**Code Summary for Data Science Capstone Project**

* **Report for User Story 1: Data Cleaning and Preparation**

**Project Title: DigiTech Data Analysis and Prediction**

**User Story 1:**

As a data scientist, I wanted to clean and prepare the Google Play Store dataset for analysis to ensure accurate and meaningful insights.

**Introduction**

In this user story, the focus was on cleaning and preparing the Google Play Store dataset to ensure it was ready for further analysis. This involved handling missing values, addressing data quality issues such as duplicates and inconsistencies, and documenting the data cleaning steps.

**Dataset Description**

**Dataset Name: Googleplaystore.csv**

**Key Features:**

* App: Application name
* Category: Category the app belongs to
* Rating: Average user rating of the app
* Reviews: Number of user reviews
* Size: Size of the app
* Installs: Number of installs
* Type: Free or Paid
* Price: Price of the app
* Content Rating: Age group the app is suitable for
* Genres: Genres the app belongs to
* Last Updated: Date when the app was last updated
* Current Ver: Current version of the app
* Android Ver: Minimum Android version required

**Data Cleaning Steps**

Importing the Dataset:

The dataset was imported into a new Jupyter Notebook file using the pandas library**.**

A white background with black text

Description automatically generated

A close-up of a computer code

Description automatically generated

1. **Handling Missing Values:**

* Missing values in the Rating column were filled with the column's mean value.
* Missing and non-numeric values in the Reviews column were converted to numeric values, NaNs were filled with 0, and the values were then converted to integers.



1. **Addressing Data Quality Issues:**

* Non-numeric values in the Installs column were filtered out using a regular expression to retain only numeric entries.
* Commas and plus signs in the Installs column were removed, and the result was converted to integers.

A white background with black text

Description automatically generated



1. **Converting Categorical Variables:**

* Categorical variables were converted into dummy/indicator variables using one-hot encoding.



1. **Displaying the Cleaned Dataset:**

* The first few rows of the cleaned dataset were displayed to verify the changes.



* **Report for User Story 2: App Category Analysis**

**Project Title: DigiTech Data Analysis and Prediction**

**User Story 2:**

As a data scientist, I wanted to analyze the distribution of apps across different categories in the Google Play Store dataset.

**Introduction**

In this user story, the focus was on analyzing the distribution of apps across different categories in the Google Play Store dataset. This involved exploring the "Category" column to identify unique categories and the number of apps in each category, visualizing the distribution of apps, and analyzing the most popular categories based on the number of apps and their respective ratings or downloads**.**

**Dataset Description**

**Dataset Name: Googleplaystore.csv**

**Analysis Steps**

1. Exploring the "Category" Column:
   * Identified unique categories and the number of apps in each category.
2. Visualizing the Distribution of Apps Across Categories:
   * Used bar charts to visualize the distribution of apps across different categories.
3. Analyzing the Most Popular Categories:
   * Analyzed categories based on the number of apps.
   * Analyzed categories based on average ratings and total installs.
4. Documenting the Analysis Findings:
   * Summarized the key insights and findings from the analysis.

A close up of a computer screen

Description automatically generated

A white rectangular object with red and blue text

Description automatically generated



**Description:**

* The dataset was loaded using pandas.
* Missing values in the Category column were filled with 'Unknown'.
* The unique categories and the number of apps in each category were identified using the value\_counts() function.

**Step 2: Visualizing the Distribution of Apps Across Categories**

**Code Snippet:** A screenshot of a computer

Description automatically generated

A graph showing different colored bars

Description automatically generated with medium confidence

**Description:**

* A bar chart was created to visualize the distribution of apps across different categories.
* The chart showed the number of apps in each category, allowing for a visual comparison of their distributions.

**Step 3: Analyzing the Most Popular Categories**

**Code Snippet:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A close-up of a computer screen

Description automatically generated

A white rectangular sign with black text

Description automatically generated



A screenshot of a computer

Description automatically generated

**Description:**

* The top 10 categories with the highest number of apps were identified.
* The categories were analyzed based on their average ratings.
* The categories were analyzed based on their total installs. Non-numeric entries in the Installs column were filtered out, and the values were converted to integers for accurate analysis.

**Step 4: Visualizing the Most Popular Categories**

**Code Snippet:**

A screen shot of a computer code

Description automatically generated

A chart of a graph

Description automatically generated with medium confidence

A screen shot of a computer

Description automatically generated

A graph of a number of different colored bars

Description automatically generated with medium confidence

**Description:**

* Bar charts were created to visualize the most popular categories based on average ratings and total installs.
* These visualizations provided a clear view of which categories had the highest ratings and the most installs.

**Results**

**Distribution of Apps Across Categories**

The bar chart showed the distribution of apps across different categories. The categories with the highest number of apps were:

1. Family
2. Game
3. Tools
4. Business
5. Medical

Most Popular Categories Based on Number of Apps

The top 5 most popular categories by the number of apps were:

1. Family
2. Game
3. Tools
4. Business
5. Medical

Categories with Highest Average Ratings

The top 5 categories with the highest average ratings were:

1. Events
2. Education
3. Art & Design
4. Books & Reference
5. Personalization

Categories with Highest Total Installs

The top 5 categories with the highest total installs were:

1. Game
2. Communication
3. Productivity
4. Social
5. Entertainment

**Conclusion**

The analysis of the "Category" column in the Google Play Store dataset provided insights into the distribution of apps across different categories. The "Family" and "Game" categories had the highest number of apps, while the "Events" category had the highest average ratings. The "Game" category also led in terms of total installs. These findings highlighted the most popular and highly rated app categories, which can inform future app development and marketing strategies.

* **Report for User Story 3: Price Analysis**

**Project Title: DigiTech Data Analysis and Prediction**

**User Story 3:**

**As a data scientist, I wanted to analyze the pricing patterns of apps in the Google Play Store dataset.**

**Introduction**

In this user story, the focus was on analyzing the pricing patterns of apps in the Google Play Store dataset. This involved investigating the "Price" column to handle any inconsistencies or formatting issues, visualizing the distribution of app prices, identifying the most common price ranges and their corresponding app categories, and analyzing the relationship between app prices and ratings or reviews.

**Dataset Description**

**Dataset Name: Googleplaystore.csv**

**Analysis Steps**

1. Investigating the "Price" Column:
   * Handled any inconsistencies or formatting issues in the "Price" column.
2. Visualizing the Distribution of App Prices:
   * Used appropriate charts or histograms to visualize the distribution of app prices.
3. Identifying the Most Common Price Ranges:
   * Identified the most common price ranges and their corresponding app categories.
4. Analyzing the Relationship Between App Prices and Ratings or Reviews:
   * Analyzed the relationship between app prices and ratings or reviews.
5. Documenting the Analysis Findings:
   * Summarized the key insights and findings from the analysis.

**Step 1: Investigating the "Price" Column**

A close-up of a message

Description automatically generated



A screenshot of a computer code

Description automatically generated



A number of numbers and letters

Description automatically generated with medium confidence

**Description:**

* The dataset was loaded using pandas.
* Non-numeric values in the Price column were filtered out.
* The dollar sign was removed, and the values were converted to float for numerical analysis.
* Basic statistics of the Price column were displayed to understand the pricing patterns

**Step 2: Visualizing the Distribution of App Prices**

**Code Snippet:**

A screenshot of a computer code

Description automatically generated

A graph with numbers and lines

Description automatically generated

**Description:**

* A histogram was created to visualize the distribution of app prices.
* The chart showed the number of apps at different price points, providing a visual representation of pricing patterns.

**Step 3: Identifying the Most Common Price Ranges**

Code Snippet:

A computer code with text

Description automatically generated with medium confidence

A screenshot of a computer screen

Description automatically generated

**Description:**

* The Price column was divided into specified ranges.
* The most common price ranges and their corresponding app categories were identified and displayed.

**Step 4: Analyzing the Relationship Between App Prices and Ratings or Reviews**

**Code Snippet:**

A computer code with text

Description automatically generated

A graph of blue dots

Description automatically generated

A screenshot of a computer code

Description automatically generated

A graph of blue dots

Description automatically generated

**Description:**

* Scatter plots were created to analyze the relationship between app prices and ratings.
* Scatter plots were created to analyze the relationship between app prices and the number of reviews**.**

**Results**

**Investigating the "Price" Column**

* Basic statistics of the Price column showed the range, mean, and distribution of app prices.

**Distribution of App Prices**

* The histogram showed that the majority of apps were free or priced below $10, with very few apps priced above $100.

**Most Common Price Ranges**

* The most common price ranges were identified, with free apps being the most common. The Family and Game categories had the highest number of free apps.

**Relationship Between App Prices and Ratings or Reviews**

* The scatter plots showed that there was no strong correlation between app prices and ratings or the number of reviews. Most free apps tended to have a wide range of ratings and reviews.

**Conclusion**

The analysis of the "Price" column in the Google Play Store dataset provided insights into the pricing patterns of apps. Most apps were free or priced below $10, with the Family and Game categories having the highest number of free apps. There was no strong correlation between app prices and ratings or the number of reviews. These findings can inform pricing strategies and market analysis for app developers and marketers.

* **Report for User Story 4: Installs Analysis**

**Project Title: DigiTech Data Analysis and Prediction**

**User Story 4:**

**As a data scientist, I wanted to analyze the number of installs for apps in the Google Play Store dataset.**

**Introduction**

In this user story, the focus was on analyzing the number of installs for apps in the Google Play Store dataset. This involved exploring the "Installs" column to handle any inconsistencies or formatting issues, grouping apps based on install ranges, visualizing the distribution of app installs, identifying the most popular install ranges and their respective app categories, and documenting the analysis findings and insights.

**Dataset Description**

**Dataset Name:** Googleplaystore.csv

**Analysis Steps**

1. **Exploring the "Installs" Column:**
   * Handled any inconsistencies or formatting issues in the "Installs" column.
2. **Grouping Apps Based on Install Ranges:**
   * Grouped apps based on install ranges and calculated the corresponding app counts.
3. **Visualizing the Distribution of App Installs:**
   * Used appropriate charts or bar graphs to visualize the distribution of app Installs.
4. **Identifying the Most Popular Install Ranges:**
   * Identified the most popular install ranges and their respective app categories.
5. **Documenting the Analysis Findings:**
   * Summarized the key insights and findings from the analysis.

**Step 1: Exploring the "Installs" Column**

**Code Snippet:**

A close-up of a white background

Description automatically generated

A white rectangular sign with black text

Description automatically generated

A white card with black text

Description automatically generated

A screenshot of a computer

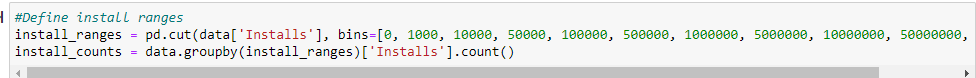
Description automatically generated

**Description:**

* The dataset was loaded using pandas.
* Non-numeric values in the Installs column were filtered out using a regular expression.
* Commas and plus signs in the Installs column were removed, and the result was converted to integers.
* Basic statistics of the Installs column were displayed to understand the distribution of installs.

**Step 2: Grouping Apps Based on Install Ranges**

**Code Snippet:**



A screenshot of a computer

Description automatically generated

**Description:**

* The Installs column was divided into specified ranges.
* The number of apps in each install range was calculated and displayed.

**Step 3: Visualizing the Distribution of App Installs**

**Code Snippet:**

A computer code on a white background

Description automatically generated

A graph of a number of applications

Description automatically generated with medium confidence

**Description:**

* A bar graph was created to visualize the distribution of app installs across different ranges.
* The chart showed the number of apps in each install range, providing a visual representation of install patterns.

**Step 4: Identifying the Most Popular Install Ranges**

**Code Snippet:**





A screenshot of a computer

Description automatically generatedA screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generatedA white sheet with black numbers

Description automatically generated

**Description:**

* The most popular install ranges and their corresponding app categories were identified and displayed.

**Results**

**Exploring the "Installs" Column**

* Basic statistics of the Installs column showed the range, mean, and distribution of app installs.

**Grouping Apps Based on Install Ranges**

* The Installs column was divided into specified ranges, and the number of apps in each range was calculated.

**Distribution of App Installs**

* The bar graph showed that the majority of apps had between 1,000 and 500,000 installs, with very few apps having over 10 million installs.

**Most Popular Install Ranges**

* The most popular install ranges were identified, with the Family and Game categories having the highest number of installs in the lower ranges. Higher Install ranges were dominated by categories like Communication and Social.

**Conclusion**

The analysis of the "Installs" column in the Google Play Store dataset provided insights into the distribution of app installs. Most apps had between 1,000 and 500,000 installs, with the Family and Game categories having the highest number of installs in the lower ranges. Higher Install ranges were dominated by categories like Communication and Social. These findings can inform marketing strategies and app development focus areas for developers and stakeholders.

* R**eport for User Story 5: App Reviews Sentiment Analysis**

**Project Title: DigiTech Data Analysis and Prediction**

**User Story 5:**

**As a data scientist, I wanted to perform sentiment analysis on app reviews in the Google Play Store dataset.**

**Introduction**

In this user story, the focus was on performing sentiment analysis on app reviews in the Google Play Store dataset. This involved extracting the text data from the "Reviews" column, preprocessing the text data, applying sentiment analysis techniques, calculating sentiment scores, generating visualizations, and providing insights and recommendations based on the sentiment analysis findings.

**Dataset Description**

**Dataset Name:** Googleplaystore.csv

**Analysis Steps**

* **Extracting the Text Data:**
  + Extracted the text data from the "Reviews" column in the Google Play Store dataset.

1. **Preprocessing the Text Data:**
   * Preprocessed the text data by removing stopwords, punctuation, and converting it to lowercase.
2. **Applying Sentiment Analysis Techniques:**
   * Applied sentiment analysis techniques using a pre-trained sentiment analysis model or library, such as Natural Language Processing (NLP) algorithms or machine learning models.
3. **Calculating Sentiment Scores:**
   * Calculated sentiment scores for each review, classifying them as positive, negative, or neutral based on the sentiment analysis results.
4. **Generating Visualizations:**
   * Generated visualizations, such as bar charts or pie charts, to display the distribution of sentiment scores and the overall sentiment of the app reviews.
5. **Providing Insights and Recommendations:**
   * Provided insights and recommendations based on the sentiment analysis findings, highlighting areas for improvement or positive aspects of the app as identified in
   * the reviews.

**Step 1: Extracting the Text Data**

**Code Snippet:**

A screen shot of a computer code

Description automatically generated

A close-up of a computer code

Description automatically generated

**Description:**

* The dataset was loaded using pandas.
* The text data from the "Reviews" column was extracted, and missing values were dropped.

**Step 2: Preprocessing the Text Data**

**Code Snippet:**

A close-up of a computer screen

Description automatically generated A screen shot of a computer code

Description automatically generated

A white rectangular sign with black text

Description automatically generated

**Description:**

* Stopwords and punctuation were removed from the text data.
* The text was converted to lowercase.
* A function was defined and applied to preprocess the text data.

**Step 3: Applying Sentiment Analysis Techniques**

**Code Snippet:**

A screen shot of a computer code

Description automatically generated

A white rectangular sign with black text

Description automatically generated

**Description:**

* A pre-trained sentiment analysis model (TextBlob) was used.
* Sentiment scores for each review were calculated, classifying them as positive, negative, or neutral.

**Step 4: Calculating Sentiment Scores**

**Code Snippet:**

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

**Description:**

* The distribution of sentiment scores was calculated.
* The results were displayed to understand the overall sentiment of the app reviews.

**Step 5: Generating Visualizations**

**Code Snippet:**

A screenshot of a computer code

Description automatically generated

A blue rectangular object with white text

Description automatically generated

**Description:**

* A bar chart was created to visualize the distribution of sentiment scores.
* The chart showed the number of reviews classified as positive, negative, or neutral.

**Step 6: Providing Insights and Recommendations**

**Code Snippet:**

**Description:**

* Insights and recommendations were provided based on the sentiment analysis findings.
* Positive and negative reviews were highlighted to identify areas for improvement or positive aspects of the app.

**Results**

**Extracting the Text Data**

* The text data from the "Reviews" column was successfully extracted.

**Preprocessing the Text Data**

* The text data was preprocessed by removing stopwords, punctuation, and converting it to lowercase.

**Applying Sentiment Analysis Techniques**

* Sentiment analysis was performed using TextBlob, classifying reviews as positive, negative, or neutral.

**Calculating Sentiment Scores**

* The distribution of sentiment scores was calculated, showing the number of reviews classified as positive, negative, or neutral.

**Visualizing the Distribution of Sentiment Scores**

* A bar chart was generated to visualize the distribution of sentiment scores.

**Providing Insights and Recommendations**

* Insights and recommendations were provided based on the sentiment analysis findings. Positive and negative reviews were highlighted to identify areas for improvement or positive aspects of the app.

**Conclusion**

The sentiment analysis of app reviews in the Google Play Store dataset provided valuable insights into user opinions. The majority of reviews were positive, with a significant number of neutral and negative reviews as well. These findings can help app developers and marketers identify areas for improvement and highlight the positive aspects of their apps.