Alda - Final project proposal

Netflix original programming

I. Introduction

This project aims to conduct an exploratory data analysis (EDA) on Netflix original programming, focusing on various aspects such as genre distribution, release trends, and runtime statistics. By analyzing data from Wikipedia, the project will provide insights into Netflix's content strategy and programming trends over time.

The chosen project type is EDA. This involves analyzing and visualizing data to uncover patterns, trends, and insights.

II. Project Objective

The goal of this project is to explore and analyze Netflix's original programming to understand the distribution of genres, trends in release years, and patterns in runtime and episodes. The objective is to present these insights in an interactive and user-friendly manner.

Questions to Answer:

- What are the Netflix release years?
- What are the most popular genres of shows on Netflix?
- What are the top 5 languages of shows on Netflix?
- What is the number of shows released in the past 5 years by language?
- What is the most popular time to release a show?
- What are the shows with the largest number of seasons and episodes?
- What is the distribution of average runtime? Which shows have the minimum/maximum runtime?
- What are the top 10 genres of upcoming shows?
- What are the top 5 languages of upcoming shows?
- What are the top genres of shows in development?

III. Data Description

The dataset will be created by scraping the Wikipedia page List of Netflix Original Programming. The final DataFrame will have the following columns: Title, Genre, Premiere, Seasons, Runtime, Status, and Language.

The data will initially be in HTML table format, which will be converted to a structured DataFrame with seven columns representing various attributes of Netflix originals. The final dataset will be in CSV format with cleaned and structured data.

IV. Methodology

1. **Data Collection:**
   - Use Python libraries such as requests and BeautifulSoup to scrape the data from the Wikipedia page.
   - Convert the scraped HTML data into a structured DataFrame.
2. **Data Cleaning:**
   - Handle missing values/blank values and inconsistent data formats, remove all square brackets.
   - Convert data types as needed (e.g., dates, numerical values), create a release year column
3. **Exploratory Data Analysis:**
   - Descriptive statistics to summarize the dataset.
   - Grouping and aggregation to understand genre distribution and trends over time.
   - Create visualizations such as histograms, bar charts, line graphs, pie charts, and heatmaps to visualize trends and distributions.

V. Expected Deliverables

Interactive Dashboard: A fully streamlit interactive dashboard that allows users to explore the data dynamically. Users will be able to filter and view data by genre, language, and more.

PowerPoint Presentation: A summary presentation highlighting key findings from the EDA, including visualizations and insights.

Github Repository: A public repository with all relevant notebooks, datasets, and a README.md file describing the project and providing a link to the interactive dashboard.

VI. Timeline and Tasks Tasks Breakdown:

Day 1-5: Data Collection - Scrape data from Wikipedia and combine tables into a single DataFrame, Data Cleaning - Clean and preprocess the data.
Day 5-6: Exploratory Data Analysis - Perform initial analysis and generate insights.
Day 7-8: Visualization - Develop interactive dashboards and visualizations.
Day 9: Presentation Preparation - Create a PowerPoint presentation summarizing the findings.

Day 10: Final Review and Submission - Finalize the project, update the GitHub repository, and submit the project.

VII. Potential Challenges Challenges and

Solutions:

1. **Data Scraping Issues:**

   The structure of the Wikipedia page may change, which could affect data extraction and tables on the Wikipedia page may have varying lengths, leading to difficulties in scraping.

   Solution: Regularly check the scraping code and adjust as needed. And implement robust scraping logic to handle different table structures. Use tools like BeautifulSoup and pandas to manage varying row lengths and merge tables accurately.

2. **Interactive Dashboard Development:**

   Creating an engaging and user-friendly dashboard might be challenging.

   Solution: Test different visualization tools and gather feedback to improve usability.