**1). Project 1:** Real-time Data Pipeline for Stock Market Analysis. **- Objective:** Build a pipeline to ingest, process, and analyze stock market data in real-time.

**Project 2:** Scrape data from Amazon, Jumia, or any other e-commerce website to create a list of all products currently offered at a discount.-

**Project 3:** Automating Data Scrapers and Analytical Processes using Apache Airflow.

**Project 4:** Kenya YouTube Channels Analysis using Python and YouTube API. **Tools, Frameworks, and Technologies:** Python, YouTube API, Requests, Pandas, Matplotlib, Seaborn, etc etc. Feel free to use different tools and approach. **Objective:** Analyze YouTube channels in Kenya using Python. This may include video content analysis, subscriber trends, and engagement metrics.

**Project 5:** Build a Kenya and East Africa Agricultural Data Portal to provided necessary information to advise the farmers and investors interested in Farming and Agribiz in Kenya. **Tools, Frameworks, and Technologies:** Python, Django, Flask, Pandas, Numpy, Data Visualization libraries. **Objective:** Develop a data portal for agricultural data in Kenya and East Africa. This involves collecting, organizing, and presenting agricultural data for analysis.

**Project 6:** Nairobi Metropolitan House Price Prediction with Python.

**Project 7:** Fitness Data Analysis: Case Study.

**Project 8:** Crop Yield Analysis in Kenya - With Python.

**Project 9:  Movie Analysis.** Download  a movie dataset from Kaggle or  Movie Database (TMDb) API and explore genres, ratings, director popularity, or release year trends.  Also create a 3 minutes guide and docs of your dashboard and 5 minutes guide  to the visualization tool you will decide to use.In this project you can use Power BI or any other data visualization tool to create dashboards that recommend movies based on user preferences or visualize box office trends across genres.

<https://emalumhejosh.medium.com/analyzing-a-movie-data-using-power-bi-59607998e861>

<https://github.com/gopiashokan/IMDB-Movie-Analysis-with-PowerBI>

**Project 10: Analyzing Employee Data with SQL or Python/R/Scala.**  
Download the employee dataset below from Kaggle: <https://www.kaggle.com/datasets/ravindrasinghrana/employeedataset>  
and  
Use SQL queries to filter data based on department, job title, or salary range.  
You can also practice joining tables to combine employee information with related data, like department budgets.

**EXPLORING EMPLOYEE DATA: A COMPREHENSIVE ANALYSIS USING SQL AND POWER BI:**<https://medium.com/@jacobodey001/exploring-employee-data-a-comprehensive-analysis-using-sql-and-power-bi-ad324d14f845>

**Using SQL to Analyze Employee Data:**<https://medium.com/@kingsleyofori/using-sql-to-analyze-employee-data-bd63761c319d>

**Project 11: Sample Data Science Project with Sentiment Analysis using Kaggle and Flask.**  
Build a Data Science project to demonstrate sentiment analysis and classify movie reviews as positive or negative. Use a dataset from Kaggle, train a machine learning model, and deploy a Flask API for sentiment prediction.  
Download the dataset here: <https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews>.  
**Data collection:**

* Download the training and testing data (usually in CSV format) provided by the chosen dataset.

**Data Preprocessing:**

1. Use libraries like pandas and NumPy for data manipulation and cleaning.
2. Load the CSV data into a pandas dataframe.
3. Clean the text data (reviews):

* Remove punctuation and stop words.
* Lowercase all text.
* Consider stemming or lemmatization for better word representation.

1. Convert the text data into numerical features suitable for machine learning models. Common techniques include:

* Bag-of-Words: Count occurrences of words in each review.
* TF-IDF: Considers both word frequency and document frequency for weighting words.

1. Encode the sentiment labels (positive/negative) into numerical values (e.g., 0 for negative, 1 for positive).
2. Split the data into training and testing sets.

**Model Training:**

1. Use libraries like scikit-learn for model training and evaluation.
2. Choose a classification model like Naive Bayes, Logistic Regression, or Support Vector Machines (SVM).
3. Train the model on the preprocessed training data.
4. Evaluate the model's performance on the testing data using metrics like accuracy, precision, recall, and F1 score.

**Model Deployment with Flask:**

1. Install Flask using pip install Flask.
2. Create a Flask application with an endpoint to receive user input (movie review text) and return the predicted sentiment (positive or negative).
3. Define functions for data preprocessing (similar to the preprocessing step) and model loading.
4. In the endpoint function:

* Receive user input as a string (review text).
* Preprocess the text data.
* Use the loaded model to predict sentiment on the preprocessed data.
* Return the predicted sentiment as a string (e.g., "positive" or "negative").

1. Run the Flask application using flask run.

**Testing the API:**  
Use tools like Postman or curl to send a POST request with the review text in the request body to the API endpoint and see the predicted sentiment returned as JSON.  
**PLEASE NOTE:**

* This is a basic project. Real-world projects may involve more complex preprocessing techniques and model selection.
* Consider error handling and user input validation in your Flask application.
* For deployment on a server, explore containerization with Docker for a production-ready environment.