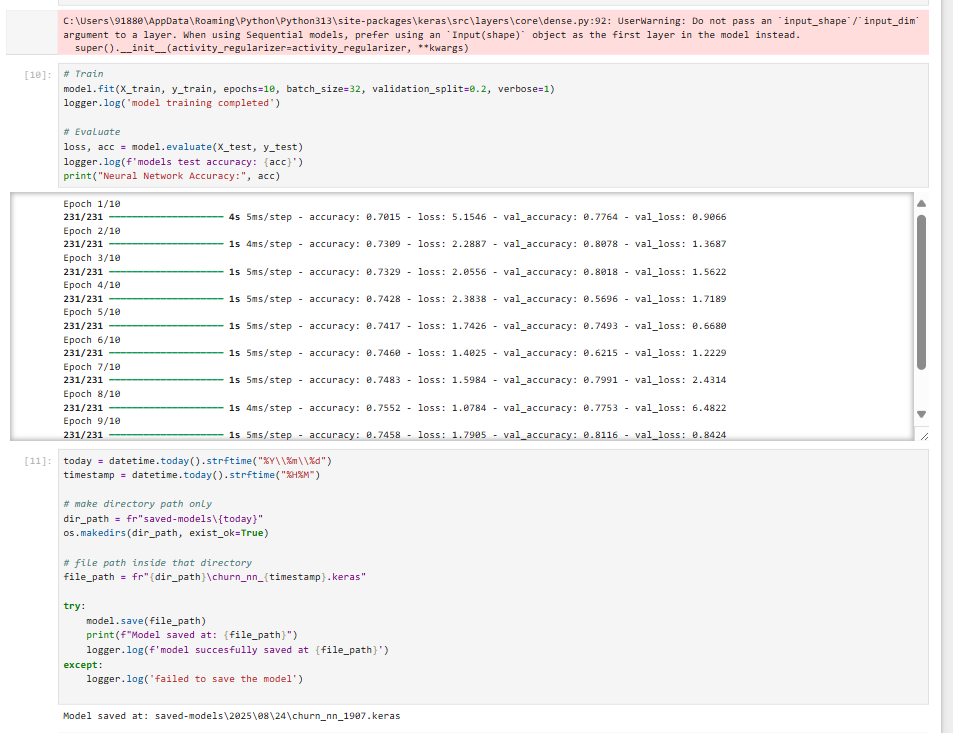
with this we can revert to previous data or simple use the model from previous trainings if there are any discrepancies

1. **Model Building**

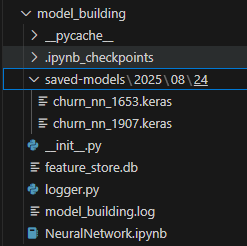
In this data is fetched from feature store and used in Neural networks to train model and save the versions of trained models

Code :





Models are saved with date and time stamps

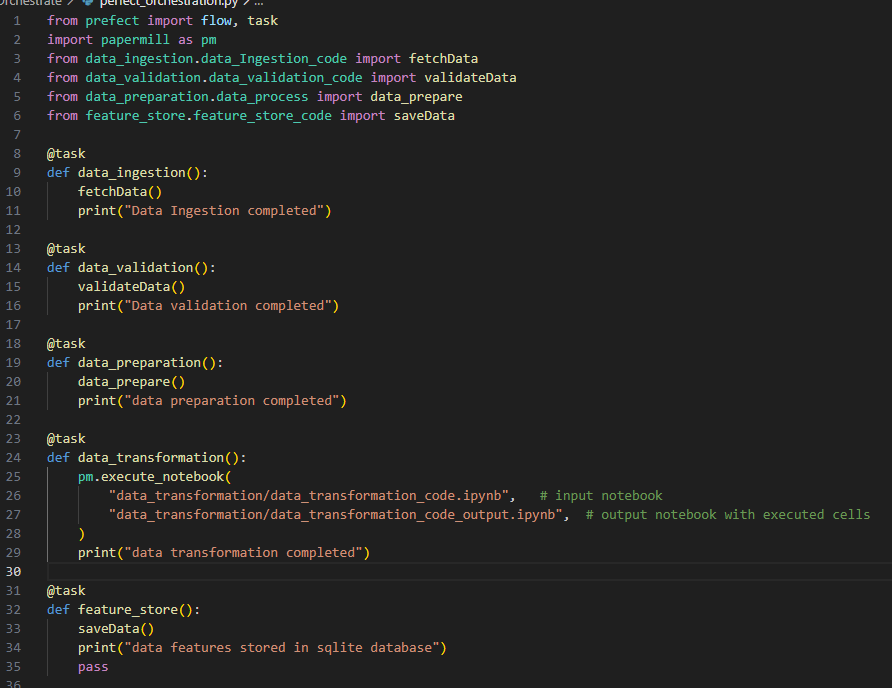


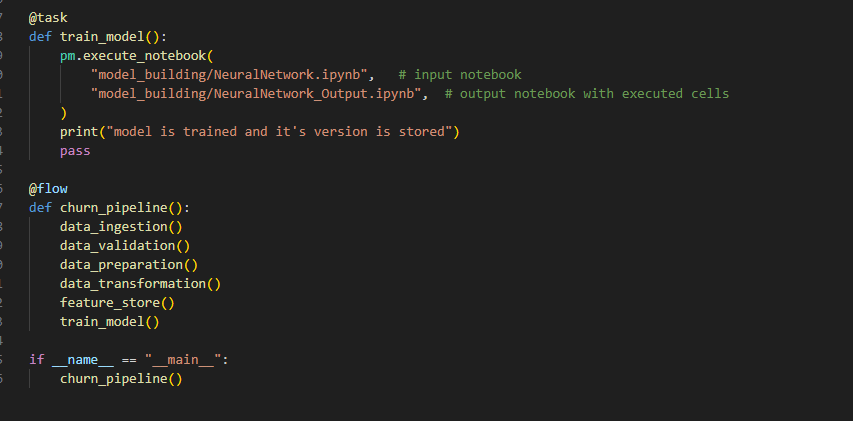
1. **Orchaestration**

For orchaestration, we used prefect we setup flows and tasks

And a cron job will run the pipeline everyday at 06:00 hrs

Code:





Output:

