WOMEN WHO CODE

How to Build Your Data Engineering Portfolio

Anju Mercian

Data Engineering Consultant
www.anjumercian.me

July 2022

Agenda

- About Me
- Portfolio Projects
- Data Engineering (DE)
- Technical Skills
- Project Steps to Guide You
- References
- Closing
- Q&A

About Anju Mercian

- Graduated from Syracuse University, NY with a Master's degree in Computer Engineering
- Worked at Intel for 5 years as an Infrastructure Engineer
- Worked at VMware for one year as a Project Manager
- Worked at Intuitive surgical for one year as a Business System Analyst
- Worked at Omdena as a Platform Engineer
- Currently working as a Data Engineering Consultant @ Meta

What Is a Portfolio Project?

"A portfolio project or a capstone project is a well thought out and designed project that showcases your skills to a potential employer or the world."

Benefits of a Portfolio Project

- *Showcase your skills*
- *Learn new skills*
- *Grow it as a side project after you have a job*
- *Build personal brand*
- *Help with Data Design Interviews*

What Is Data Engineering?

“Data engineering is the complex task of making raw data usable to data scientists and groups within an organization. Data engineering encompasses numerous specialties of data science.

In addition to making data accessible, data engineers create raw data analyses to provide predictive models and show trends for the short- and long-term. Without data engineering, it would be impossible to make sense of the huge amounts of data that are available to businesses.”

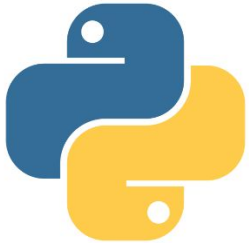
Source

Who Is a Data Engineer?

“Data engineers work in a variety of settings to build systems that collect, manage, and convert raw data into usable information for data scientists and business analysts to interpret. Their ultimate goal is to make data accessible so that organizations can use it to evaluate and optimize their performance.”

Source

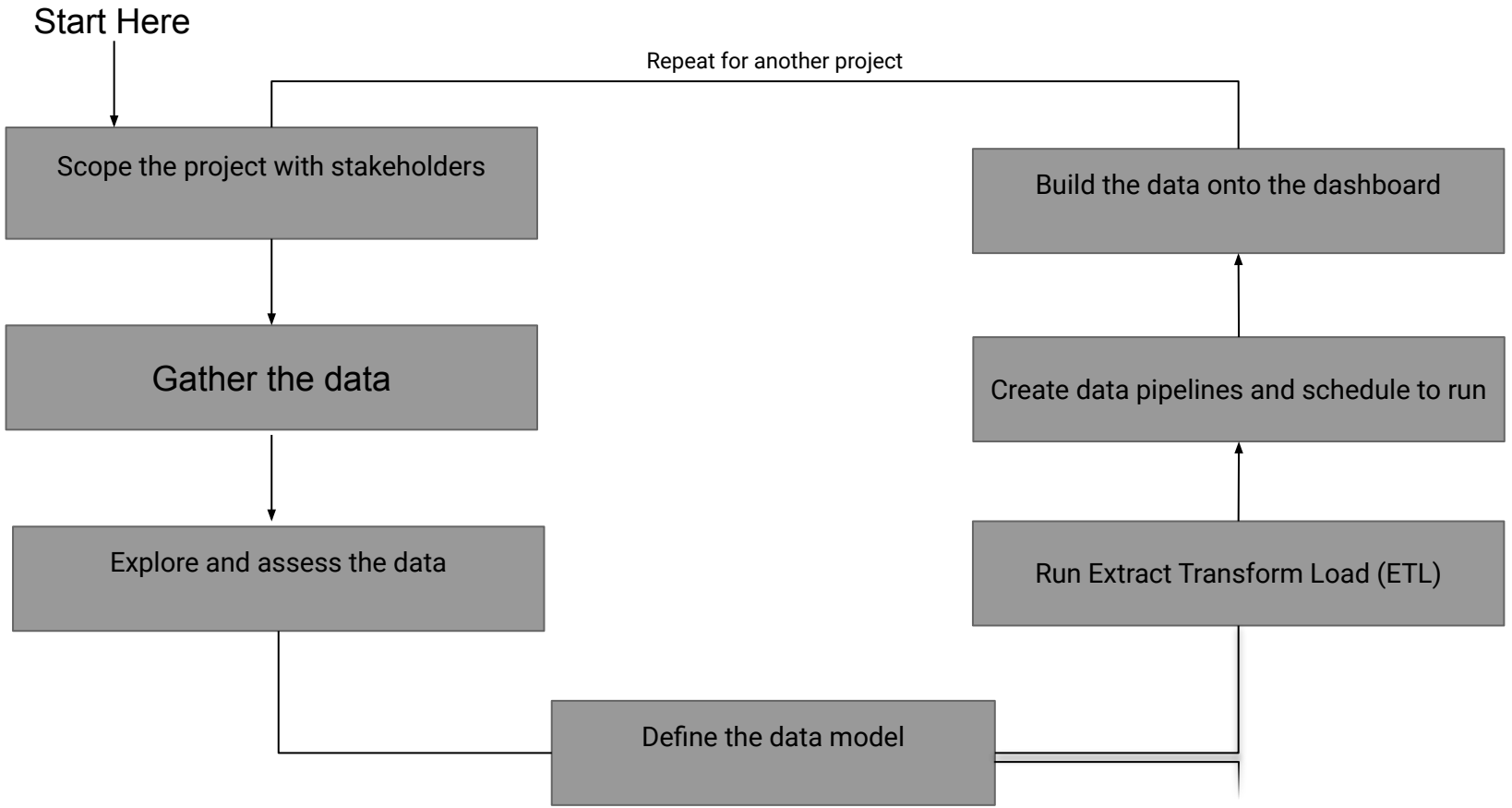
Technical Skills



EMR
RedShift
S3

Project Steps to Guide You

GitHub link


Start Here
Repeat for another project
Scope the project with stakeholders
Gather the data
Explore and assess the data
Define the data model
Run Extract Transform Load (ETL)
Create data pipelines and schedule to run
Build the data onto the dashboard

Scope the Project and Gather Data

***GOAL**:* This project aims to build a data analytics table using the US immigration data set, enriching the data with the demographics of the US cities dataset and the world temperature data for data analysis.

This is built for a data analytics table to help answer questions such as:

- Immigrants move to US cities
  - demographics/statistics?
- Immigrants move to US cities
  - Do they move to warmer weather or cooler weather?

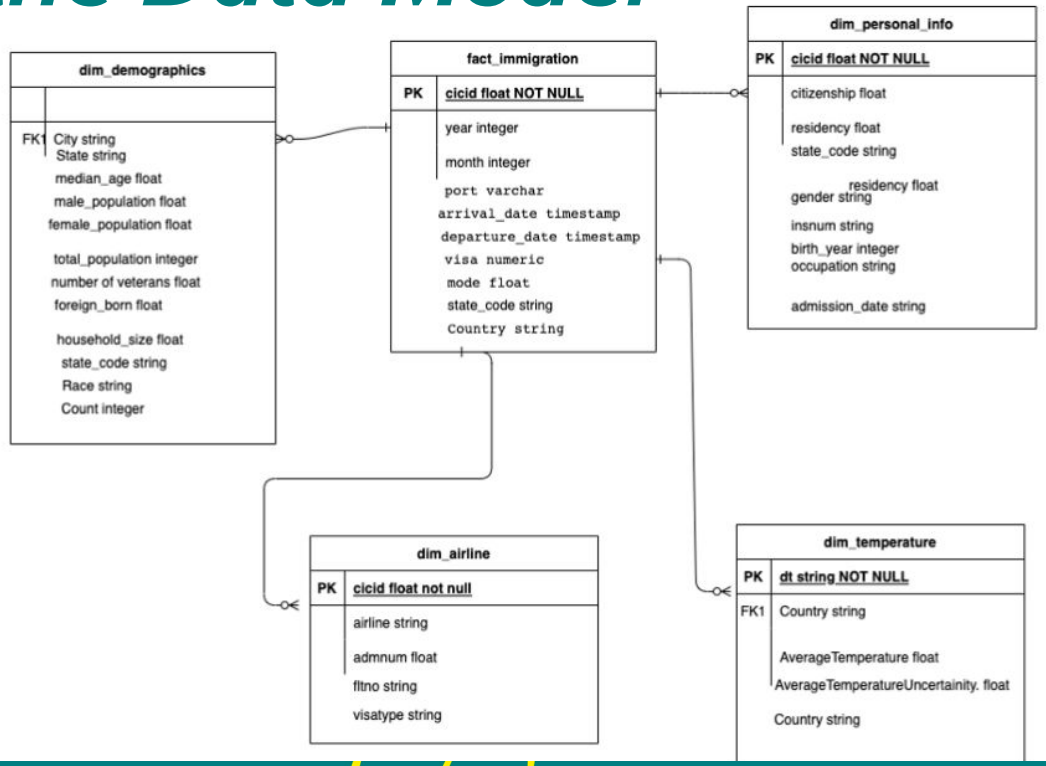
Explore and Access the Data

Data Sources:

- I94 Immigration Data: This data comes from the US National Tourism and Trade Office. A data dictionary is included in the workspace.
- World Temperature Data: This dataset came from Kaggle.
- U.S. City Demographic Data: This data comes from OpenSoft.

***Note**: Try choosing non-Kaggle data sources because Kaggle is one of the most used data sources and is relatively clean.*

Define the Data Model



[Source]

Run ETL to Model the Data

Modularized into functions the cleaning steps and creation of the fact and dimension tables. Data quality checks happened at this stage. Created the data dictionary.

***Note**: Data dictionaries and modularizing the pipelines are very important and useful for portability of code.*

Deploy

Deploy on GitHub or a dashboard.

Showcasing the answer to my question that I wrote in the beginning for why I was looking at this data:

- immigrants move to US cities with what kind of demographics statistics?
- immigrants move to US cities with what kind of temperature, do they move to warmer weather or cooler weather?
- Do they move to cities with a larger population or less population?

```
cur.execute("SELECT cicid, count(cicid), d.median_age, d.male_population, d.female_population FROM fact_immigration f \
             INNER JOIN dim_demographics d ON f.state_code = d.state_code \
             GROUP BY 1,3,4,5")
conn.commit()
print(cur.rowcount)
```

```
1081
```

```
#immigrants move to US cities with what kind of demographics statistics?
for row in cur:
    print(row)
```

```
cur.execute("SELECT d.total_population FROM fact_immigration f \
             INNER JOIN dim_demographics d ON f.state_code = d.state_code \
             GROUP BY 1")
conn.commit()
print(cur.rowcount)
```

```
123
```

```
#Do they move to cities with a larger population or less population?: Total population of the cities the immigrants moved into listed.
for row in cur:
    print(row)
```

Skills to Showcase to Stand Out

Data Quality

“Data quality is the answer to the question “How is my data?” If your data helps you with business operations and decisions, then you can say that your data is of good quality.”

- Data must be a certain size
- Data must be accurate to some margin of error
- Data must arrive within a given timeframe from the start of execution
- Pipelines must run on a particular schedule
- Data must not contain any sensitive information

Data Quality

4.2 Data Quality Checks

Explain the data quality checks you'll perform to ensure the pipeline ran as expected. These could include:

- Integrity constraints on the relational database on making sure the primary key is not duplicate.
- Unit tests for the scripts to ensure they are doing the right thing
- Source/Count checks to ensure completeness

Run Quality Checks

In [43]:
```
# Perform quality checks here
cur.execute("SELECT COUNT(*) FROM fact_immigration")
conn.commit()
if cur.rowcount < 1:
    print("No data found in table fact_immigration")

cur.execute("SELECT COUNT(*) FROM dim_airline")
conn.commit()
if cur.rowcount < 1:
    print("No data found in table dim_airline")
```

In [44]:
```
cur.execute("SELECT cicid, COUNT(cicid) FROM fact_immigration GROUP BY cicid HAVING COUNT(cicid) > 1")
conn.commit()
print(cur.rowcount)
```

```
0
```

In [45]:
```
cur.execute("SELECT cicid, count(cicid) FROM fact_immigration f \
             INNER JOIN dim_demographics d ON f.state_code = d.state_code \
             GROUP BY 1")
conn.commit()
print(cur.rowcount) #printitng the rowcount.The ETL is working, checked with mergeing fact_immigration and dim_demographics table.
```

```
196
```

Apache Spark

- Spark is currently one of the most popular tools for big data analytics. There are others like Hadoop, but Hadoop is an older technology.
- Spark is faster than Hadoop, and it is developer friendly.
- Apache Spark comes with the ability to run multiple workloads, including interactive queries, real-time analytics, machine learning, and graph processing. One application can combine multiple workloads seamlessly.

```
#from pyspark.sql import SparkSession

#spark = SparkSession.builder.\
#config("spark.jars.repositories", "https://repos.spark-packages.org/").\
#config("spark.jars.packages", "saurfang:spark-sas7bdat:2.0.0-s_2.11").\
#enableHiveSupport().getOrCreate()

#df_spark = spark.read.format('com.github.saurfang.sas.spark').load('../../data/18-83510-I94-Data-2016/i94_apr16_sub.sas7bdat')
```

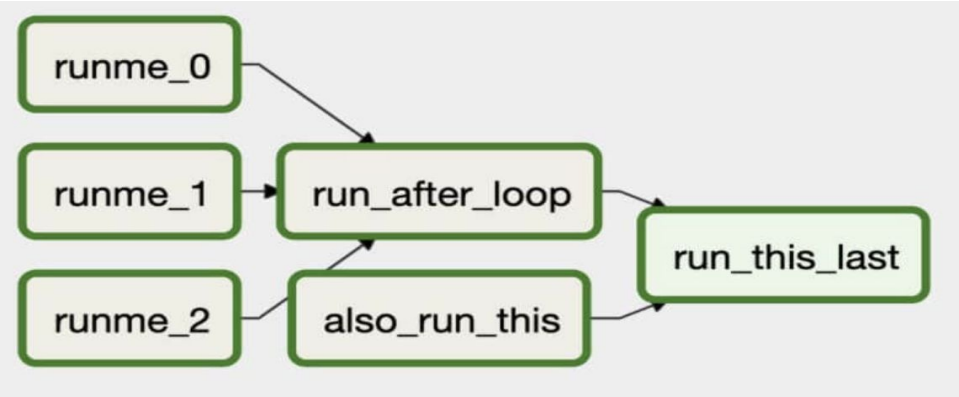
```
#write to parquet
#df_spark.write.parquet("sas_data")
#df_spark=spark.read.parquet("sas_data")
```

WOMEN WHO CODE

Apache Airflow (DAG)

Data Pipelines: A series of steps in which data is processed.

Directed Acyclic Graphs (DAGs): DAGs are a special subset of graphs in which the edges between nodes have a specific direction, and no cycles exist. When we say “no cycles exist”, this means the nodes cannot create a path back to themselves.



Apache Airflow (DAG)

Apache Airflow: “Airflow is a platform to programmatically author, schedule and monitor workflows. Use Airflow to author workflows as directed acyclic graphs (DAGs) of tasks. The Airflow scheduler executes your tasks on an array of workers while following the specified dependencies. Rich command line utilities make performing complex surgeries on DAGs a snap. The rich user interface makes it easy to visualize pipelines running in production, monitor progress, and troubleshoot issues when needed. When workflows are defined as code, they become more maintainable, versionable, testable, and collaborative."

Cloud Deployment

Amazon - Redshift, EMR, S3

Microsoft Azure

GCP (Google Cloud Platform)

References

More Details

Apache Airflow

Apache Kafka

Apache Spark

Data Quality

Sign Up Future Workshops in WWCode Python Track

Closing

- Scope requirements and gather data
- Explore and access the data.
- Define the Data Model - Data Design
- Run ETL Pipelines
- Deploy to a dashboard or github.
- Upgrade using Pyspark, Cloud, Automated pipelines.
- Essential Data Quality

Q & A

WOMEN WHO CODE®