Luis Reynoso-Perez

CS460-01-Sum2024 | Final Project - Adapting the Apriori Algorithm

Honor Code: I have neither given nor received unauthorized aid in completing this work, nor have I presented someone else's work as my own.

This program uses the Apriori algorithm to delve into employee data, aiming to uncover patterns and associations that provide insights into why employees stay or leave. By applying the Apriori algorithm, the program identifies frequent itemsets and generates association rules, helping us understand which combinations of employee attributes are most strongly linked to turnover.

The dataset includes various attributes such as Job Satisfaction, Training Opportunities, Years of Service, Work-Life Balance, Performance Score, Commute Time, Promotion History, Department, Age, and whether the employee has left the company. To get started, the data undergoes preprocessing where categorical variables are converted into a numerical format using One-Hot Encoding, and any missing values are handled. This transformed data is then analyzed using the Apriori algorithm.

The Apriori algorithm works by iteratively identifying frequent itemsets—combinations of attributes that frequently occur together in the dataset. These itemsets are used to generate association rules, which describe the likelihood of certain outcomes, like an employee leaving, given specific conditions, such as low job satisfaction and few training opportunities. The program then filters these rules based on minimum support and confidence thresholds to ensure that only the most meaningful patterns are highlighted.

Even with strict filtering criteria, the program successfully identifies key factors influencing employee retention, such as job satisfaction and training opportunities. These insights can be invaluable for HR departments as they develop targeted strategies to improve employee satisfaction and reduce turnover. By understanding these patterns, organizations can implement more effective policies and practices, creating a supportive and engaging work environment that ultimately enhances overall employee retention.

CS460-AprioriAlgorithmApplication.py 2

Users > luisreynosoperez > CS460-AprioriAlgorithmApplication.py > ...

```python
# Importing necessary libraries
import pandas as pd  # Pandas is used for data manipulation and analysis
import numpy as np  # NumPy is used for numerical operations
from itertools import combinations  # Combinations is used to generate itemset combinations
import matplotlib.pyplot as plt  # Matplotlib is used for data visualization
import networkx as nx  # NetworkX is used for creating and visualizing network graphs
import seaborn as sns  # Seaborn is used for statistical data visualization

# Setting the random seed for reproducibility
np.random.seed(42)
n = 100  # Number of employees

# Creating a more complex dataset of employee attributes
data = {
    'JobSatisfaction': np.random.choice(['Low', 'Medium', 'High'], n, p=[0.3, 0.5, 0.2]),
    'TrainingOpportunities': np.random.choice(['Few', 'Moderate', 'Many'], n, p=[0.4, 0.4, 0.2]),
    'YearsOfService': np.random.choice(['<1', '1-2', '3-5', '6-10', '10+'], n, p=[0.1, 0.2, 0.3, 0.3, 0.1]),
    'WorkLifeBalance': np.random.choice(['Poor', 'Average', 'Good'], n, p=[0.3, 0.4, 0.3]),
    'PerformanceScore': np.random.choice(['Low', 'Medium', 'High'], n, p=[0.2, 0.6, 0.2]),
    'CommuteTime': np.random.choice(['<30min', '30-60min', '60-90min', '90+min'], n, p=[0.25, 0.35, 0.25, 0.15]),
    'PromotionHistory': np.random.choice(['Never', 'Once', 'Twice', 'Thrice+'], n, p=[0.5, 0.3, 0.15, 0.05]),
    'Department': np.random.choice(['HR', 'Engineering', 'Sales', 'Marketing', 'Finance'], n),
    'Age': np.random.randint(22, 60, n),
    'Left': np.random.choice(['Yes', 'No'], n, p=[0.3, 0.7])
}

# Creating a DataFrame from the dataset
df = pd.DataFrame(data)

# One-Hot Encoding the categorical variables
df_trans = pd.get_dummies(df)

# Printing the initial and transformed DataFrames
print("Initial DataFrame:")
print(df.head())
print("\nTransformed DataFrame (One-Hot Encoded):")
print(df_trans.head())

# Implementing function to visualize the distributions of employee data
def visualize_data(df):
    fig, axes = plt.subplots(3, 3, figsize=(15, 15))
    fig.suptitle('Employee Data Distributions', fontsize=20)

    # Plotting distributions using seaborn
    sns.histplot(df['Age'], kde=True, ax=axes[0, 0])
    sns.countplot(x='JobSatisfaction', data=df, ax=axes[0, 1])
    sns.countplot(x='TrainingOpportunities', data=df, ax=axes[0, 2])
    sns.countplot(x='YearsOfService', data=df, ax=axes[1, 0])
    sns.countplot(x='WorkLifeBalance', data=df, ax=axes[1, 1])
    sns.countplot(x='PerformanceScore', data=df, ax=axes[1, 2])
    sns.countplot(x='CommuteTime', data=df, ax=axes[2, 0])
    sns.countplot(x='PromotionHistory', data=df, ax=axes[2, 1])
    sns.countplot(x='Department', data=df, ax=axes[2, 2])
```

CS460-AprioriAlgorithmApplication.py 2 ×

Users > luisreynosoperez > CS460-AprioriAlgorithmApplication.py > ...

```python
    # Adjusting the layout
    plt.tight_layout()
    plt.subplots_adjust(top=0.95)
    plt.show()

# Visualizing the employee data distributions
visualize_data(df)

# Apriori Algorithm Implementation
def apriori(transactions, min_support):
    def get_frequent_itemsets(transactions, itemsets, min_support):
        itemset_counts = {itemset: 0 for itemset in itemsets}
        for transaction in transactions:
            for itemset in itemsets:
                if all(item in transaction for item in itemset):
                    itemset_counts[itemset] += 1
        return {itemset: count for itemset, count in itemset_counts.items() if count / len(transactions) >= min_support}

    # Converting each transaction to a frozenset of the items present in it
    transactions = transactions.apply(lambda row: frozenset(row[row == 1].index), axis=1)
    itemsets = [(col,) for col in transactions.iloc[0]]
    frequent_itemsets = {}

    # Iteratively finding frequent itemsets
    while itemsets:
        curr_frequent_itemsets = get_frequent_itemsets(transactions, itemsets, min_support)
        frequent_itemsets.update(curr_frequent_itemsets)
        itemsets = list(combinations(set().union(*[set(itemset) for itemset in curr_frequent_itemsets.keys()]), len(itemsets[0]) + 1))

    return frequent_itemsets

# Generating association rules from frequent itemsets
def generate_rules(frequent_itemsets, min_confidence):
    rules = []
    for itemset in frequent_itemsets:
        if len(itemset) > 1:
            for consequent in itemset:
                antecedent = tuple(item for item in itemset if item != consequent)
                if antecedent in frequent_itemsets:
                    confidence = frequent_itemsets[itemset] / frequent_itemsets[antecedent]
                    if confidence >= min_confidence:
                        rules.append({
                            'antecedent': antecedent,
                            'consequent': (consequent,),
                            'support': frequent_itemsets[itemset] / len(df_trans),
                            'confidence': confidence
                        })
    return rules

# Applying the Apriori algorithm to find frequent itemsets
min_support = 0.1  # Adjusted for larger dataset
frequent_itemsets = apriori(df_trans, min_support)
print("\nFrequent Itemsets:")
print(frequent_itemsets)
```

CS460-AprioriAlgorithmApplication.py 2 ×

Users > luisreynosoperez > CS460-AprioriAlgorithmApplication.py > ...

```python

# Generating association rules from the frequent itemsets
min_confidence = 0.5  # Lowered confidence threshold
rules = generate_rules(frequent_itemsets, min_confidence)
print("\nGenerated Rules:")
for rule in rules:
    print(rule)

# Filtering rules to focus on retention (Left_Yes)
filtered_retention_rules = [rule for rule in rules if ('Left_Yes',) in rule['consequent']]
print("\nFiltered Retention Rules:")
print(filtered_retention_rules)

# Converting the rules to a DataFrame for better readability
retention_rules_df = pd.DataFrame(filtered_retention_rules)
print("\nRetention Rules DataFrame:")
print(retention_rules_df)

# Implementing function to visualize association rules using a network graph
def visualize_rules(rules):
    if not rules:
        print("No rules to visualize.")
        return

    G = nx.DiGraph()
    for rule in rules:
        antecedent = ', '.join(rule['antecedent'])
        consequent = ', '.join(rule['consequent'])
        G.add_edge(antecedent, consequent, weight=rule['confidence'])

    pos = nx.spring_layout(G)
    plt.figure(figsize=(12, 12))
    nx.draw(G, pos, with_labels=True, node_color='skyblue', node_size=2500, font_size=10, font_weight='bold')
    edge_labels = {(u, v): f"{d['weight']:.2f}" for u, v, d in G.edges(data=True)}
    nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels, font_color='red')
    plt.title('Association Rules Network Graph', fontsize=20)
    plt.show()

# Visualizing the filtered retention rules
visualize_rules(filtered_retention_rules)
```

```
Initial DataFrame:
  JobSatisfaction TrainingOpportunities YearsOfService WorkLifeBalance  \
0          Medium                   Few           6-10            Poor
1            High              Moderate             <1         Average
2          Medium                   Few            1-2         Average
3          Medium              Moderate           6-10         Average
4             Low                  Many           6-10            Good

  PerformanceScore CommuteTime PromotionHistory   Department  Age Left
0              Low   60-90min            Never        Sales   57   No
1             High   30-60min            Never        Sales   47  Yes
2           Medium   30-60min            Never           HR   48   No
3             High   60-90min            Never  Engineering   26   No
4           Medium   60-90min            Never  Engineering   41  Yes
```

```
Transformed DataFrame (One-Hot Encoded):
   Age  JobSatisfaction_High  JobSatisfaction_Low  JobSatisfaction_Medium  \
0   57                 False                False                    True   
1   47                  True                False                   False   
2   48                 False                False                    True   
3   26                 False                False                    True   
4   41                 False                 True                   False   

   TrainingOpportunities_Few  TrainingOpportunities_Many  \
0                       True                       False   
1                      False                       False   
2                       True                       False   
3                      False                       False   
4                      False                        True   
```

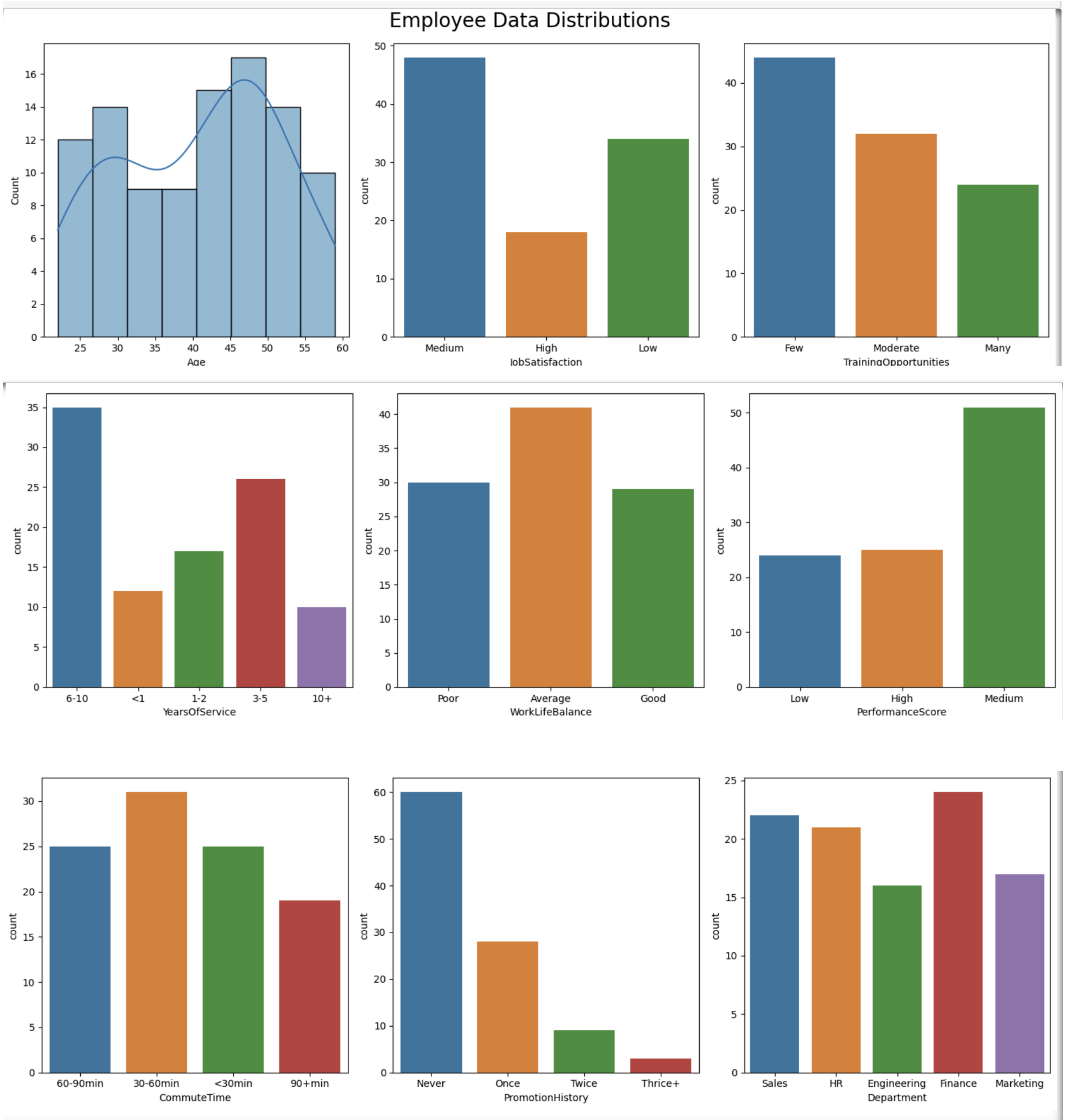
```
   TrainingOpportunities_Moderate  YearsOfService_1-2  YearsOfService_10+  \
0                           False               False               False   
1                            True               False               False   
2                           False                True               False   
3                            True               False               False   
4                           False               False               False   

   YearsOfService_3-5  ...  PromotionHistory_Once  PromotionHistory_Thrice+  \
0               False  ...                  False                    False   
1               False  ...                  False                    False   
2               False  ...                  False                    False   
3               False  ...                  False                    False   
4               False  ...                  False                    False   
```

```
   PromotionHistory_Twice  Department_Engineering  Department_Finance  \
0                   False                   False               False   
1                   False                   False               False   
2                   False                   False               False   
3                   False                    True               False   
4                   False                    True               False   

   Department_HR  Department_Marketing  Department_Sales  Left_No  Left_Yes  
0          False                 False              True     True     False  
1          False                 False              True    False      True  
2           True                 False             False     True     False  
3          False                 False             False     True     False  
4          False                 False             False    False      True  

[5 rows x 33 columns]
```


Employee Data Distributions
Count
Age
count
Medium
High
Low
JobSatisfaction
count
Few
Moderate
Many
TrainingOpportunities
count
6-10
<1
1-2
3-5
10+
YearsOfService
count
Poor
Average
Good
WorkLifeBalance
count
Low
High
Medium
PerformanceScore
count
60-90min
30-60min
<30min
90+min
CommuteTime
count
Never
Once
Twice
Thrice+
PromotionHistory
count
Sales
HR
Engineering
Finance
Marketing
Department

Frequent Itemsets:
{('YearsOfService_6-10',): 35, ('TrainingOpportunities_Few',): 44, ('JobSatisfaction_Medium',): 48, ('Department_Sa
les',): 22, ('Left_No',): 58, ('PerformanceScore_Low',): 24, ('WorkLifeBalance_Poor',): 30, ('PromotionHistory_Neve
r',): 60, ('CommuteTime_60-90min',): 25, ('YearsOfService_6-10', 'TrainingOpportunities_Few'): 16, ('YearsOfService
_6-10', 'JobSatisfaction_Medium'): 15, ('YearsOfService_6-10', 'Department_Sales'): 10, ('YearsOfService_6-10', 'Le
ft_No'): 18, ('YearsOfService_6-10', 'WorkLifeBalance_Poor'): 14, ('YearsOfService_6-10', 'PromotionHistory_Neve
r'): 23, ('YearsOfService_6-10', 'CommuteTime_60-90min'): 10, ('TrainingOpportunities_Few', 'JobSatisfaction_Mediu
m'): 20, ('TrainingOpportunities_Few', 'Left_No'): 25, ('TrainingOpportunities_Few', 'WorkLifeBalance_Poor'): 13,
('TrainingOpportunities_Few', 'PromotionHistory_Never'): 25, ('TrainingOpportunities_Few', 'CommuteTime_60-90min'):
15, ('JobSatisfaction_Medium', 'Department_Sales'): 13, ('JobSatisfaction_Medium', 'Left_No'): 26, ('JobSatisfactio
n_Medium', 'PerformanceScore_Low'): 14, ('JobSatisfaction_Medium', 'WorkLifeBalance_Poor'): 12, ('JobSatisfaction_M
edium', 'PromotionHistory_Never'): 27, ('JobSatisfaction_Medium', 'CommuteTime_60-90min'): 15, ('Department_Sales',
'Left_No'): 11, ('Department_Sales', 'PromotionHistory_Never'): 16, ('Left_No', 'PerformanceScore_Low'): 14, ('Left
_No', 'WorkLifeBalance_Poor'): 18, ('Left_No', 'PromotionHistory_Never'): 33, ('Left_No', 'CommuteTime_60-90min'):
16, ('PerformanceScore_Low', 'WorkLifeBalance_Poor'): 11, ('PerformanceScore_Low', 'PromotionHistory_Never'): 15,
('PerformanceScore_Low', 'CommuteTime_60-90min'): 11, ('WorkLifeBalance_Poor', 'PromotionHistory_Never'): 17, ('Pro
motionHistory_Never', 'CommuteTime_60-90min'): 15, ('YearsOfService_6-10', 'TrainingOpportunities_Few', 'PromotionH
istory_Never'): 11, ('YearsOfService_6-10', 'JobSatisfaction_Medium', 'PromotionHistory_Never'): 11, ('YearsOfServi
ce_6-10', 'Left_No', 'PromotionHistory_Never'): 11, ('TrainingOpportunities_Few', 'JobSatisfaction_Medium', 'Left_N

m'): 20, ('TrainingOpportunities_Few', 'Left_No'): 25, ('TrainingOpportunities_Few', 'WorkLifeBalance_Poor'): 13,
('TrainingOpportunities_Few', 'PromotionHistory_Never'): 25, ('TrainingOpportunities_Few', 'CommuteTime_60-90min'):
15, ('JobSatisfaction_Medium', 'Department_Sales'): 13, ('JobSatisfaction_Medium', 'Left_No'): 26, ('JobSatisfactio
n_Medium', 'PerformanceScore_Low'): 14, ('JobSatisfaction_Medium', 'WorkLifeBalance_Poor'): 12, ('JobSatisfaction_M
edium', 'PromotionHistory_Never'): 27, ('JobSatisfaction_Medium', 'CommuteTime_60-90min'): 15, ('Department_Sales',
'Left_No'): 11, ('Department_Sales', 'PromotionHistory_Never'): 16, ('Left_No', 'PerformanceScore_Low'): 14, ('Left
_No', 'WorkLifeBalance_Poor'): 18, ('Left_No', 'PromotionHistory_Never'): 33, ('Left_No', 'CommuteTime_60-90min'):
16, ('PerformanceScore_Low', 'WorkLifeBalance_Poor'): 11, ('PerformanceScore_Low', 'PromotionHistory_Never'): 15,
('PerformanceScore_Low', 'CommuteTime_60-90min'): 11, ('WorkLifeBalance_Poor', 'PromotionHistory_Never'): 17, ('Pro
motionHistory_Never', 'CommuteTime_60-90min'): 15, ('YearsOfService_6-10', 'TrainingOpportunities_Few', 'PromotionH
istory_Never'): 11, ('YearsOfService_6-10', 'JobSatisfaction_Medium', 'PromotionHistory_Never'): 11, ('YearsOfServi
ce_6-10', 'Left_No', 'PromotionHistory_Never'): 11, ('TrainingOpportunities_Few', 'JobSatisfaction_Medium', 'Left_N
o'): 11, ('TrainingOpportunities_Few', 'JobSatisfaction_Medium', 'PromotionHistory_Never'): 11, ('TrainingOpportuni
ties_Few', 'Left_No', 'PromotionHistory_Never'): 11, ('JobSatisfaction_Medium', 'Left_No', 'PromotionHistory_Neve
r'): 14, ('Left_No', 'WorkLifeBalance_Poor', 'PromotionHistory_Never'): 10}

Generated Rules:
{'antecedent': ('YearsOfService_6-10',), 'consequent': ('Left_No',), 'support': 0.18, 'confidence': 0.5142857142857
142}
{'antecedent': ('YearsOfService_6-10',), 'consequent': ('PromotionHistory_Never',), 'support': 0.23, 'confidence':
0.6571428571428571}

Generated Rules:
{'antecedent': ('YearsOfService_6-10',), 'consequent': ('Left_No',), 'support': 0.18, 'confidence': 0.5142857142857
142}
{'antecedent': ('YearsOfService_6-10',), 'consequent': ('PromotionHistory_Never',), 'support': 0.23, 'confidence':
0.6571428571428571}
{'antecedent': ('TrainingOpportunities_Few',), 'consequent': ('Left_No',), 'support': 0.25, 'confidence': 0.5681818
181818182}
{'antecedent': ('TrainingOpportunities_Few',), 'consequent': ('PromotionHistory_Never',), 'support': 0.25, 'confide
nce': 0.5681818181818182}
{'antecedent': ('CommuteTime_60-90min',), 'consequent': ('TrainingOpportunities_Few',), 'support': 0.15, 'confidenc
e': 0.6}
{'antecedent': ('Department_Sales',), 'consequent': ('JobSatisfaction_Medium',), 'support': 0.13, 'confidence': 0.5
909090909090909}
{'antecedent': ('JobSatisfaction_Medium',), 'consequent': ('Left_No',), 'support': 0.26, 'confidence': 0.5416666666
666666}
{'antecedent': ('PerformanceScore_Low',), 'consequent': ('JobSatisfaction_Medium',), 'support': 0.14, 'confidence':
0.5833333333333334}
{'antecedent': ('JobSatisfaction_Medium',), 'consequent': ('PromotionHistory_Never',), 'support': 0.27, 'confidenc
e': 0.5625}
{'antecedent': ('CommuteTime_60-90min',), 'consequent': ('JobSatisfaction_Medium',), 'support': 0.15, 'confidence':

```
{'antecedent': ('PerformanceScore_Low',), 'consequent': ('JobSatisfaction_Medium',), 'support': 0.14, 'confidence': 0.5833333333333334}
{'antecedent': ('JobSatisfaction_Medium',), 'consequent': ('PromotionHistory_Never',), 'support': 0.27, 'confidence': 0.5625}
{'antecedent': ('CommuteTime_60-90min',), 'consequent': ('JobSatisfaction_Medium',), 'support': 0.15, 'confidence': 0.6}
{'antecedent': ('Department_Sales',), 'consequent': ('Left_No',), 'support': 0.11, 'confidence': 0.5}
{'antecedent': ('Department_Sales',), 'consequent': ('PromotionHistory_Never',), 'support': 0.16, 'confidence': 0.7272727272727273}
{'antecedent': ('PerformanceScore_Low',), 'consequent': ('Left_No',), 'support': 0.14, 'confidence': 0.5833333333333334}
{'antecedent': ('WorkLifeBalance_Poor',), 'consequent': ('Left_No',), 'support': 0.18, 'confidence': 0.6}
{'antecedent': ('PromotionHistory_Never',), 'consequent': ('Left_No',), 'support': 0.33, 'confidence': 0.55}
{'antecedent': ('Left_No',), 'consequent': ('PromotionHistory_Never',), 'support': 0.33, 'confidence': 0.5689655172413793}
{'antecedent': ('CommuteTime_60-90min',), 'consequent': ('Left_No',), 'support': 0.16, 'confidence': 0.64}
{'antecedent': ('PerformanceScore_Low',), 'consequent': ('PromotionHistory_Never',), 'support': 0.15, 'confidence': 0.625}
{'antecedent': ('WorkLifeBalance_Poor',), 'consequent': ('PromotionHistory_Never',), 'support': 0.17, 'confidence': 0.5666666666666667}
```

```
{'antecedent': ('CommuteTime_60-90min',), 'consequent': ('PromotionHistory_Never',), 'support': 0.15, 'confidence': 0.6}
{'antecedent': ('YearsOfService_6-10', 'TrainingOpportunities_Few'), 'consequent': ('PromotionHistory_Never',), 'support': 0.11, 'confidence': 0.6875}
{'antecedent': ('YearsOfService_6-10', 'JobSatisfaction_Medium'), 'consequent': ('PromotionHistory_Never',), 'support': 0.11, 'confidence': 0.7333333333333333}
{'antecedent': ('YearsOfService_6-10', 'Left_No'), 'consequent': ('PromotionHistory_Never',), 'support': 0.11, 'confidence': 0.6111111111111112}
{'antecedent': ('TrainingOpportunities_Few', 'JobSatisfaction_Medium'), 'consequent': ('Left_No',), 'support': 0.11, 'confidence': 0.55}
{'antecedent': ('TrainingOpportunities_Few', 'JobSatisfaction_Medium'), 'consequent': ('PromotionHistory_Never',), 'support': 0.11, 'confidence': 0.55}
{'antecedent': ('JobSatisfaction_Medium', 'PromotionHistory_Never'), 'consequent': ('Left_No',), 'support': 0.14, 'confidence': 0.5185185185185185}
{'antecedent': ('JobSatisfaction_Medium', 'Left_No'), 'consequent': ('PromotionHistory_Never',), 'support': 0.14, 'confidence': 0.5384615384615384}
{'antecedent': ('WorkLifeBalance_Poor', 'PromotionHistory_Never'), 'consequent': ('Left_No',), 'support': 0.1, 'confidence': 0.5882352941176471}
{'antecedent': ('Left_No', 'WorkLifeBalance_Poor'), 'consequent': ('PromotionHistory_Never',), 'support': 0.1, 'confidence': 0.5555555555555556}
```

:


```
'confidence': 0.5185185185185185}
{'antecedent': ('JobSatisfaction_Medium', 'Left_No'), 'consequent': ('PromotionHistory_Never',), 'support': 0.14, 'confidence': 0.5384615384615384}
{'antecedent': ('WorkLifeBalance_Poor', 'PromotionHistory_Never'), 'consequent': ('Left_No',), 'support': 0.1, 'confidence': 0.5882352941176471}
{'antecedent': ('Left_No', 'WorkLifeBalance_Poor'), 'consequent': ('PromotionHistory_Never',), 'support': 0.1, 'confidence': 0.5555555555555556}
```


```
Filtered Retention Rules:
[]

Retention Rules DataFrame:
Empty DataFrame
Columns: []
Index: []
No rules to visualize.
```