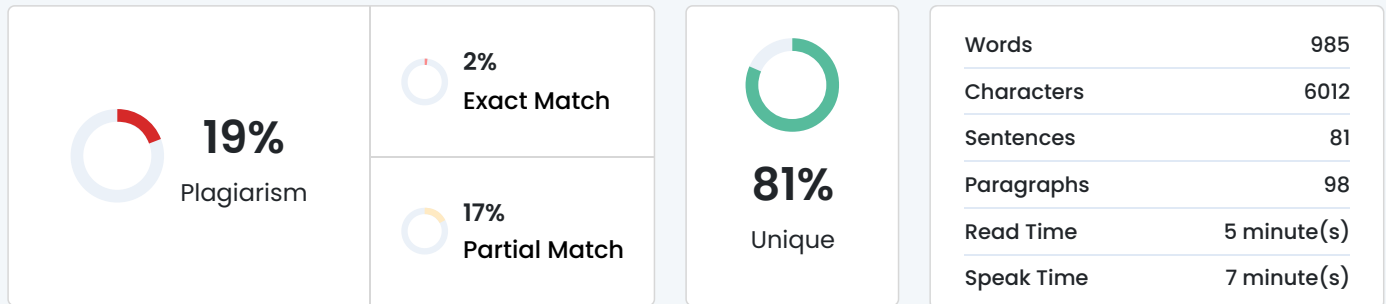


Plagiarism Scan Report



Content Checked For Plagiarism

4. Checking the Data-Randomly (Sampling Technique)

The code shown in the image is performing random sampling on a Data Frame `df` using Pandas in Python.
Explanation:

- `df.sample(...)`: This method is used to randomly sample rows from the Data Frame `df`.
- `frac=0.1`: This means that 10% of the rows from the Data Frame will be randomly sampled.
- `random_state=166`: This sets a seed for the random number generator to ensure reproducibility — the same rows will be sampled every time this code is run.
- `.head()`: This returns the first 5 rows of the sampled data.

Purpose:

This is likely being done for data exploration or debugging, allowing the user to inspect a small, random subset of the data without working with the full dataset.

5. Taking a copy of the dataset

The code in the image is doing data inspection

This creates a copy of the original Data Frame `df` and assigns it to `new_df`. This is a safety step to avoid altering the original data.

- `main_column` stores the list of all column names.
- `data_show_case` is an empty dictionary where results will be stored.

This loop does the following for each column in the Data Frame:

- If the number of unique values in the column is more than 10 → label it as "Greater 10 values".
- Otherwise → display the actual unique values (using `unique()`).
- Converts the `data_show_case` dictionary into a list of items (column name and values).
- Creates a new Data Frame (`df_show_case`) with two columns:
 - o "Column Name": The name of the column
 - o "Unique Values": Either the list of unique values or the string "Greater 10 values"

This helps in quickly identifying which columns are:

- Categorical/Boolean/Discrete (with limited values)
- Continuous/Numerical/High cardinality (with many distinct values)

It helps you understand the structure of the dataset.

- Which columns may need encoding (like Gender, Payment, etc.)
- Which ones might be IDs or irrelevant (AccountID)
- Which ones may need binning or transformation.

6. Replacing all the unwanted values:

The code in the image is doing data cleaning and transformation on a Data Frame (`new_df`) to prepare it for further analysis or modeling.

- Fixing gender labels: Replaces short gender codes ('F', 'M') with full forms ('Female', 'Male').
- Fixing account user count: Replaces invalid/incorrect symbol @ with the value '2'.
- Standardizing account segment labels: Merges different variations of segment labels like 'Regular +' and 'Super +' into a single label: 'Super Plus'.
- Fixing login device values: Replaces the invalid &&&& with the mode (most frequent value) of the Login_device column.
- Handling missing or placeholder values in revenue: Replaces # (a placeholder) with NaN (missing value). Then fills those missing values with the mode (most frequent) value from the original Data Frame df.
- Fixing other miscellaneous placeholder characters:

Replaces:

with 3

* with 4

% with 8

This may apply across the entire Data Frame (not limited to a column).

It shows the updated unique values per column, after all the cleaning steps.

- Cleans inconsistent or invalid values (@, #, &&&&, %, etc.)
- Standardizes category labels
- Prepares the dataset for analysis or modeling (ensuring reliable data types and values)

The code and chart in the image are performing exploratory data analysis (EDA) using pie charts to visualize two categorical distributions:

```
fig, ax = plt.subplots(1, 2, figsize=(12, 6))
```

- Creates a figure with 1 row and 2 columns of subplots.
- figsize=(12, 6) makes the entire figure 12 inches wide and 6 inches tall.
- ax is a list of axes: ax[0] is the left pie chart, ax[1] is the right pie chart.
- Sets the title "Churn Ratio" on the first subplot.
- Uses the values from new_df['Churn'].value_counts() to plot the pie:

0 is mapped to 'Not Churn'

1 is mapped to 'Churn'

- autopct="%.2f%%" shows percentages on the pie slices (e.g., 83.16%).
- Sets the title "Gender Distribution" on the second subplot.
- Uses value_counts() to:

Get values (counts)

Get labels (like 'Male', 'Female')

- Displays percentages on slices again.

```
plt.show()
```

- Displays both plots side-by-side in one visual output.

Quickly understand class imbalance in the Churn label.

The distribution of genders, useful for segmentation or demographic analysis.

Chart Summary

Left Pie Chart: Churn Ratio

- Not Churn: 83.16% (majority of users)
- Churn: 16.84% (customers who left)

Right Pie Chart: Gender Distribution

- Male: 60.5%
- Female: 39.5%

Checking Churn value: count new_df['Churn'].value_counts ()

Churn Count

0 9364

1 1896

This Python code is using Seaborn and Matplotlib to generate a count plot showing the distribution of customer churn by gender.

- plt.title("Churn Distribution by Gender") Sets the title of the plot

Creates a count plot:

- `x='Gender'`: Categories on the x-axis are Male and Female.
- `hue=new_df['Churn'].astype(str)`: Bars are coloured based on churn status (converted to str for plotting).
- `palette='viridis'`: Uses the "viridis" color palette.
- Labels for the x and y axes.
- Sets a custom legend:

Title: "Churn"

Labels: "Not Churned" and "Churned"

The bar plot shows the number of churned and not churned customers, separated by gender:

- Two groups on the x-axis: Female and Male.
- Each group has two bars:
 - o Darker color = Not Churned
 - o Lighter green = Churned (colors based on the viridis palette).

This code uses Seaborn and Matplotlib to visualize the distribution of customer account tenure (in months) using a histogram with a KDE (Kernel Density Estimate) curve.

- `matplotlib.rcParams['figure.figsize'] = (8,8)` Sets the figure size to 8x8 inches.
- `plt.title('Tenure of Account in Months')` Adds a title to the plot.
- `sns.histplot(data=new_df, x='Tenure', kde=True, palette='pastel')`

Creates a histogram of the Tenure column:

`x='Tenure'`: The x-axis represents account tenure (in months).

`kde=True`: Adds a KDE (smoothed density) curve on top of the histogram.

`palette='pastel'`: Tries to use a pastel color palette (though palette isn't used in histplot; might be ignored).

Matched Source

Similarity 2%

Title: [How to check if a dictionary is empty? – Stack Overflow](#)

Apr 20, 2014 ♦ All the empty dictionaries will be classified as false-negatives. If the dictionary is not empty, then the code will process only one key and...Missing: • `data_show_case` | Show results with:

<https://stackoverflow.com/questions/23177439/how-to-check-if-a-dictionary-is-empty>

Similarity 2%

Title: [Using Multiple Subplots in Visualise data using Matplotlib – CodeChef](#)

The most common way to create subplots is using `plt.subplots()`. You can create a grid of subplots / stack subplots vertically. This creates a figure with 1 row and 2 columns of subplots.

<https://www.codechef.com/learn/course/matplotlib/LMPL03/problems/LMPLPR18>

Similarity 2%

Title: [How to Plot Value Counts in Pandas | GeeksforGeeks](#)

Jul 22, 2024 ♦ In this article, we'll learn how to plot value counts using `provide`, which can help us quickly understand the frequency distribution of values in a dataset. Missing: `new_df['Churn']`

<https://www.geeksforgeeks.org/how-to-plot-value-counts-in-pandas>

Similarity 2%

Title: [How to Create Multiple Subplots in Python Using Matplotlib and Pandas](#)

Apr 26, 2025 · `df.plot(x='Date', y='Profit', ax=axes[1])` plots the Profit column against the Date column on the second subplot (index 1). `axes[1].set_title('Profit Over Time')` sets the title of the second subplot. Adjust Layout

<https://iifx.dev/en/articles/171016063>

Similarity 2%

Title:[How to set the hue range for a numeric variable using a colored ...](#)

Mar 17, 2020 ♦ How to set the hue range for a numeric variable using a colored bubble plot in seaborn, python? Ask Question. Asked 5 years ago. Modified 5♦...Missing: new_df[Churn .astype str Bars churn (converted

<https://stackoverflow.com/questions/60721598/how-to-set-the-hue-range-for-a-numeric-variable-using-a-colored-bubble-plot-in-s/60723957>

Similarity 2%

Title:[Guide to Churn Prediction : Part 5— Graphical analysis - Mage AI](#)

Feb 9, 2022 ♦ The length of each bar represents the number of values in ... Yes represents churned customers, while No represents non-churned customers.Missing: shows | Show results with:

<https://m.mage.ai/guide-to-churn-prediction-part-5-graphical-analysis-c4a357e66408>

Similarity 2%

Title:[Custom Matplotlib Title - Python Graph Gallery](#)

This post aims to explain how to add a title to your python chart, and how to customise it. This is done through the 'title()' function of matplotlib.Missing: Tenure Months)

<https://python-graph-gallery.com/190-custom-matplotlib-title>

Similarity 2%

Title:[Pandas DataFrame: Shuffle a given DataFrame rows - w3resource](#)

May 1, 2025 ♦ This code shuffles the rows of the Pandas DataFrame df randomly using the sample method with frac=1, which means to sample all rows.

<https://www.w3resource.com/python-exercises/pandas/python-pandas-data-frame-exercise-40.php>

Similarity 2%

Title:[Legacy pie chart - MATLAB - MathWorks](#)

Axes object. Use ax to plot the pie chart in a specific axes instead of the current axes (gca).

<https://www.mathworks.com/help/matlab/ref/pie.html>
