https://www.linkedin.com/in/hijirdella/ | https://github.com/hijirdella | hijirdw@gmail.com

Data Engineering

Hijir Della Wirasti

Data Engineering

FINAL PROJECT

DISCUSSION TOPIC FOR TODAY

01 Introduction

02 Education

03 Professional Experience

04 Overview Project

05 Project Background

06 Problem Statement

07 Data Platform Understanding

08 Data Understanding

07 Data Modeling (Business)

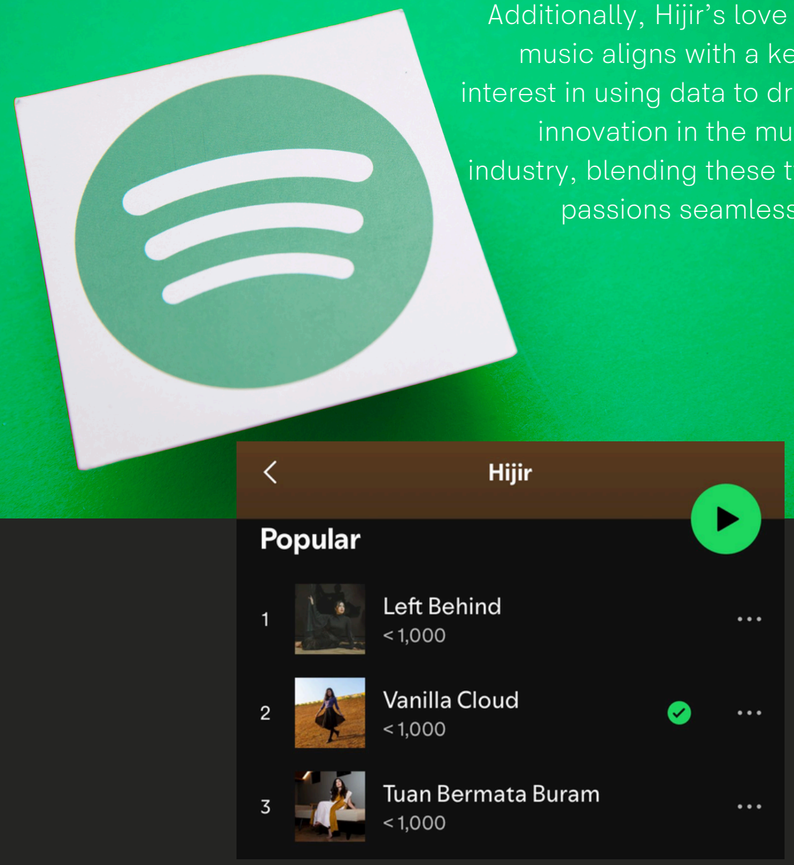
08 Conclusion & Recommendation

Self Overview

ABOUT HIJIR

Hijir is a student with a strong passion for learning, especially in technology and music. This curiosity drives an eagerness to explore fields like digital marketing, data engineering, data science, and Business Intelligence. Hijir is fascinated by how data and technology can create impactful business solutions, whether through optimizing marketing strategies, building data infrastructure, or transforming raw data into actionable insights.

Additionally, Hijir's love for music aligns with a keen interest in using data to drive innovation in the music industry, blending these two passions seamlessly.



Hijir Della Wirasti

Formal and informal

EDUCATION

Institut Teknologi Bandung

-Bachelor's in Ocean Engineering-
Grade: 3.21 / 4.00
2011 - 2016

Universitas Pendidikan Indonesia

Bachelor's in Music Education-
Grade: 3.57 / 4.00
2013 - 2019

REVOU

Digital Marketing
Grade: 94.62 / 100
Jan 2022 - Apr 2022

Telkom University

-Master's in Information Systems-
Grade: 3.71 / 4.00
Sep 2022 - Oct 2024
Hijir has completed her thesis and is waiting for the final defense schedule.

WORKING

PT Dinamaritama Konsultan Rekayasa

Ocean Engineer Consultant
July 2016 - March 2022

Managed 20 government projects, reduced maritime risks, ensured design safety (for Coastal Protection Structures, Offshore Platforms, Docks, Flood Control Engineering), maximized profitability through budget management, developed execution plans, and created cost estimates for informed investments.

PT Astana Digital Nusantara
(Katalis Insight, Katarsis, Madebyhumans)

Digital Marketing Specialist
May 2022 - October 2022

Elevated brand visibility for fashion and beauty clients by leveraging Key Opinion Leaders (KOL) and analyzing marketing metrics to ensure efficient resource allocation and consistent conversion growth. Orchestrated successful campaigns across multiple media channels, including Meta Ads and TikTok Ads, achieving a ROAS consistently exceeding 10, collaborated with Metaverse to launch Southeast Asia's first Augmented Reality Ads for Vietjet Air.

Sony Music Entertainment Indonesia

Artist & Repertoire Executive
October 2022 - October 2023

Achieved a monthly release of four songs, managed 136 annual song releases, directed production for 29 Sony artists, nurtured careers, curated music, collaborated cross-functionally, and built local music community connections.

Believe

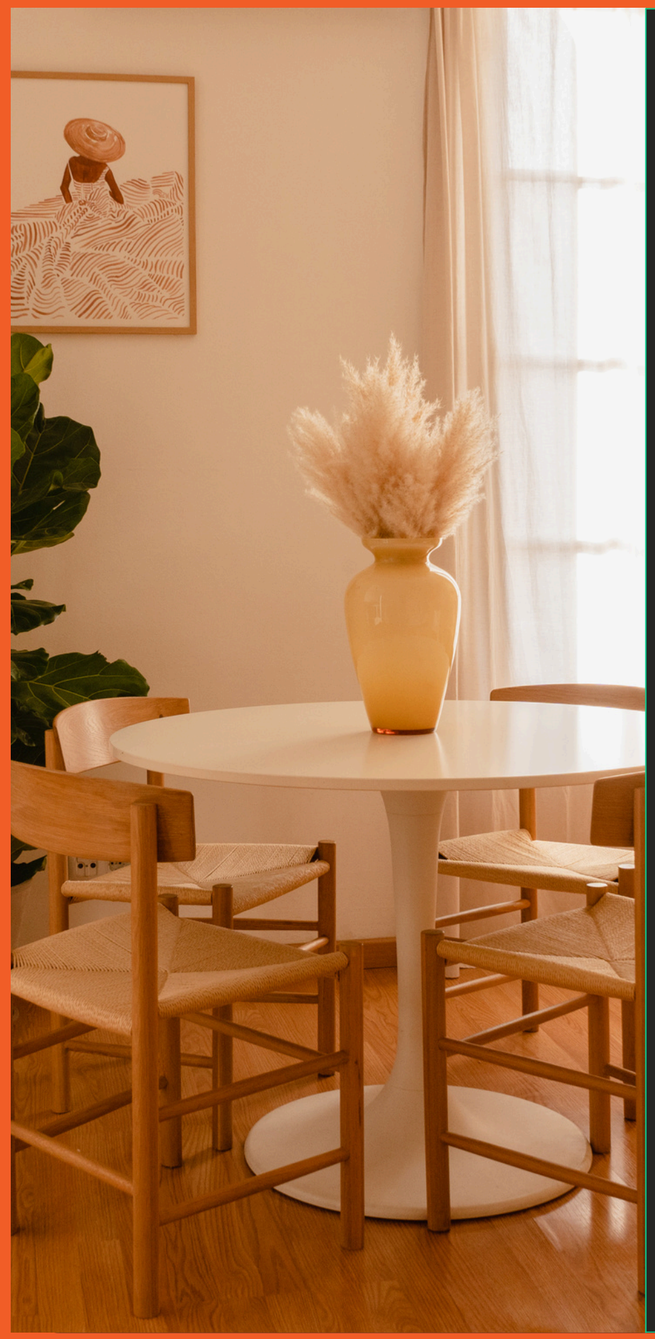
Artist & Repertoire Manager
December 2023 - May 2024

Successfully acquired Javanese Pop and hyperlocal artists. A&R acted more as sales, did marketing analytics for revenue projection and contract manager.

DATA ENGINEERING PROJECT:

Building a Telco Churn Analysis Pipeline

End-to-end Telco Churn Data Engineering Project using Docker, Python, Apache Airflow, PostgreSQL, and Spark. This project automates data ingestion, processing, and storage through a scalable pipeline. Airflow orchestrates the workflow, Spark handles batch processing, and PostgreSQL serves as the primary database.



MAIN PROJECT

https://github.com/hijirdella/Telco-Churn-Data-Pipeline

1. Project Background

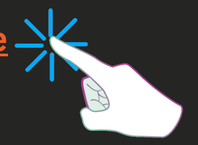
The project involves building a scalable data pipeline for analyzing customer churn in the telecommunications sector. The main objective is to automate data ingestion, transformation, and churn prediction, ensuring a reliable, scalable, and repeatable workflow. This analysis is vital for telecommunication companies to understand customer retention patterns, enabling them to strategize on reducing churn. The project will benefit the company by providing actionable insights to minimize customer loss and maximize revenue.

2. Problem Statement

The primary problem this project aims to solve is the lack of efficient and automated churn analysis for a large dataset in the telecommunications industry. Without an automated system, analyzing customer churn can be slow and prone to errors, leading to delayed insights. This project seeks to implement a solution that uses Apache Airflow and Apache Spark to automate and scale the process of churn analysis. Success will be measured by the accuracy of churn predictions, processing time improvements, and the pipeline's scalability.

MAIN PROJECT

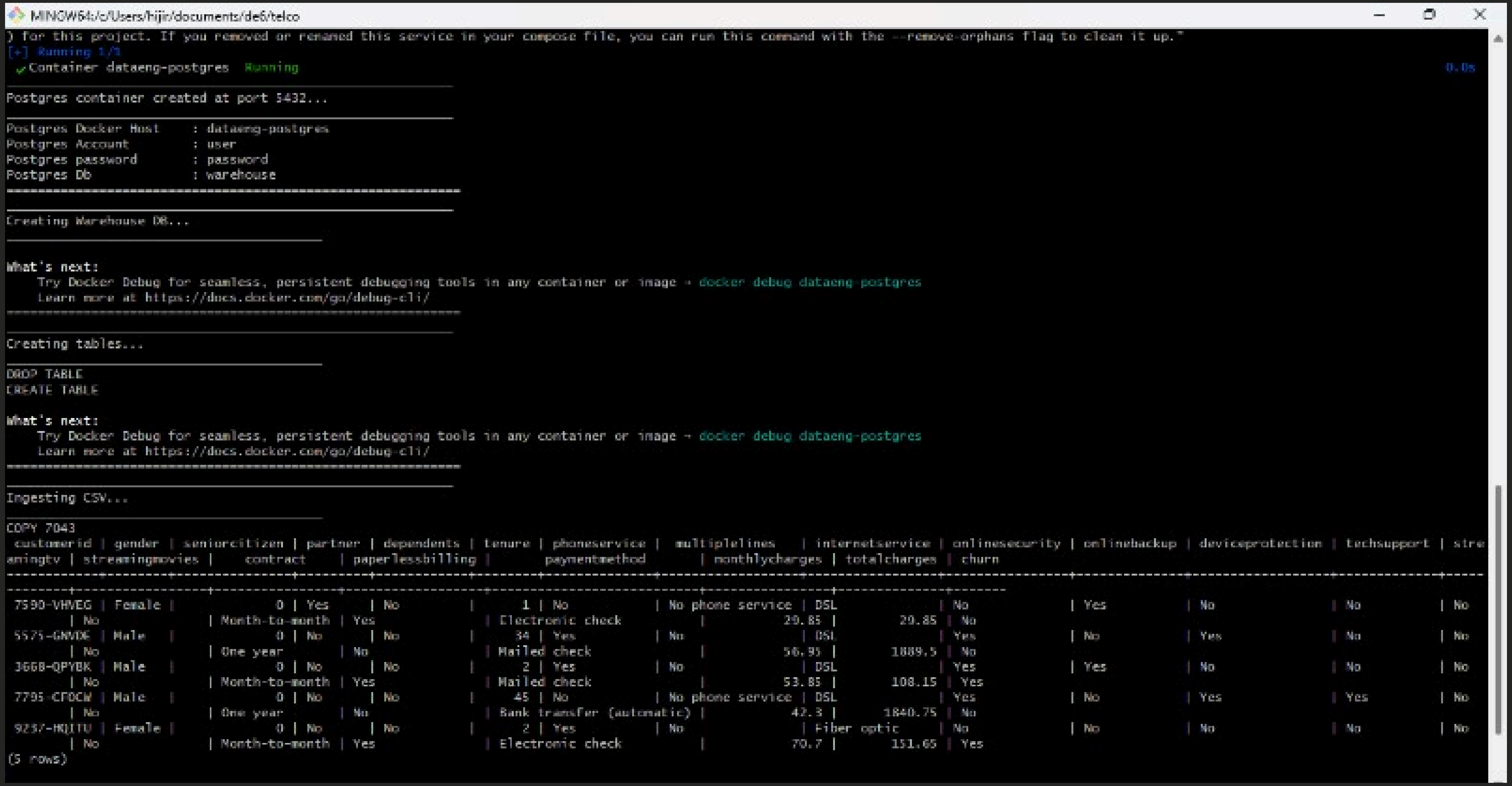
https://github.com/hijirdella/Telco-Churn-Data-Pipeline



3. Data Platform Understanding

The data pipeline integrates various tools to manage the entire flow from raw data ingestion to final churn predictions. Data is sourced from customer datasets stored in a PostgreSQL database, and Apache Airflow is used to orchestrate tasks. Spark handles the data transformation and analysis processes. The data flow is managed using Docker containers, ensuring consistency across environments.

CSV


MINGW64:/c/Users/hijir/documents/de6/telco
) for this project. If you removed or renamed this service in your compose file, you can run this command with the --remove-orphans flag to clean it up."
[+] Running 1/1
✔ Container dataeng-postgres Running 0.0s
Postgres container created at port 5432...
Postgres Docker Host : dataeng-postgres
Postgres Account : user
Postgres password : password
Postgres Db : warehouse
Creating Warehouse DB...
What's next:
Try Docker Debug for seamless, persistent debugging tools in any container or image → docker debug dataeng-postgres
Learn more at https://docs.docker.com/go/debug-cli/
Creating tables...
DROP TABLE
CREATE TABLE
What's next:
Try Docker Debug for seamless, persistent debugging tools in any container or image → docker debug dataeng-postgres
Learn more at https://docs.docker.com/go/debug-cli/
Ingesting CSV...
COPY 7043
customerid | gender | seniorcitizen | partner | dependents | tenure | phoneservice | multiplelines | internetservice | onlinesecurity | onlinebackup | deviceprotection | techsupport | stre
amingtv | streamingmovies | contract | paperlessbilling | paymentmethod | monthlycharges | totalcharges | churn
7590-VHVEG | Female | 0 | Yes | No | 1 | No | No phone service | DSL | No | Yes | No | No | No
| No | Month-to-month | Yes | Electronic check | 29.85 | 29.85 | No
5575-GNVDE | Male | 0 | No | No | 34 | Yes | No | DSL | Yes | No | Yes | No | No
| No | One year | No | Mailed check | 56.95 | 1889.5 | No
3668-QPYBK | Male | 0 | No | No | 2 | Yes | No | DSL | Yes | Yes | No | No | No
| No | Month-to-month | Yes | Mailed check | 53.85 | 108.15 | Yes
7795-CFOCW | Male | 0 | No | No | 45 | No | No phone service | DSL | Yes | No | Yes | Yes | No
| No | One year | No | Bank transfer (automatic) | 42.3 | 1840.75 | No
9237-HQITU | Female | 0 | No | No | 2 | Yes | No | Fiber optic | No | No | No | No | No
| No | Month-to-month | Yes | Electronic check | 70.7 | 151.65 | Yes
(5 rows)

Telco-Churn-Data-Pipeline / Makefile

Code | Blame | 190 lines (163 loc) · 8.33 KB | Code 55% faster with GitHub Copilot | Raw

```
98          @echo '============================================================'
99
100     postgres-create-table:
101         @echo '_____________________________________________________'
102         @echo 'Creating tables...'
103         @echo '_____________________________________________'
104         @docker exec -it ${POSTGRES_CONTAINER_NAME} psql -U ${POSTGRES_USER} -d ${POSTGRES_DW_DB} -f sql//ddl-telco.sql
105         @echo '============================================================'
106
107     postgres-ingest-csv:
108         @echo '_____________________________________________________'
109         @echo 'Ingesting CSV...'
110         @echo '_____________________________________________'
111         @docker exec -it ${POSTGRES_CONTAINER_NAME} psql -U ${POSTGRES_USER} -d ${POSTGRES_DW_DB} -f sql//ingest-telco.sql
112         @echo '============================================================'
113
114     postgres-create-warehouse:
115         @echo '_____________________________________________________'
116         @echo 'Creating Warehouse DB...'
117         @echo '_____________________________________________'
118         @docker exec -it ${POSTGRES_CONTAINER_NAME} psql -U ${POSTGRES_USER} -d ${POSTGRES_DB} -f sql//warehouse-ddl.sql
119         @echo '============================================================'
120
121     kafka: kafka-create
122
```

Telco-Churn-Data-Pipeline / docker / docker-compose-postgres.yml

hijirdella Final Project Data Engineering Dibimbing

Code | Blame | 26 lines (24 loc) · 624 Bytes | Code 55% faster with GitHub Copilot

```
1   version: '3.8'
2
3   services:
4     dibimbing-dataeng-postgres:
5       image: postgres:11
6       container_name: ${POSTGRES_CONTAINER_NAME}
7       restart: unless-stopped
8       hostname: ${POSTGRES_CONTAINER_NAME}
9       networks:
10        - dataeng-network
11      environment:
12        - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
13        - POSTGRES_USER=${POSTGRES_USER}
14        - POSTGRES_DB=${POSTGRES_DB}
15        - PGDATA=/var/lib/postgresql/data/pgdata
16        - POSTGRES_DW_DB =${POSTGRES_DW_DB}
17      volumes:
18        - ../sql:/sql
19        - ../data:/data
20      ports:
21        - ${POSTGRES_PORT}:5432
22
23  networks:
24    dataeng-network:
25      driver: bridge
26      external: true
```

Telco-Churn-Data-Pipeline / docker / docker-compose-airflow.yml

hijirdella Final Project Data Engineering Dibimbing

Code | Blame | 58 lines (56 loc) · 2.06 KB | Code 55% faster with Gi

```yaml
version: '3.7'
services:
  scheduler:
    image: dataeng-dibimbing/airflow
    container_name: ${AIRFLOW_SCHEDULER_CONTAINER_NAME}
    hostname: ${AIRFLOW_SCHEDULER_CONTAINER_NAME}
    command: scheduler
    restart: always
    environment:
      - AIRFLOW__CORE__SQL_ALCHEMY_CONN=postgresql+psycopg2://$
      - AIRFLOW__CORE__EXECUTOR=LocalExecutor
      - POSTGRES_USER=${POSTGRES_USER}
      - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
      - POSTGRES_DB=${POSTGRES_DB}
      - POSTGRES_CONTAINER_NAME=${POSTGRES_CONTAINER_NAME}
      - POSTGRES_PORT=${POSTGRES_PORT}
      - SPARK_MASTER_HOST_NAME=${SPARK_MASTER_HOST_NAME}
      - SPARK_MASTER_PORT=${SPARK_MASTER_PORT}
      - POSTGRES_DW_DB =${POSTGRES_DW_DB}
    volumes:
```

Telco-Churn-Data-Pipeline / docker / docker-compose-airflow.yml

Code | Blame | 58 lines (56 loc) · 2.06 KB | Code 55% faster with GitHu

```yaml
    depends_on:
      - scheduler
    environment:
      - AIRFLOW__CORE__SQL_ALCHEMY_CONN=postgresql+psycopg2://${PO
      - AIRFLOW__CORE__EXECUTOR=LocalExecutor
      - POSTGRES_USER=${POSTGRES_USER}
      - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
      - POSTGRES_DB=${POSTGRES_DB}
      - POSTGRES_CONTAINER_NAME=${POSTGRES_CONTAINER_NAME}
      - POSTGRES_PORT=${POSTGRES_PORT}
      - SPARK_MASTER_HOST_NAME=${SPARK_MASTER_HOST_NAME}
      - SPARK_MASTER_PORT=${SPARK_MASTER_PORT}
      - POSTGRES_DW_DB =${POSTGRES_DW_DB}
    volumes:
      - ../dags:/opt/airflow/dags
      - ../logs:/opt/airflow/logs
      - ../spark-scripts:/spark-scripts
      - ../scripts:/scripts
      - ../results:/opt/airflow/results
    ports:
      - ${AIRFLOW_WEBSERVER_PORT}:8080

networks:
  default:
    name: dataeng-network
```

MAIN PROJECT

https://github.com/hijirdella/Telco-Churn-Data-Pipeline

4. Data Understanding

The dataset used includes customer information such as demographics, service usage, and billing details, with attributes like customer ID, gender, tenure, monthly charges, and churn status. The data is structured in CSV format and contains thousands of rows. The dataset requires cleansing for missing values.

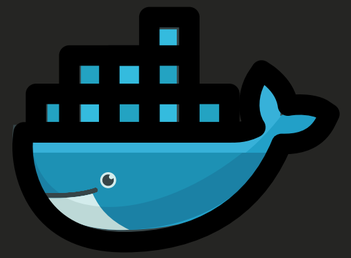
```
In [2]: # Load the data from the CSV file
        data = pd.read_csv("Telco-Customer-Churn.csv")

        # Display the first few rows of the dataset
        print("First few rows of the dataset:")
        display(data.head())
```

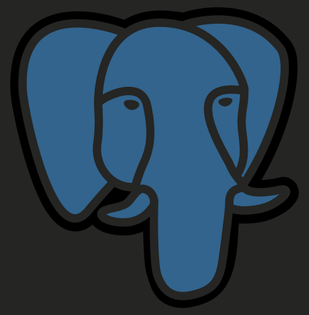
First few rows of the dataset:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	Devi
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	

5 rows × 21 columns

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object
 20  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```





MAIN PROJECT

https://github.com/hijirdella/Telco-Churn-Data-Pipeline

4. Data Understanding (2)

Descriptive statistics of the dataset:

	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

```python
# Konversi kolom 'TotalCharges' ke tipe numerik, menggantikan nilai tidak valid menjadi NaN
data['TotalCharges'] = pd.to_numeric(data['TotalCharges'], errors='coerce')

# Cek baris yang memiliki nilai NaN setelah konversi
invalid_rows = data[data['TotalCharges'].isna()]
print("Baris yang memiliki nilai tidak valid di 'TotalCharges':")
print(invalid_rows)

# Isi nilai NaN dengan nilai median dari kolom 'TotalCharges'
median_value = data['TotalCharges'].median()
data['TotalCharges'].fillna(median_value, inplace=True)
print(f"Nilai median yang digunakan untuk menggantikan NaN: {median_value}")

# Simpan data yang sudah dibersihkan ke file CSV baru
data.to_csv('telco_customer.csv', index=False)
print("File CSV berhasil disimpan dengan nilai NaN diganti menggunakan median.")
```

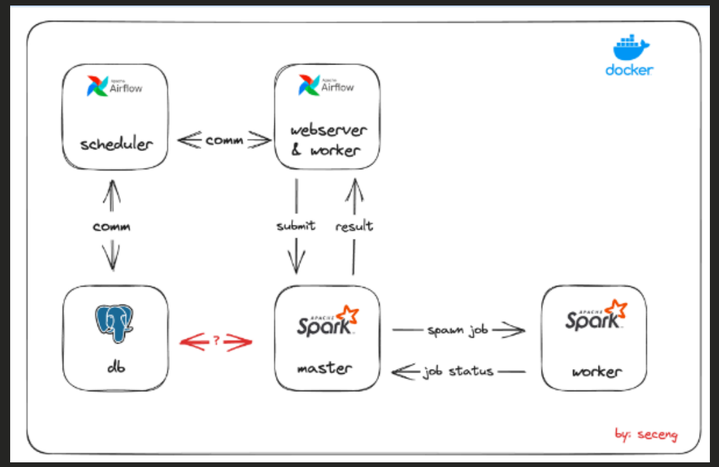
```
                 PaymentMethod  MonthlyCharges  TotalCharges Churn
488   Bank transfer (automatic)           52.55           NaN    No
753               Mailed check           20.25           NaN    No
936               Mailed check           80.85           NaN    No
1082              Mailed check           25.75           NaN    No
1340    Credit card (automatic)           56.05           NaN    No
3331              Mailed check           19.85           NaN    No
3826              Mailed check           25.35           NaN    No
4380              Mailed check           20.00           NaN    No
5218              Mailed check           19.70           NaN    No
6670              Mailed check           73.35           NaN    No
6754  Bank transfer (automatic)           61.90           NaN    No
Nilai median yang digunakan untuk menggantikan NaN: 1397.475
File CSV berhasil disimpan dengan nilai NaN diganti menggunakan median.
```

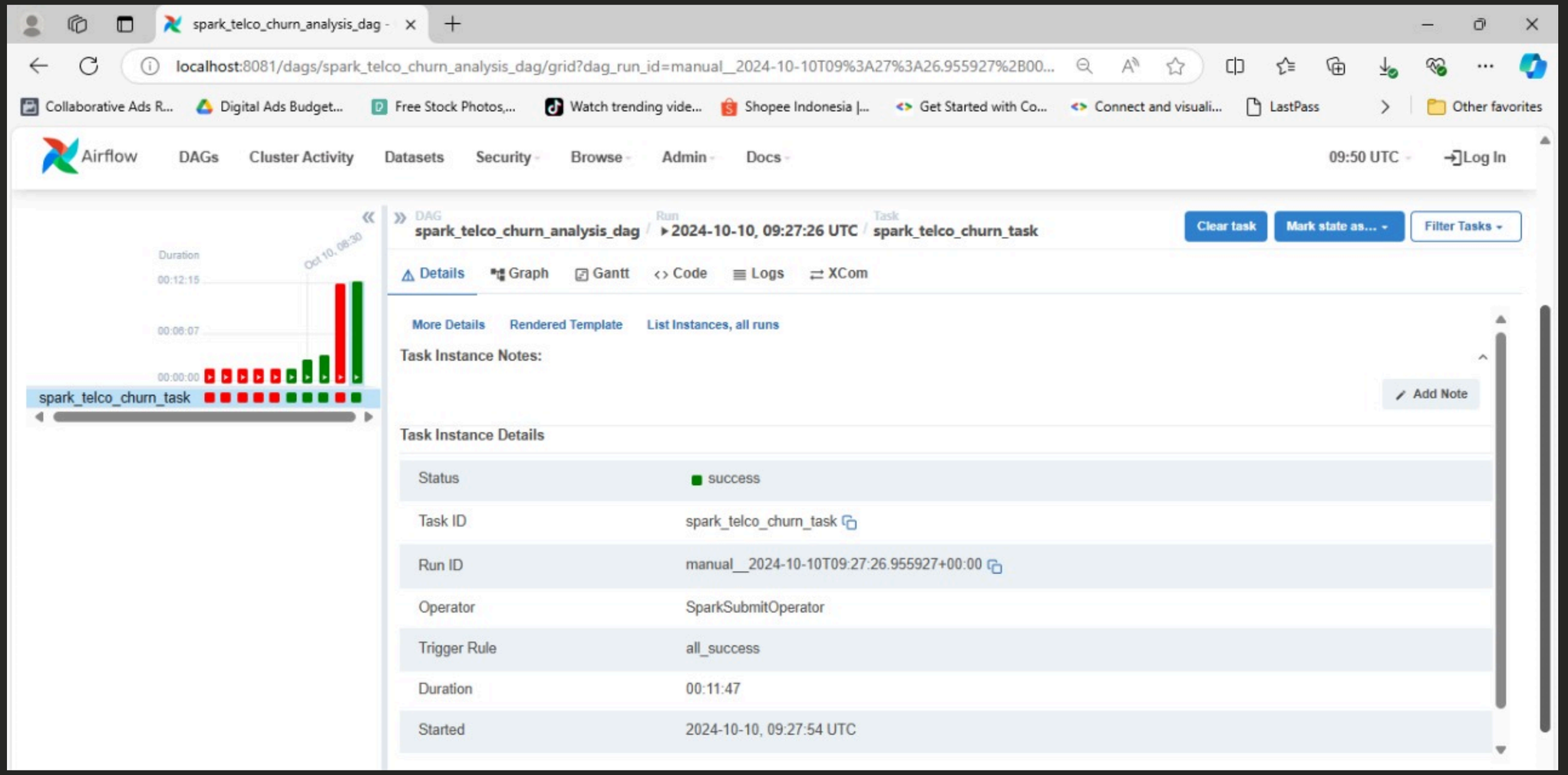
MAIN PROJECT

https://github.com/hijirdella/Telco-Churn-Data-Pipeline

5. Transformation & Consideration

The project uses Apache Spark for batch processing due to the size of the dataset. Given the large data volume, batch processing is appropriate to handle data transformations efficiently. The transformation includes data cleaning, aggregation, and preparation for churn analysis. The pipeline follows the Directed Acyclic Graph (DAG) structure orchestrated by Airflow, ensuring that tasks such as data extraction, transformation, loading (ETL), and analysis are performed in sequence. The architecture is designed to be scalable, with Docker ensuring a consistent and isolated environment for each component.



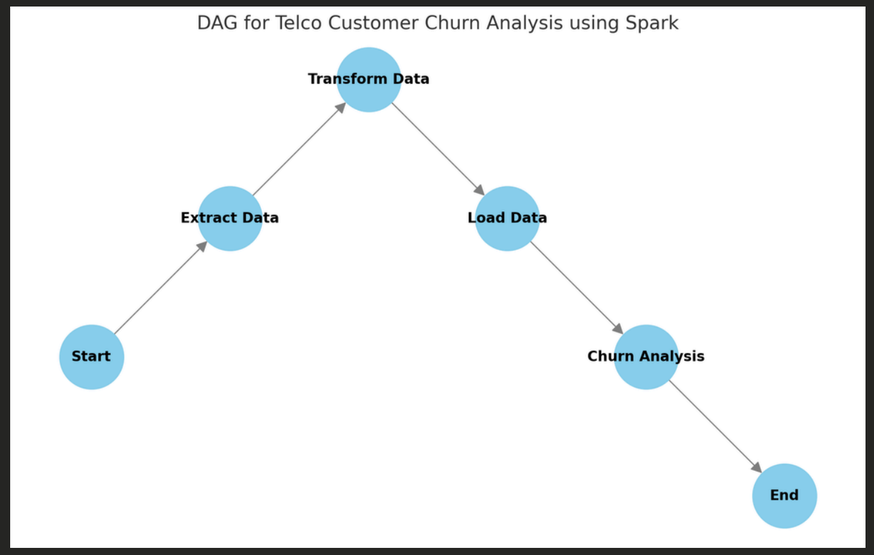

spark_telco_churn_analysis_dag
localhost:8081/dags/spark_telco_churn_analysis_dag/grid?dag_run_id=manual__2024-10-10T09%3A27%3A26.955927%2B00...
Airflow
DAGs
Cluster Activity
Datasets
Security
Browse
Admin
Docs
09:50 UTC
Log In
Duration
00:12:15
00:06:07
00:00:00
spark_telco_churn_task
DAG
spark_telco_churn_analysis_dag
Run
2024-10-10, 09:27:26 UTC
Task
spark_telco_churn_task
Clear task
Mark state as...
Filter Tasks
Details
Graph
Gantt
Code
Logs
XCom
More Details
Rendered Template
List Instances, all runs
Task Instance Notes:
Add Note
Task Instance Details
Status
success
Task ID
spark_telco_churn_task
Run ID
manual__2024-10-10T09:27:26.955927+00:00
Operator
SparkSubmitOperator
Trigger Rule
all_success
Duration
00:11:47
Started
2024-10-10, 09:27:54 UTC

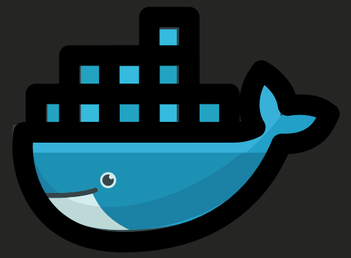
MAIN PROJECT

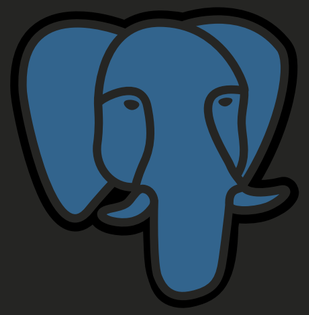
https://github.com/hijirdella/Telco-Churn-Data-Pipeline

6. Data Modeling (Business)

The data model is designed to support business needs by creating tables that provide insights into customer behavior. These include partitioning and clustering tables based on key metrics like customer tenure and contract type to improve query performance. The goal is to design an efficient schema that optimizes storage while allowing for fast access to churn-related data.







MAIN PROJECT

https://github.com/hijirdella/Telco-Churn-Data-Pipeline

6. Data Modeling (Business) (2)

Churn Berdasarkan Internet Service:

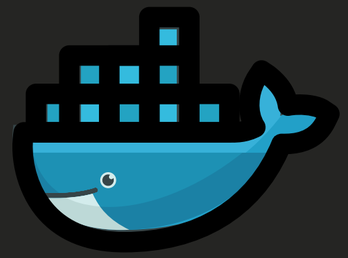
InternetService	TotalChurnedCustomers	TotalCustomers	ChurnRate
Fiber optic	1297	3096	0.4189276485788114
DSL	459	2421	0.1895910780669145
No	113	1526	0.07404980340760157

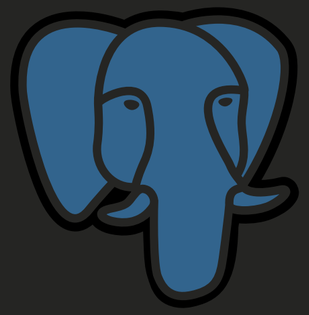
Rata-rata Monthly Charges Berdasarkan Contract Type:

Contract	AvgMonthlyCharges
Month-to-month	66.39849032258037
One year	65.04860828241674
Two year	60.770412979351

Churn Berdasarkan Contract Type:

Contract	TotalChurnedCustomers	TotalCustomers	ChurnRate
Month-to-month	1655	3875	0.4270967741935484
One year	166	1473	0.11269517990495587
Two year	48	1695	0.02831858407079646





MAIN PROJECT

https://github.com/hijirdella/Telco-Churn-Data-Pipeline

6. Data Modeling (Business) (2)

Churn Berdasarkan Tenure:

tenure	TotalChurnedCustomers	TotalCustomers	ChurnRate
0	0	11	0.0
1	380	613	0.6199021207177814
2	123	238	0.5168067226890757
3	94	200	0.47
4	83	176	0.4715909090909091
5	64	133	0.48120300751879697
6	40	110	0.36363636363636365
7	51	131	0.3893129770992366
8	42	123	0.34146341463414637
9	46	119	0.3865546218487395
10	45	116	0.3879310344827586
11	31	99	0.31313131313131315
12	38	117	0.3247863247863248
13	38	109	0.3486238532110092
14	24	76	0.3157894736842105
15	37	99	0.37373737373737376
16	28	80	0.35
17	26	87	0.2988505747126437
18	24	97	0.24742268041237114
19	19	73	0.2602739726027397

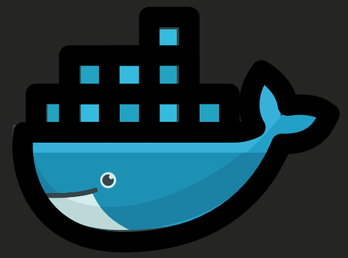
only showing top 20 rows

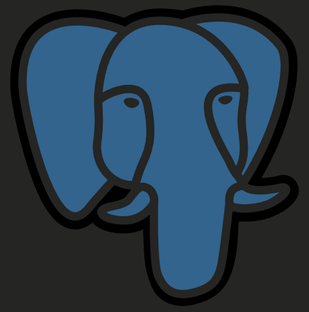
Distribusi Pelanggan Berdasarkan Metode Pembayaran:

PaymentMethod	count
Electronic check	2365
Mailed check	1612
Bank transfer (au...	1544
Credit card (auto...	1522

Rata-rata Total Charges Berdasarkan Senior Citizen:

SeniorCitizen	AvgTotalCharges
1	2810.4651926444826
0	2177.023801050675





MAIN PROJECT

https://github.com/hijirdella/Telco-Churn-Data-Pipeline

7. Conclusion & Recommendation

The platform demonstrates its ability to process large datasets, automate data transformations, and deliver meaningful insights through churn prediction. While the platform is scalable and reliable, there are limitations, such as the need for further optimization in real-time processing and the reliance on batch processing for large datasets. Future work could include integrating real-time data streams to enable more timely churn predictions.

CONTACT ME

https://github.com/hijirdella

hijirdw@gmail.com

Jakarta Selatan

Hijir Della Wirasti

Data Engineer Final Project

THANK YOU !