Data Cleaning for Decision-Making

22/03/2025

stratified_history.csv
8.1 MB

Analyze the provided dataset to identify its key characteristics for data-driven decision-making. Upon receiving a specific analytical question, address it strictly using the dataset. Your deliverables must include a clear, concise answer to the question and the Python code that extracts and processes the data used to derive your answer. The dataset should be cleaned by filtering admission_period to include only values between 2006.1 and 2019.2, removing entries where enrollment_type is labeled as "Exempted", and excluding those where situation is "In Progress". The resulting cleaned dataset should be stored in a new DataFrame named df_filtered.

qwen-max-latest

To address your request, I will perform the following steps:

1. **Load the dataset** : Read the CSV file into a DataFrame.
2. **Clean the dataset** :
   - Filter `admission_period` to include only values between 2006.1 and 2019.2.
   - Remove entries where `enrollment_type` is labeled as "Exempted".
   - Exclude rows where `situation` is "In Progress".
3. **Store the cleaned dataset** : Save the filtered data in a new DataFrame named `df_filtered`.
4. **Provide Python code** : Include the code used for the above operations.

Here is the Python code to achieve this:

python
```python
import pandas as pd

# Step 1: Load the dataset
df = pd.read_csv('stratified_history.csv')

# Step 2: Clean the dataset
# Filter admission_period to include only values between 2006.1 and 2019.2
df_filtered = df[(df['admission_period'] >= 2006.1) & (df['admission_period'] <= 2019.2)]

# Remove entries where enrollment_type is labeled as "Exempted"
df_filtered = df_filtered[df_filtered['enrollment_type'] != 'Exempted']

# Exclude rows where situation is "In Progress"
df_filtered = df_filtered[df_filtered['situation'] != 'In Progress']

# Step 3: Display basic information about the cleaned dataset
print("Shape of the cleaned dataset:", df_filtered.shape)
print("\nColumns in the cleaned dataset:\n", df_filtered.columns.tolist())
print("\nFirst few rows of the cleaned dataset:\n", df_filtered.head())

# The cleaned dataset is now stored in df_filtered
```

Explanation of the Code

1. **Loading the dataset** : The `pd.read_csv()` function reads the CSV file into a Pandas DataFrame.
2. **Filtering `admission_period`** : A logical condition is applied to retain only rows where `admission_period` falls within the specified range.
3. **Removing "Exempted" entries** : Rows with `enrollment_type` equal to "Exempted" are excluded using a logical condition.
4. **Excluding "In Progress" situations** : Rows with `situation` equal to "In Progress" are removed similarly.
5. **Resulting dataset** : The cleaned dataset is stored in `df_filtered`.

Key Characteristics of the Cleaned Dataset

After running the code, you can inspect the shape, columns, and sample rows of the cleaned dataset to understand its structure. This cleaned dataset can now be used for further data-driven decision-making.

Let me know if you have a specific analytical question or need further analysis on this dataset!

Using appropriate statistical models (e.g., ANOVA, multiple regression), evaluate whether the type of admission affects the number of semesters required for graduation while controlling for confounding factors such as socioeconomic status and program choice.

qwen-max-latest

To evaluate whether the type of admission affects the number of semesters required for graduation while controlling for confounding factors such as socioeconomic status and program choice, we will follow these steps:

1. Problem Breakdown

- **Dependent Variable (Outcome)** : `semesters_to_graduate` (number of semesters required for graduation).
- **Independent Variable (Predictor of Interest)** : `admission_type` (e.g., Entrance Exam, Unified Selection System (SISU)).
- **Confounding Variables** :
  - Socioeconomic status indicators (`quota_reservation` , `school_background` , `income`).
  - Program choice (`academic_unit` or similar program-related variables).
- **Statistical Model** :
  - Use **multiple regression** to assess the effect of `admission_type` on `semesters_to_graduate` while controlling for confounders.
  - Optionally, use **ANOVA** to test whether there are significant differences in graduation times across admission types.

2. Data Preparation

From the dataset description, we need to clean and prepare the data for analysis:

- Filter the cleaned dataset (`df_filtered`) from the previous cleaning steps.
- Ensure that all relevant variables (`semesters_to_graduate` , `admission_type` , confounders) are properly encoded and ready for modeling.

3. Statistical Approach

Step 1: Exploratory Data Analysis (EDA)

- Check the distribution of `semesters_to_graduate` to identify potential outliers or skewness.
- Explore the relationship between `admission_type` and `semesters_to_graduate` using summary statistics and visualizations (e.g., boxplots).

Step 2: Multiple Regression

- Fit a multiple regression model where:

$$\text{semesters\_to\_graduate} = \beta_0 + \beta_1(\text{admission\_type}) + \beta_2(\text{confounders}) + \epsilon$$

- Interpret the coefficient β_1 to understand the effect of `admission_type` on graduation time, adjusted for confounders.

Step 3: ANOVA

- Perform an ANOVA test to determine if there are statistically significant differences in mean graduation times across different `admission_type` categories.

4. Python Code Implementation

Here is the Python code to perform the analysis:

python

```python
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
from statsmodels.stats.anova import anova_lm
import matplotlib.pyplot as plt
import seaborn as sns

# Step 1: Load the cleaned dataset
# Assuming df_filtered is already created from the previous steps
# df_filtered = ...

# Step 2: Exploratory Data Analysis (EDA)
# Check the distribution of semesters_to_graduate
plt.figure(figsize=(8, 6))
sns.histplot(df_filtered['semesters_to_graduate'], kde=True, bins=20)
plt.title('Distribution of Semesters to Graduate')
plt.xlabel('Semesters to Graduate')
plt.ylabel('Frequency')
plt.show()

# Boxplot of semesters_to_graduate by admission_type
plt.figure(figsize=(10, 6))
sns.boxplot(x='admission_type', y='semesters_to_graduate', data=df_filtered)
plt.title('Semesters to Graduate by Admission Type')
plt.xlabel('Admission Type')
plt.ylabel('Semesters to Graduate')
plt.xticks(rotation=45)
plt.show()

# Step 3: Multiple Regression
# Define the regression formula
formula = 'semesters_to_graduate ~ C(admission_type) + C(quota_reservation) + school_background

# Fit the model
model = smf.ols(formula, data=df_filtered).fit()

# Print the regression summary
print(model.summary())

# Step 4: ANOVA
# Perform ANOVA to test differences in semesters_to_graduate across admission types
anova_results = anova_lm(model)
print("\nANOVA Results:")
print(anova_results)
```

```python
print(anova_results)

# Step 5: Interpretation
# Extract the p-value for admission_type from the ANOVA table
p_value_admission_type = anova_results.loc['C(admission_type)', 'PR(>F)']
print(f"\nP-value for admission_type: {p_value_admission_type}")

# Conclusion based on p-value
if p_value_admission_type < 0.05:
    print("There is a statistically significant difference in semesters to graduate across admis
else:
    print("No statistically significant difference in semesters to graduate across admission typ
```

5. Explanation of the Code

1. **EDA** :
   - Visualize the distribution of `semesters_to_graduate` to check for normality or skewness.
   - Use a boxplot to compare graduation times across different `admission_type` categories.
2. **Multiple Regression** :
   - The formula specifies `semesters_to_graduate` as the dependent variable and includes `admission_type` as the main predictor of interest, along with confounders (`quota_reservation` , `school_background` , `income`).
   - The `C()` function is used to treat categorical variables as factors.
3. **ANOVA** :
   - The ANOVA test evaluates whether the mean graduation times differ significantly across `admission_type` categories.
4. **Interpretation** :
   - If the p-value for `admission_type` is less than 0.05, it indicates a statistically significant association between admission type and graduation time, even after controlling for confounders.

6. Expected Output

- **Regression Summary** : Provides coefficients, standard errors, and p-values for each predictor.
- **ANOVA Results** : Tests the overall significance of `admission_type` .
- **Conclusion** : Based on the p-value, determines whether admission type significantly affects graduation time.

7. Limitations

- The analysis assumes that all relevant confounders are included in the model.
- Missing data in key variables may bias the results.
- Non-linear relationships or interactions between variables are not explicitly modeled here but can be explored further.

Let me know if you'd like to refine this analysis or explore additional aspects!