Name-Vedant Dhoke Div:D15C Roll No-9 A.Y.-2024-25

DS LAB 4

Aim: Implementation of Statistical Hypothesis Test using Scipy and Sci-kit learn.

Problem Statement: Perform the following Tests:Correlation Tests:
a) Pearson's Correlation Coefficient
b) Spearman's Rank Correlation
c) Kendall's Rank Correlation
d) Chi-Squared Test

a) Pearson's Correlation Coefficient

```python
from scipy.stats import pearsonr

excluded_columns = ['Unnamed: 0']
numeric_cols = [col for col in train_df.select_dtypes(include=['number']).columns if col not in excluded_columns]

# Compute Pearson correlation
print("\nPearson’s Correlation Coefficient:")
for i in range(len(numeric_cols)):
    for j in range(i + 1, len(numeric_cols)):
        col1, col2 = numeric_cols[i], numeric_cols[j]
        corr, _ = pearsonr(train_df[col1], train_df[col2])
        print(f"Pearson correlation between {col1} and {col2}: {corr:.4f}")
```

```
Pearson’s Correlation Coefficient:
Pearson correlation between id and popularity: 0.0736
Pearson correlation between id and vote_average: -0.5373
Pearson correlation between id and vote_count: 0.1149
Pearson correlation between popularity and vote_average: -0.2973
Pearson correlation between popularity and vote_count: 0.2058
Pearson correlation between vote_average and vote_count: -0.6040
```

This calculates the Pearson correlation coefficient between numeric columns in your dataset, excluding the "Unnamed: 0" column. It first filters out numerical columns and then iterates over each pair to compute their correlation using pearsonr from scipy.stats.

Pearson correlation measures the linear relationship between two variables, with values ranging from -1 (strong negative correlation) to 1 (strong positive correlation). The output suggests that id has little correlation with other numerical features, while popularity and vote_count show a positive correlation, meaning more votes generally indicate higher popularity.

On the other hand, vote_average and vote_count have a negative correlation, implying that a higher number of votes does not always lead to a higher average rating. This analysis helps in understanding how different numerical features relate to each other in your dataset.

b) Spearman's Rank Correlation

```
from scipy.stats import spearmanr

# Compute Spearman correlation
print("\nSpearman's Rank Correlation:")
for i in range(len(numeric_cols)):
    for j in range(i + 1, len(numeric_cols)):
        col1, col2 = numeric_cols[i], numeric_cols[j]
        corr, _ = spearmanr(train_df[col1], train_df[col2])
        print(f"Spearman correlation between {col1} and {col2}: {corr:.4f}")
```

```
Spearman's Rank Correlation:
Spearman correlation between id and popularity: -0.1665
Spearman correlation between id and vote_average: -0.3710
Spearman correlation between id and vote_count: 0.0849
Spearman correlation between popularity and vote_average: -0.5039
Spearman correlation between popularity and vote_count: 0.6677
Spearman correlation between vote_average and vote_count: -0.8222
```

This calculates Spearman's rank correlation coefficient between numeric columns in your dataset. Spearman's correlation measures the strength and direction of the monotonic relationship between variables, making it useful for detecting non-linear associations.

The output shows that popularity and vote_count have a strong positive correlation, meaning that as one increases, the other generally does too. However, vote_average and vote_count have a strong negative correlation, suggesting that movies with more votes tend to have lower average ratings.

The correlation between popularity and vote_average is also negative, indicating that more popular movies do not necessarily have higher ratings. Compared to Pearson's correlation, Spearman's approach considers ranks rather than absolute values, making it more robust against outliers.

c) Kendall’s Rank Correlation

```
from scipy.stats import kendalltau

# Compute Kendall correlation
print("\nKendall’s Rank Correlation:")
for i in range(len(numeric_cols)):
    for j in range(i + 1, len(numeric_cols)):
        col1, col2 = numeric_cols[i], numeric_cols[j]
        corr, _ = kendalltau(train_df[col1], train_df[col2])
        print(f"Kendall correlation between {col1} and {col2}: {corr:.4f}")
```

```
Kendall’s Rank Correlation:
Kendall correlation between id and popularity: -0.0943
Kendall correlation between id and vote_average: -0.2485
Kendall correlation between id and vote_count: -0.0014
Kendall correlation between popularity and vote_average: -0.3038
Kendall correlation between popularity and vote_count: 0.4383
Kendall correlation between vote_average and vote_count: -0.6656
```

This calculates Kendall’s rank correlation coefficient between numeric columns in your dataset. Kendall’s correlation measures the strength and direction of the ordinal association between two variables by considering the concordance of ranked pairs.

The output shows that popularity and vote_count have a moderate positive correlation, meaning that as popularity increases, vote count tends to increase as well. However, vote_average and vote_count have a strong negative correlation, indicating that movies with higher vote counts tend to have lower average ratings.

The correlation between popularity and vote_average is also negative, suggesting that popular movies do not necessarily receive higher ratings. Kendall’s correlation is more robust than Spearman’s for small datasets and is particularly useful when dealing with ordinal data or rankings.

d) Chi-Squared Test

```python
from scipy.stats import chi2_contingency
import pandas as pd

categorical_columns = train_df.select_dtypes(include=['object']).columns

print("\nChi-Squared Test for Categorical Variables:")
for i in range(len(categorical_columns)):
    for j in range(i + 1, len(categorical_columns)):  # Avoid duplicate comparisons
        col1, col2 = categorical_columns[i], categorical_columns[j]
        contingency_table = pd.crosstab(train_df[col1], train_df[col2])
        chi2, p, _, _ = chi2_contingency(contingency_table)
        print(f"Chi-Squared test between {col1} and {col2}: Chi2 = {chi2:.4f}, p-value = {p:.4f}")
```

```
Chi-Squared Test for Categorical Variables:
Chi-Squared test between original_title and original_language: Chi2 = 12072.0000, p-value = 0.0000
Chi-Squared test between original_title and release_date: Chi2 = 54324.0000, p-value = 0.0000
Chi-Squared test between original_title and media_type: Chi2 = 0.0000, p-value = 1.0000
Chi-Squared test between original_language and release_date: Chi2 = 12072.0000, p-value = 0.0000
Chi-Squared test between original_language and media_type: Chi2 = 4888.1157, p-value = 0.0000
Chi-Squared test between release_date and media_type: Chi2 = 0.0000, p-value = 1.0000
```

The given code performs a Chi-Squared test for independence between categorical variables in the dataset. This statistical test helps determine whether two categorical variables are significantly associated or independent.

The test works by comparing the observed frequencies of occurrences with the expected frequencies under the assumption of independence. A low p-value (< 0.05) suggests a significant relationship between the variables, while a high p-value (≥ 0.05) indicates no association. From the results, strong associations are observed between original_title and original_language, as well as release_date, with p-values of 0.0000, indicating dependency.

On the other hand, the test between original_title and media_type, as well as release_date and media_type, gives a p-value of 1.0000, showing no correlation between these variables. These findings help in feature selection and preprocessing for machine learning models by identifying relevant categorical relationships.

Conclusion-The correlation analysis using four different techniques—Pearson, Spearman, Kendall, and Chi-Square—provides valuable insights into relationships between numerical and categorical variables. Pearson correlation measures linear relationships, showing how one variable changes proportionally with another.

Spearman and Kendall correlations capture monotonic relationships, making them more robust for non-linear associations. The Chi-Square test evaluates categorical dependencies, determining whether two categorical variables are related. While Pearson is effective for continuous data with normal distribution, Spearman and Kendall are preferable for ordinal or non-linear data.

The Chi-Square test helps identify categorical variable dependencies, guiding feature selection in machine learning models. Together, these techniques provide a comprehensive understanding of data relationships, ensuring better preprocessing and model accuracy.