DATA1030Project

December 3, 2024

[2]:
```
import kagglehub

# Download latest version
path = kagglehub.dataset_download("utkarshx27/
 ↪which-resume-attributes-drive-job-callbacks")

print("Path to dataset files:", path)
```

```
Warning: Looks like you're using an outdated `kagglehub` version, please
consider updating (latest version: 0.3.4)
Path to dataset files: /Users/fruit/.cache/kagglehub/datasets/utkarshx27/which-
resume-attributes-drive-job-callbacks/versions/1
```

[3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv("resume.csv")
```

[4]:
```
#shape of df
print(df.shape[0], 'rows')
print(df.shape[1], 'columns')
print()

#head
print(df.head())
```

```
4870 rows
30 columns

   job_ad_id job_city   job_industry    job_type  job_fed_contractor  \
0        384  Chicago  manufacturing  supervisor                 NaN
1        384  Chicago  manufacturing  supervisor                 NaN
2        384  Chicago  manufacturing  supervisor                 NaN
3        384  Chicago  manufacturing  supervisor                 NaN
4        385  Chicago  other_service   secretary                 0.0

   job_equal_opp_employer job_ownership  job_req_any  job_req_communication  \
```

```
0                      1       unknown            1                      0
1                      1       unknown            1                      0
2                      1       unknown            1                      0
3                      1       unknown            1                      0
4                      1     nonprofit            1                      0

   job_req_education  … honors  worked_during_school  years_experience  \
0                  0  …      0                     0                 6
1                  0  …      0                     1                 6
2                  0  …      0                     1                 6
3                  0  …      0                     0                 6
4                  0  …      0                     1                22

  computer_skills  special_skills volunteer military employment_holes  \
0               1               0         0        0                1
1               1               0         1        1                0
2               1               0         0        0                0
3               1               1         1        0                1
4               1               0         0        0                0

   has_email_address  resume_quality
0                  0            low
1                  1           high
2                  0            low
3                  1           high
4                  1           high

[5 rows x 30 columns]
```

[5]:
```
#column type
print(df.dtypes)
print()
```

```
job_ad_id                  int64
job_city                  object
job_industry              object
job_type                  object
job_fed_contractor       float64
job_equal_opp_employer     int64
job_ownership             object
job_req_any                int64
job_req_communication      int64
job_req_education          int64
job_req_min_experience    object
job_req_computer           int64
job_req_organization       int64
job_req_school            object
received_callback          int64
```

```
firstname                  object
race                       object
gender                     object
years_college               int64
college_degree              int64
honors                      int64
worked_during_school        int64
years_experience            int64
computer_skills             int64
special_skills              int64
volunteer                   int64
military                    int64
employment_holes            int64
has_email_address           int64
resume_quality             object
dtype: object
```

[6]:
```
#missing values
placeholders = ['unknown', 'UNK', 'Unknown']
missing_values = df.isnull() | df.isin(placeholders)
missing_count = missing_values.sum()
print(missing_count)
```

```
job_ad_id                     0
job_city                      0
job_industry                  0
job_type                      0
job_fed_contractor         1768
job_equal_opp_employer        0
job_ownership              1992
job_req_any                   0
job_req_communication         0
job_req_education             0
job_req_min_experience     2746
job_req_computer              0
job_req_organization          0
job_req_school                0
received_callback             0
firstname                     0
race                          0
gender                        0
years_college                 0
college_degree                0
honors                        0
worked_during_school          0
years_experience              0
computer_skills               0
```

```
special_skills               0
volunteer                    0
military                     0
employment_holes             0
has_email_address            0
resume_quality               0
dtype: int64
```

[7]:
```
#check target variable
print(df['received_callback'].describe)
print('Target variable Music Effects is categorical')
print()
```

```
<bound method NDFrame.describe of 0       0
1       0
2       0
3       0
4       0
       ..
4865    0
4866    0
4867    0
4868    0
4869    0
Name: received_callback, Length: 4870, dtype: int64>
Target variable Music Effects is categorical
```

[8]:
```
df.describe()
```

[8]:

	job_ad_id	job_fed_contractor	job_equal_opp_employer	job_req_any	\
count	4870.000000	3102.000000	4870.000000	4870.000000	
mean	651.777823	0.114765	0.291170	0.787269	
std	388.690698	0.318789	0.454349	0.409281	
min	1.000000	0.000000	0.000000	0.000000	
25%	306.250000	0.000000	0.000000	1.000000	
50%	647.000000	0.000000	0.000000	1.000000	
75%	979.750000	0.000000	1.000000	1.000000	
max	1344.000000	1.000000	1.000000	1.000000	

	job_req_communication	job_req_education	job_req_computer	\
count	4870.000000	4870.000000	4870.000000	
mean	0.124846	0.106776	0.437166	
std	0.330578	0.308860	0.496087	
min	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	
75%	0.000000	0.000000	1.000000	

```
max                 1.000000           1.000000          1.000000

       job_req_organization  received_callback  years_college  college_degree  \
count           4870.000000        4870.000000    4870.000000     4870.000000
mean               0.072690           0.080493       3.618480        0.719507
std                0.259654           0.272083       0.714997        0.449286
min                0.000000           0.000000       0.000000        0.000000
25%                0.000000           0.000000       3.000000        0.000000
50%                0.000000           0.000000       4.000000        1.000000
75%                0.000000           0.000000       4.000000        1.000000
max                1.000000           1.000000       4.000000        1.000000

            honors  worked_during_school  years_experience  computer_skills  \
count  4870.000000           4870.000000       4870.000000      4870.000000
mean      0.052772              0.559548          7.842916         0.820534
std       0.223601              0.496492          5.044612         0.383782
min       0.000000              0.000000          1.000000         0.000000
25%       0.000000              0.000000          5.000000         1.000000
50%       0.000000              1.000000          6.000000         1.000000
75%       0.000000              1.000000          9.000000         1.000000
max       1.000000              1.000000         44.000000         1.000000

       special_skills    volunteer     military  employment_holes  \
count     4870.000000  4870.000000  4870.000000       4870.000000
mean         0.328747     0.411499     0.097125          0.448049
std          0.469806     0.492156     0.296159          0.497345
min          0.000000     0.000000     0.000000          0.000000
25%          0.000000     0.000000     0.000000          0.000000
50%          0.000000     0.000000     0.000000          0.000000
75%          1.000000     1.000000     0.000000          1.000000
max          1.000000     1.000000     1.000000          1.000000

       has_email_address
count        4870.000000
mean            0.479261
std             0.499621
min             0.000000
25%             0.000000
50%             0.000000
75%             1.000000
max             1.000000
```

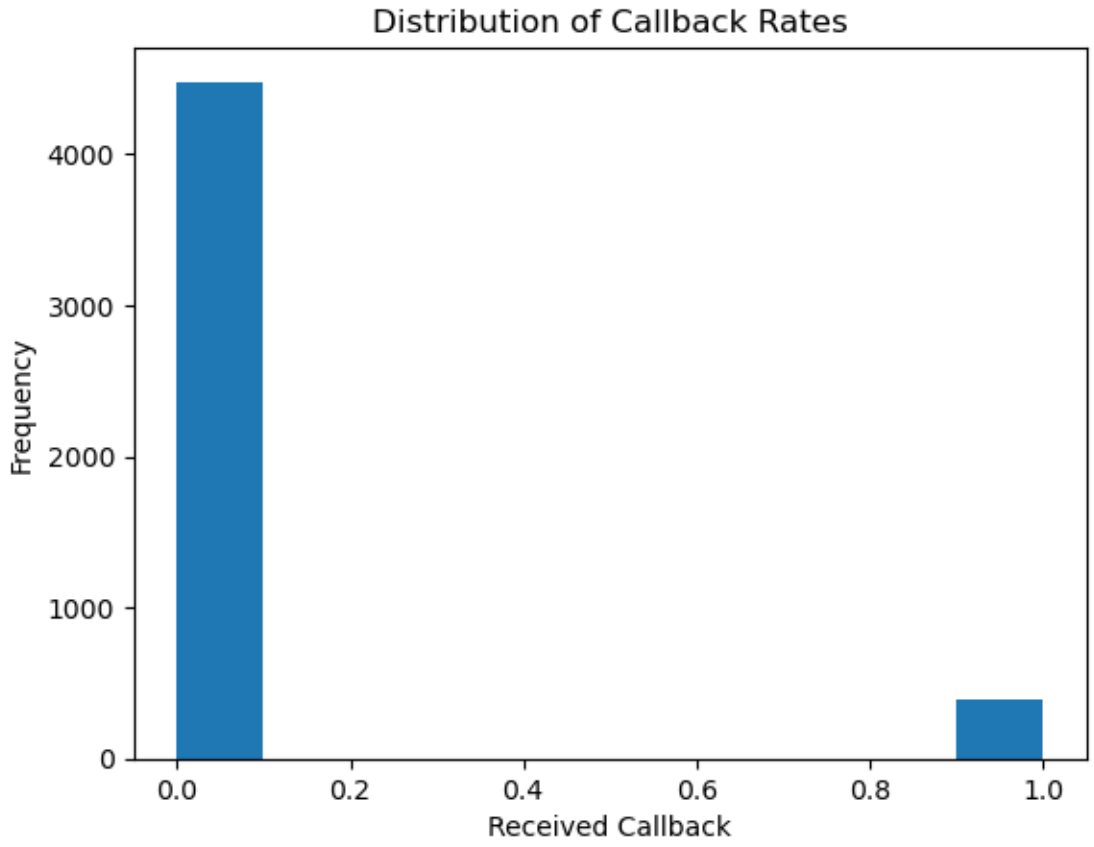
[9]:
```
df.columns
```

[9]:
```
Index(['job_ad_id', 'job_city', 'job_industry', 'job_type',
       'job_fed_contractor', 'job_equal_opp_employer', 'job_ownership',
       'job_req_any', 'job_req_communication', 'job_req_education',
```

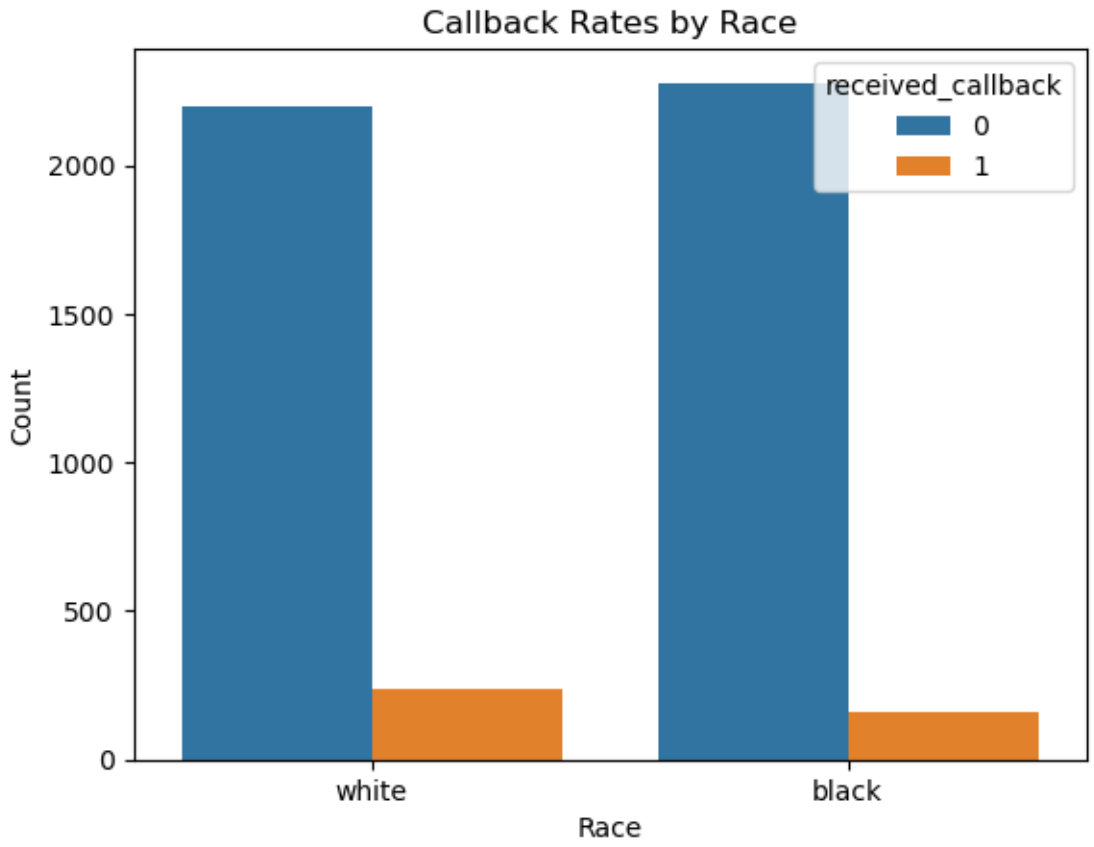
```
       'job_req_min_experience', 'job_req_computer', 'job_req_organization',
       'job_req_school', 'received_callback', 'firstname', 'race', 'gender',
       'years_college', 'college_degree', 'honors', 'worked_during_school',
       'years_experience', 'computer_skills', 'special_skills', 'volunteer',
       'military', 'employment_holes', 'has_email_address', 'resume_quality'],
      dtype='object')
```

[10]:
```
#sns.pairplot(df, hue = "received_callback")
#plt.show()
```

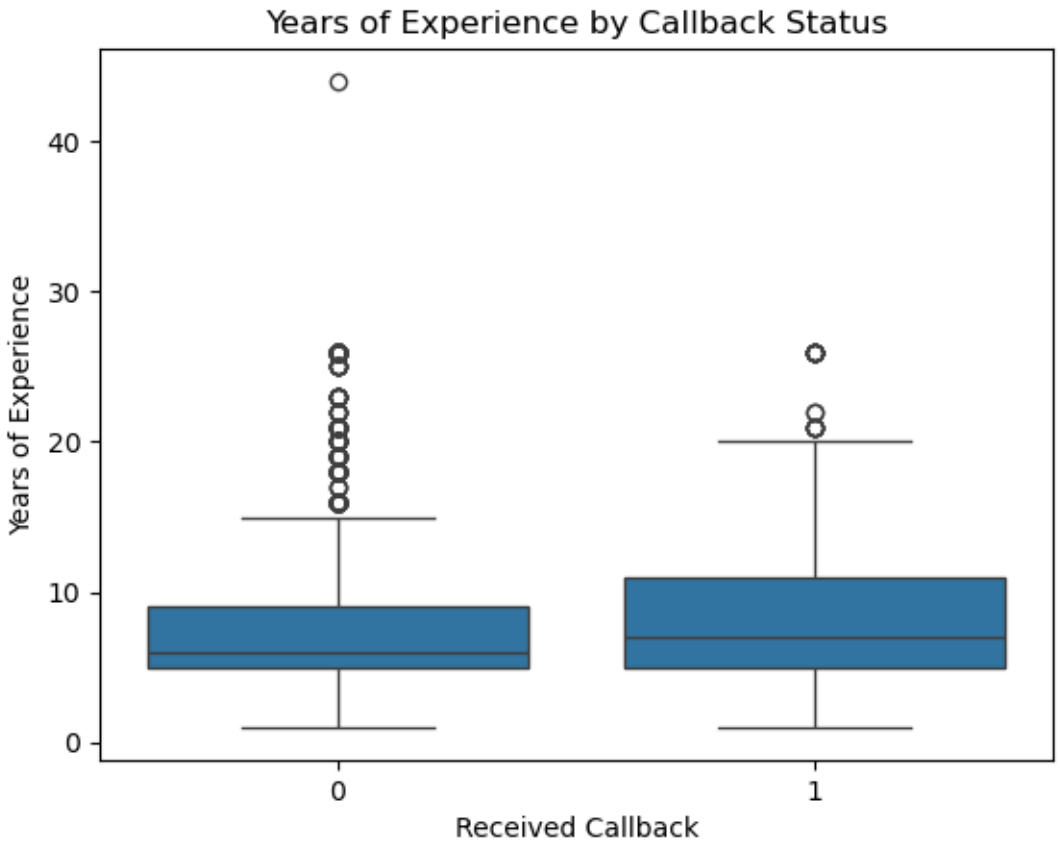
[11]:
```
# Plot a histogram of callback rates

plt.hist(df["received_callback"])
plt.xlabel("Received Callback")
plt.ylabel("Frequency")
plt.title("Distribution of Callback Rates")
plt.show()
```



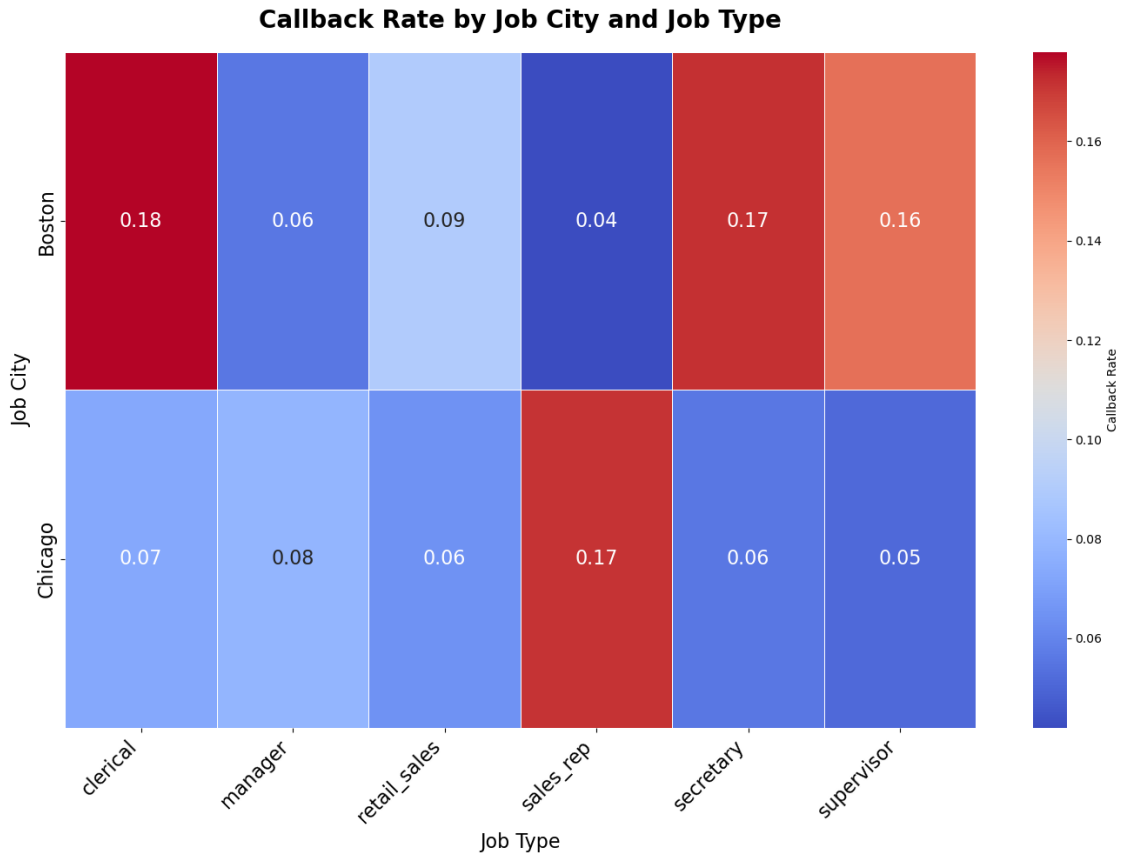
[12]:
```
sns.countplot(x="race", hue="received_callback", data=df)
plt.xlabel("Race")
plt.ylabel("Count")
plt.title("Callback Rates by Race")
plt.show()
```



[13]:
```
sns.boxplot(x="received_callback", y="years_experience", data=df)
plt.xlabel("Received Callback")
plt.ylabel("Years of Experience")
plt.title("Years of Experience by Callback Status")
plt.show()
```

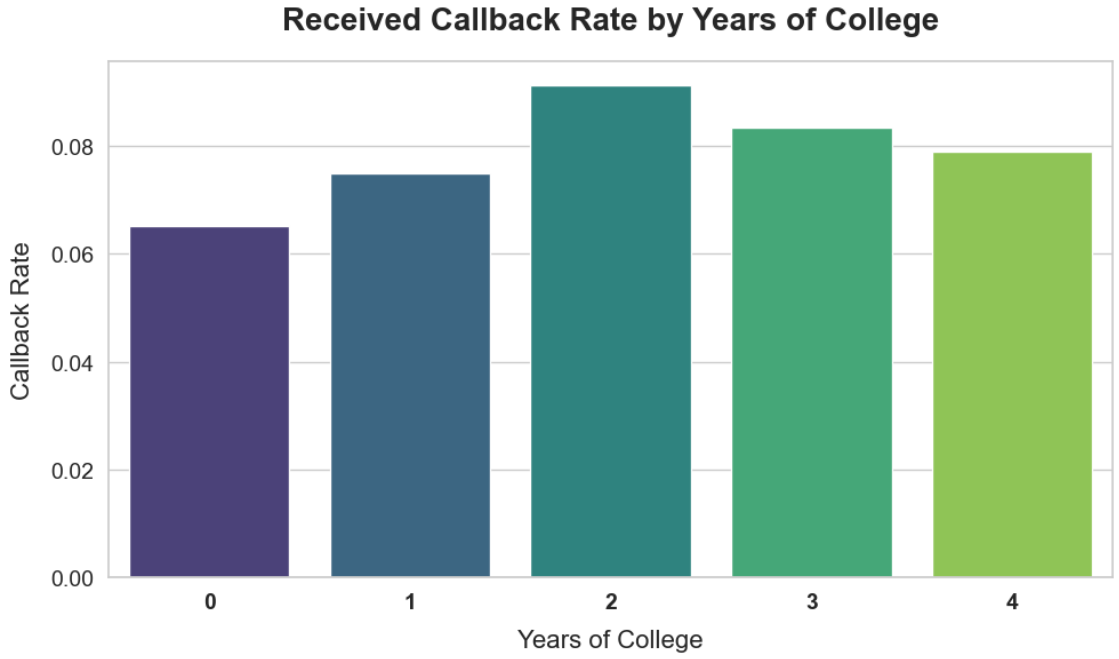


[14]:
```
heatmap_data = df.pivot_table(index='job_city', columns='job_type',␣
 ↪values='received_callback', aggfunc='mean')

# Plotting the heatmap with an improved design
plt.figure(figsize=(14, 10))
sns.heatmap(heatmap_data, cmap='coolwarm', annot=True, fmt=".2f", linewidths=0.
 ↪5, cbar_kws={'label': 'Callback Rate'}, annot_kws={"size": 16})
plt.title('Callback Rate by Job City and Job Type', fontsize=20,␣
 ↪fontweight='bold', pad=20)
plt.xticks(rotation=45, ha='right', fontsize=16)
plt.yticks(fontsize=16)
plt.xlabel("Job Type", fontsize=16)
plt.ylabel("Job City", fontsize=16)
plt.tight_layout()
plt.show()
```



[15]:
```
# Barplot for Received Callback Rate by Years of College
plt.figure(figsize=(10, 6))
sns.set(style="whitegrid")

# Create the barplot
sns.barplot(x='years_college', y='received_callback', data=df, ci=None,␣
 ↪palette='viridis')

# Customizing the plot
plt.title('Received Callback Rate by Years of College', fontsize=20,␣
 ↪fontweight='bold', pad=20)
plt.xlabel('Years of College', fontsize=16, labelpad=10)
plt.ylabel('Callback Rate', fontsize=16, labelpad=10)
plt.xticks(fontsize=14, weight='bold')
plt.yticks(fontsize=14)

# Display the plot
plt.tight_layout()
plt.show()
```

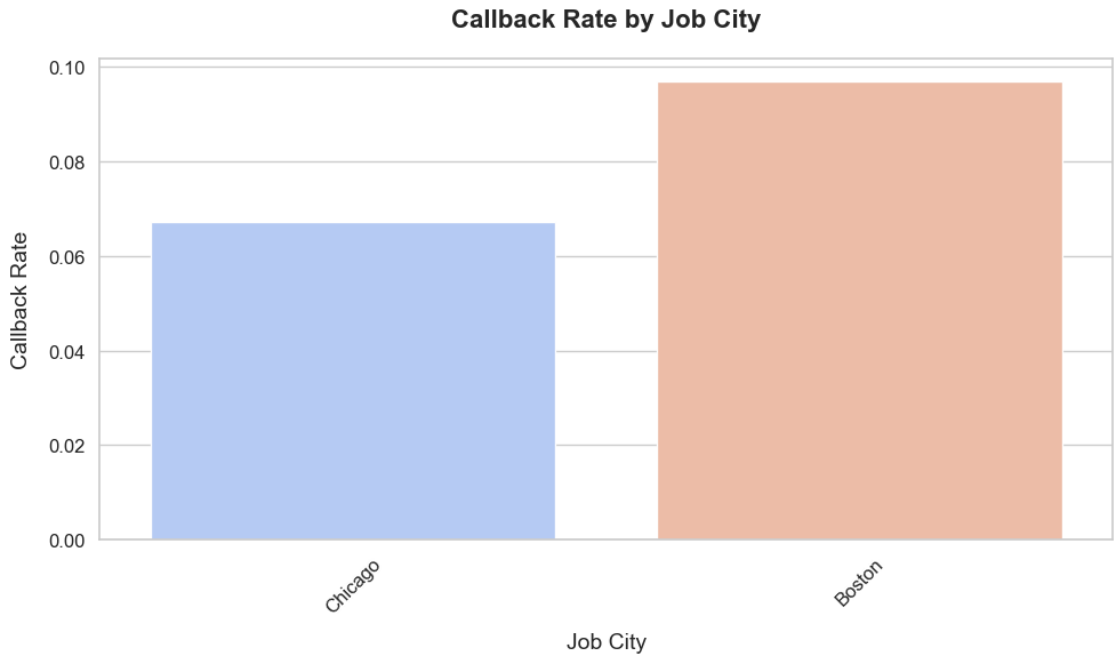
```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/3909797406.py:6
: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='years_college', y='received_callback', data=df, ci=None,
palette='viridis')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/3909797406.py:6
: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x='years_college', y='received_callback', data=df, ci=None,
palette='viridis')
```



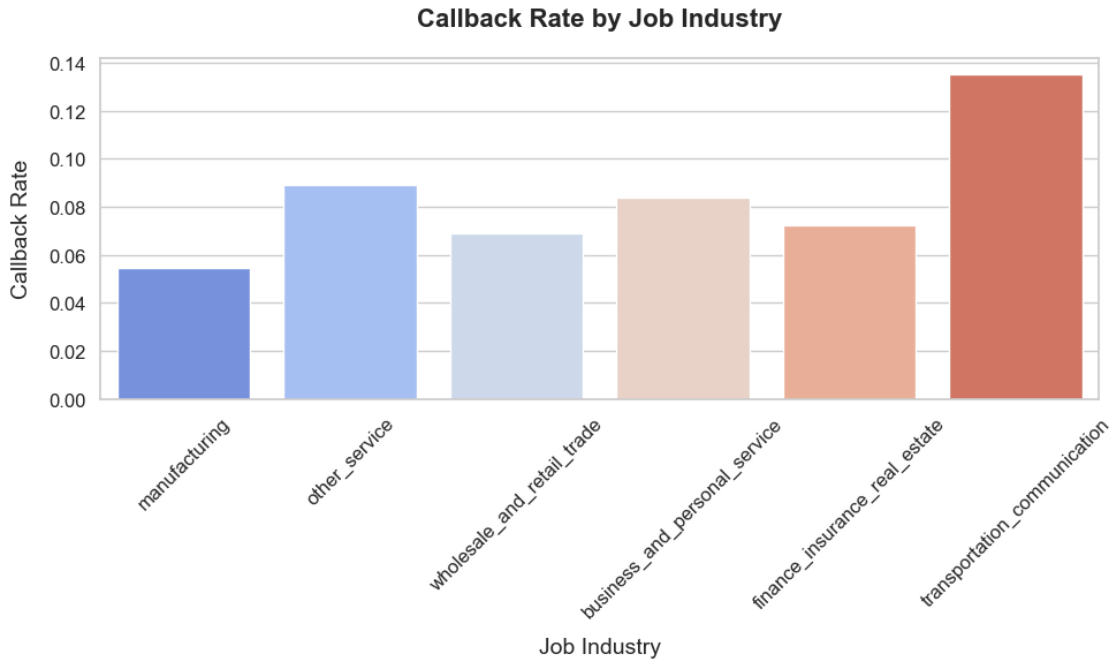
[16]:
```
# List of categorical features (excluding the target 'received_callback')
categorical_features = ['job_city', 'job_industry', 'job_type',␣
 ↪'job_fed_contractor', 'job_equal_opp_employer',
                         'job_ownership', 'job_req_any',␣
 ↪'job_req_communication', 'job_req_education', 'job_req_computer',
                         'job_req_organization', 'race', 'gender',␣
 ↪'college_degree', 'honors', 'worked_during_school',
```

```
                        'computer_skills', 'special_skills', 'volunteer',␣
 ↪'military', 'employment_holes', 'has_email_address']

# Loop through each categorical feature and create a bar plot for callback rate
for feature in categorical_features:
    plt.figure(figsize=(10, 6))
    sns.set(style="whitegrid")

    # Create a bar plot for each categorical feature vs callback rate
    sns.barplot(x=feature, y='received_callback', data=df, ci=None,␣
 ↪palette='coolwarm')

    # Customize the plot
    plt.title(f'Callback Rate by {feature.replace("_", " ").title()}',␣
 ↪fontsize=16, fontweight='bold', pad=20)
    plt.xlabel(f'{feature.replace("_", " ").title()}', fontsize=14, labelpad=10)
    plt.ylabel('Callback Rate', fontsize=14, labelpad=10)
    plt.xticks(rotation=45, fontsize=12)  # Rotate x-axis labels for better␣
 ↪readability
    plt.yticks(fontsize=12)

    # Display the plot
    plt.tight_layout()
    plt.show()
```

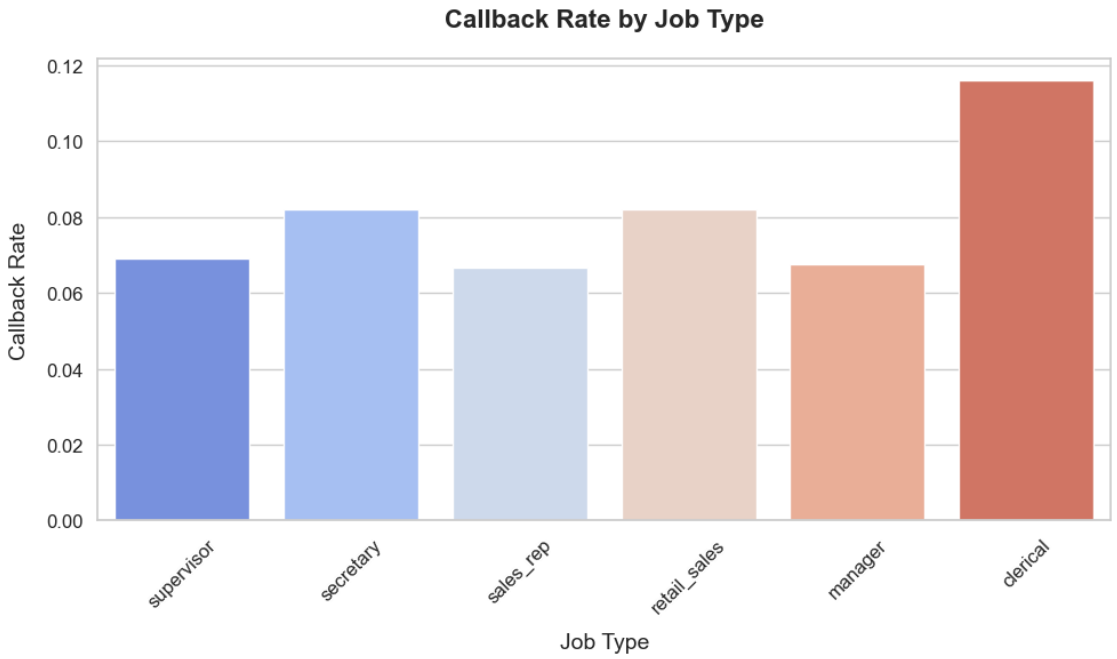
```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



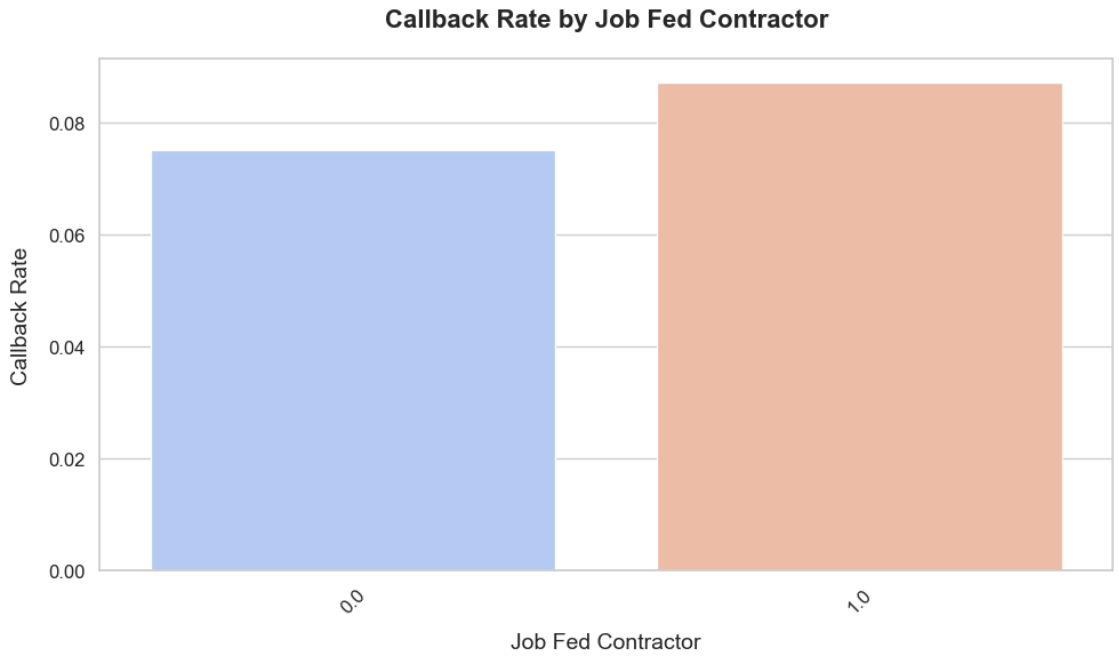

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



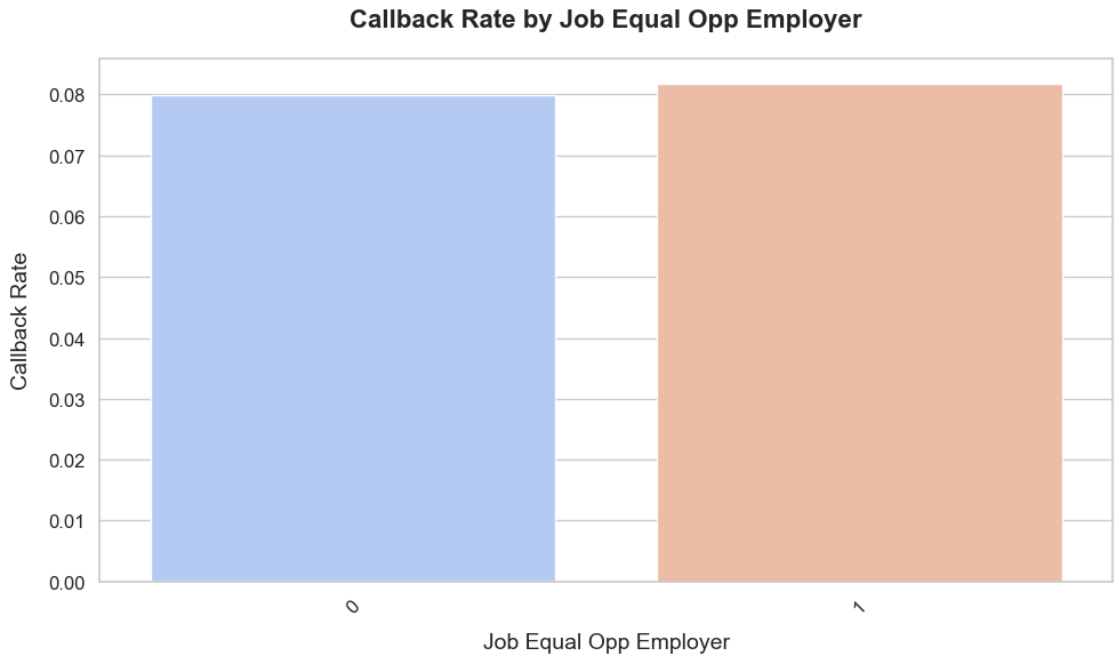

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



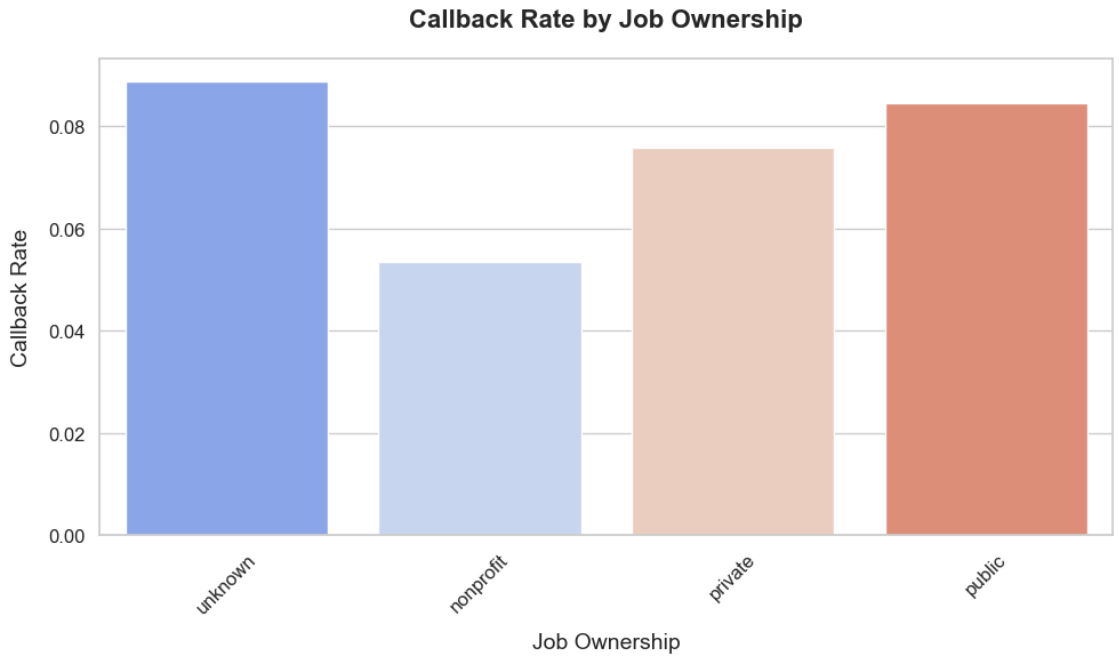

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



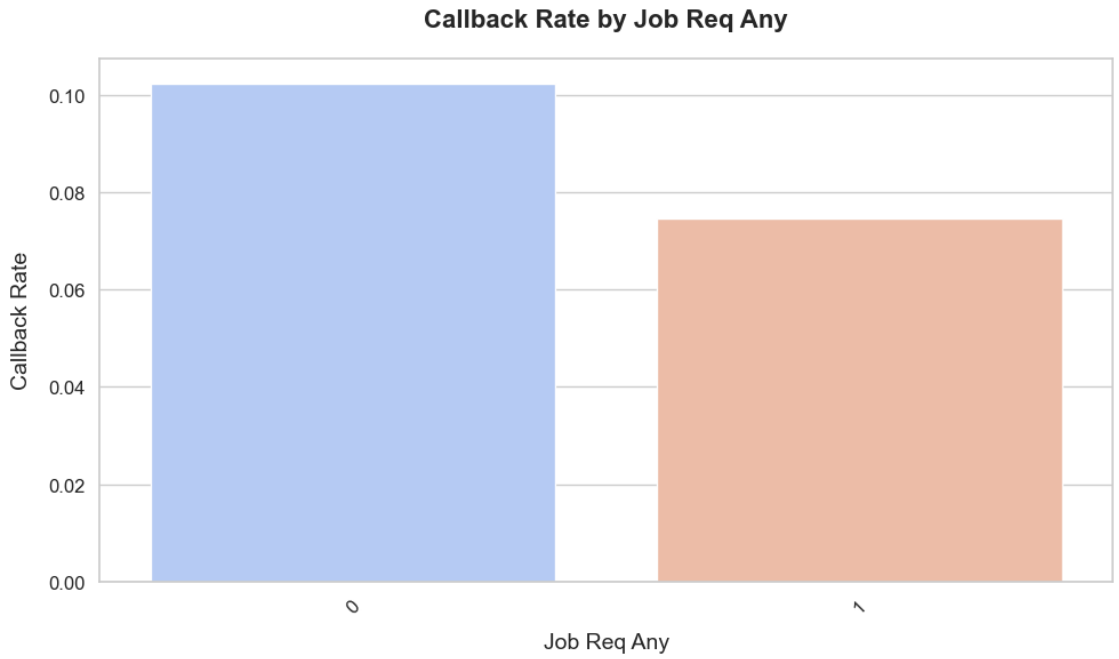

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



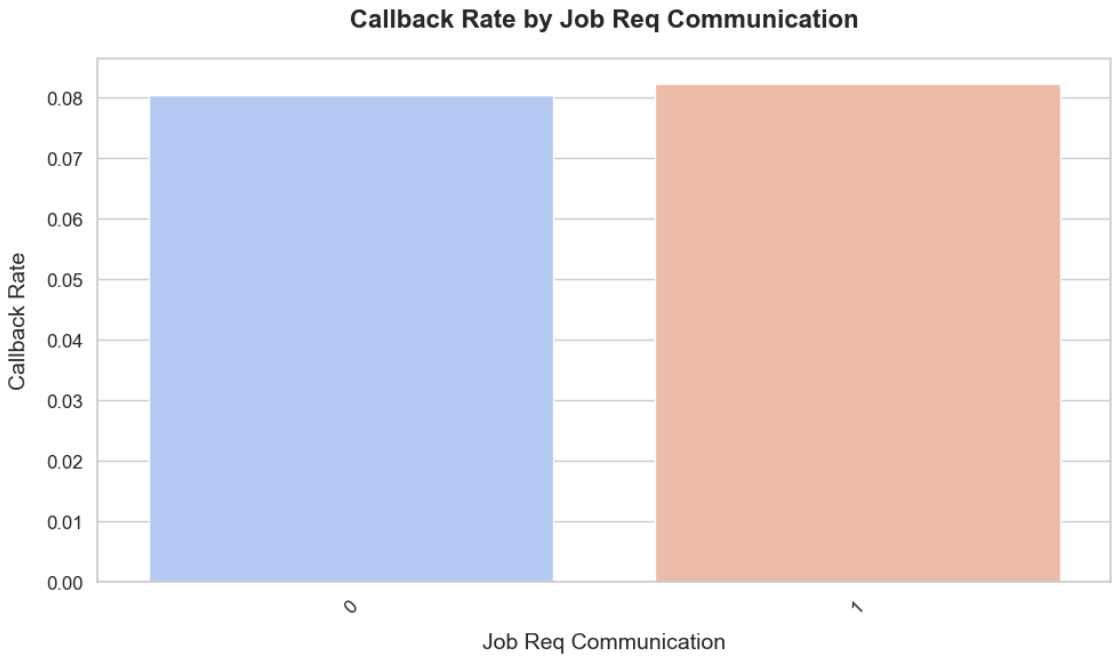

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



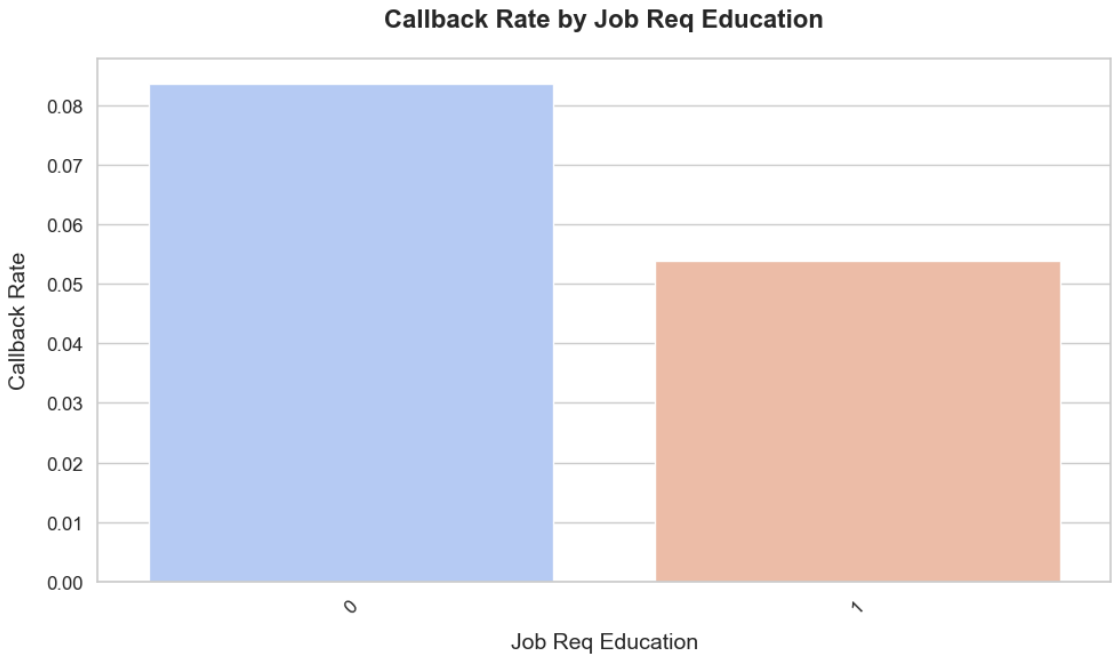

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



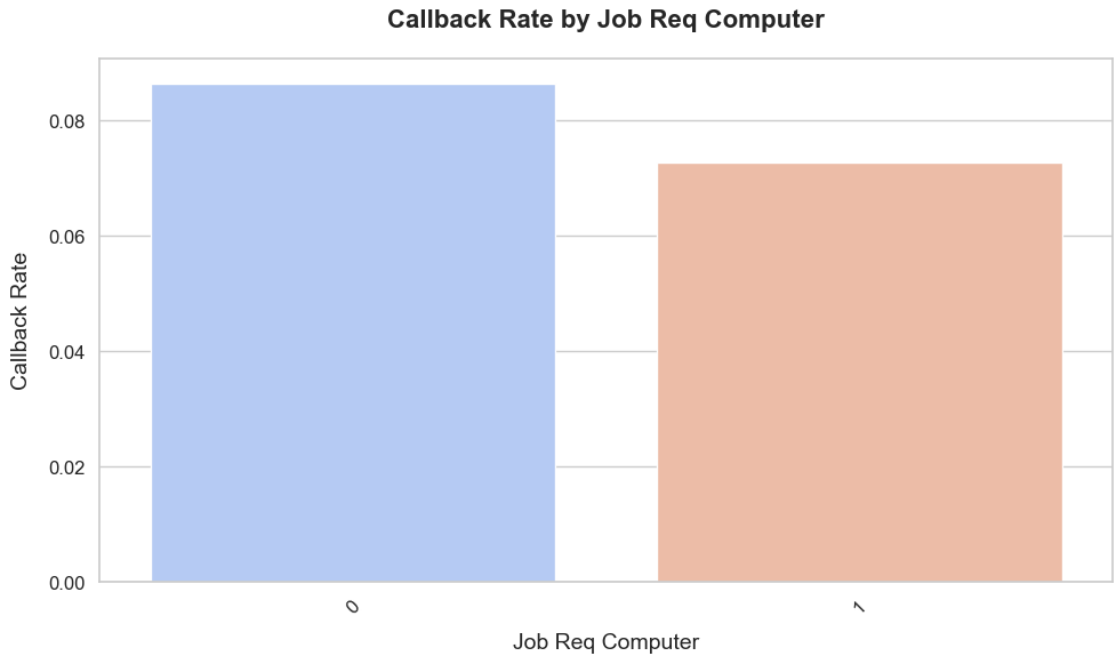

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



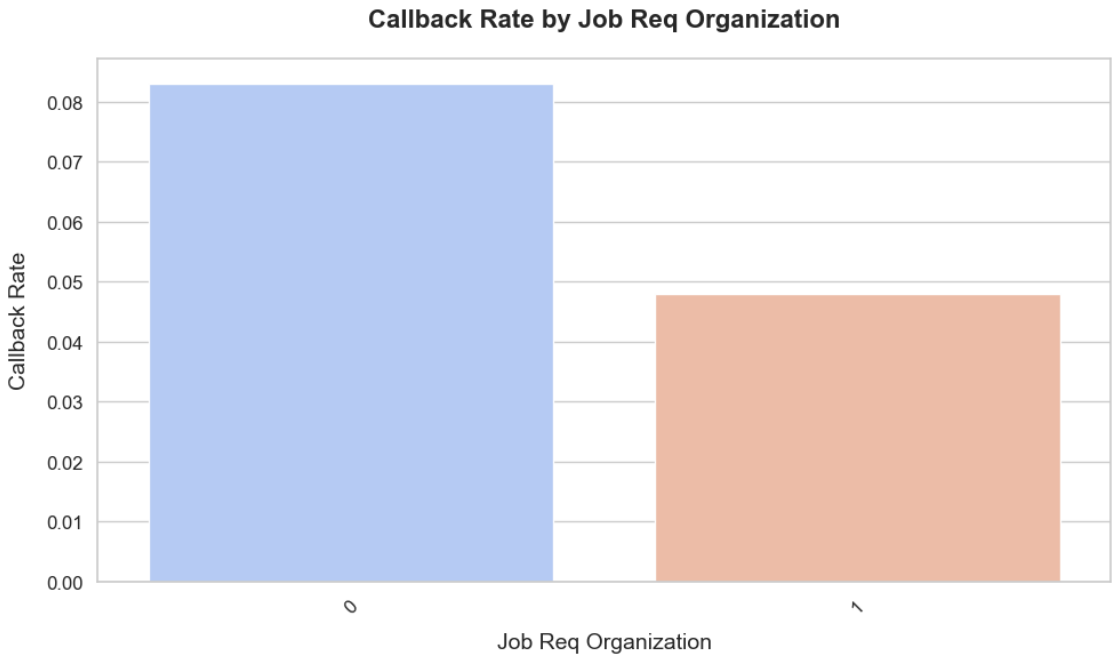

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



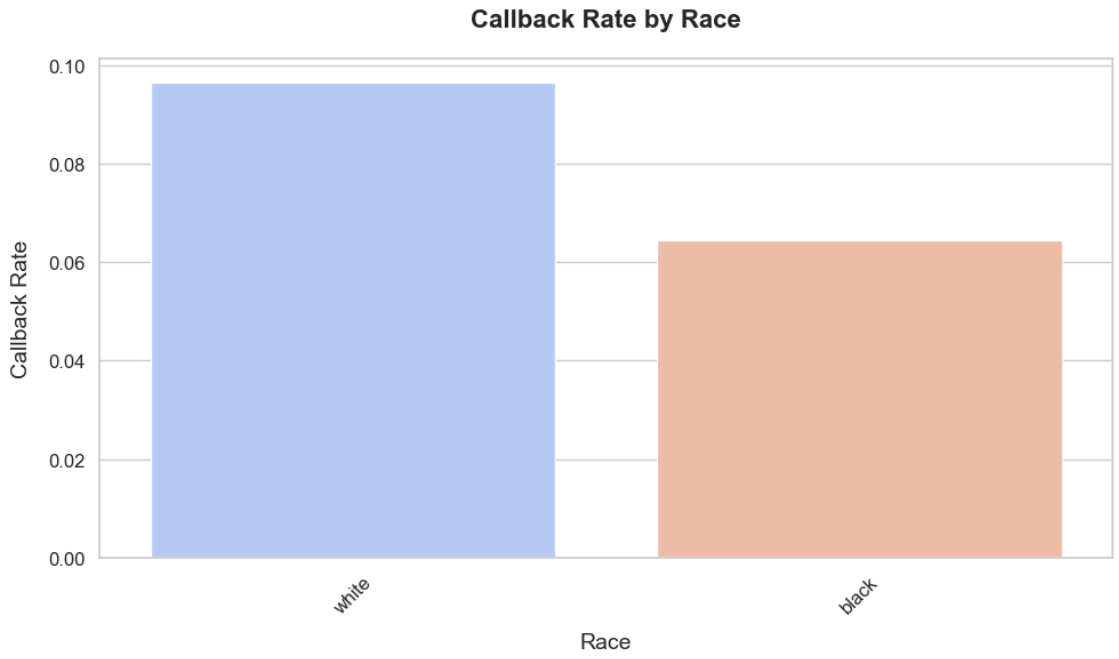

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



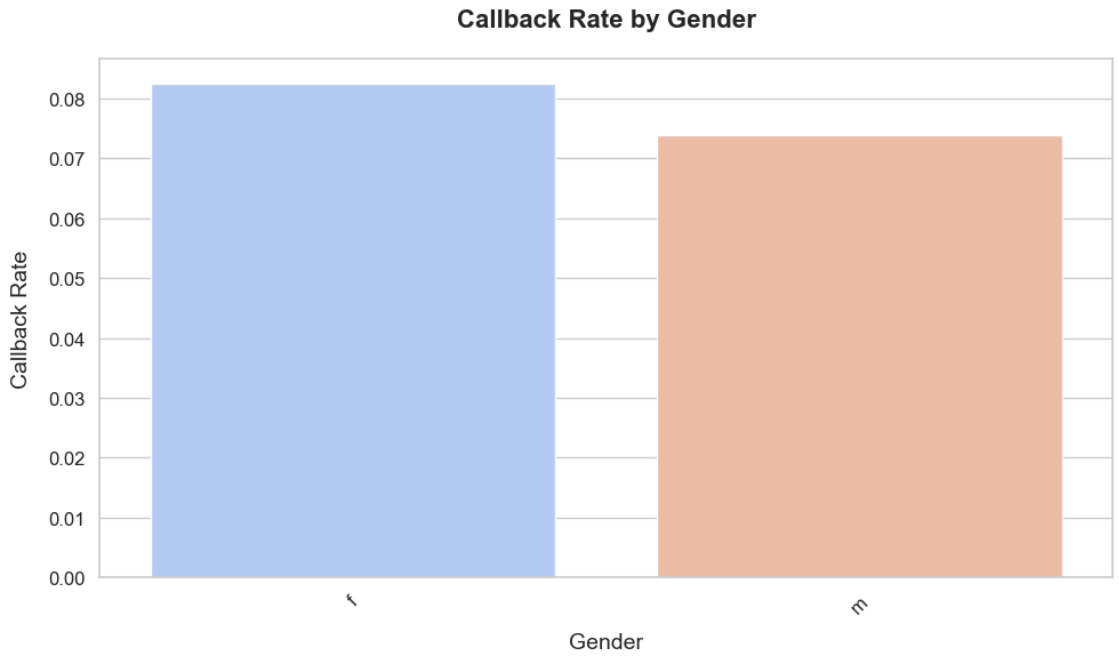

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



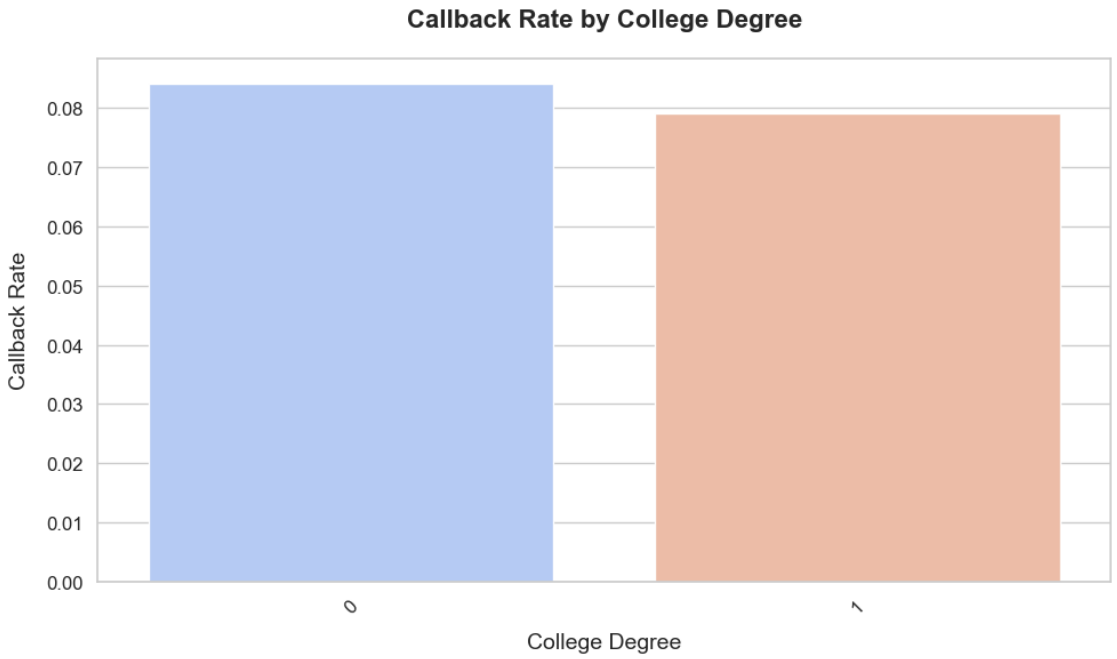

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



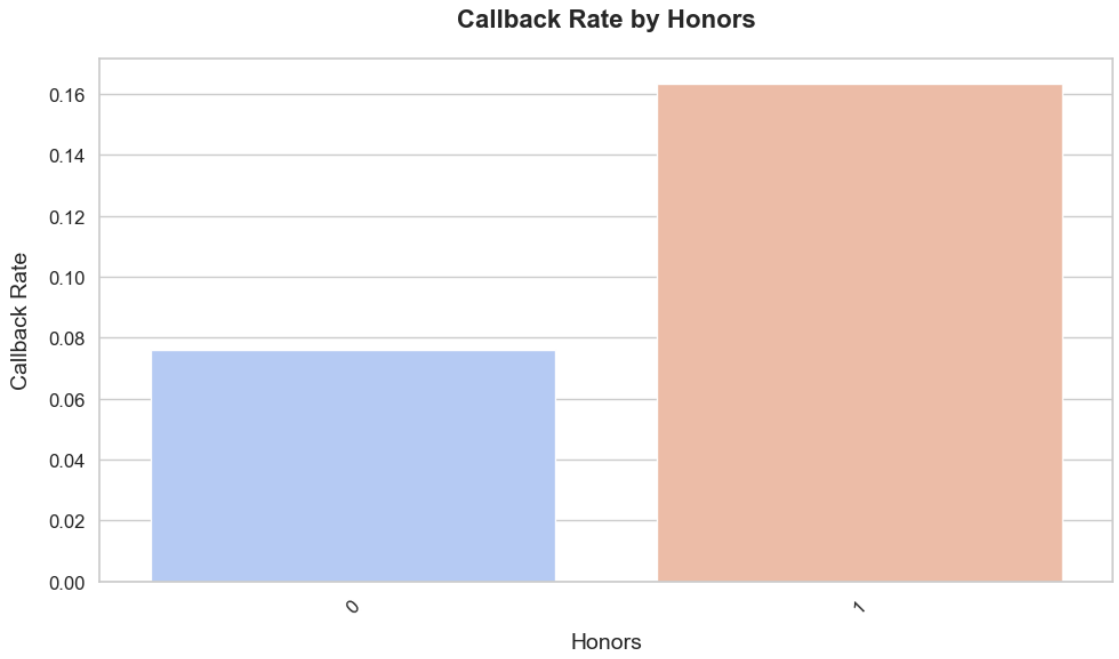

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



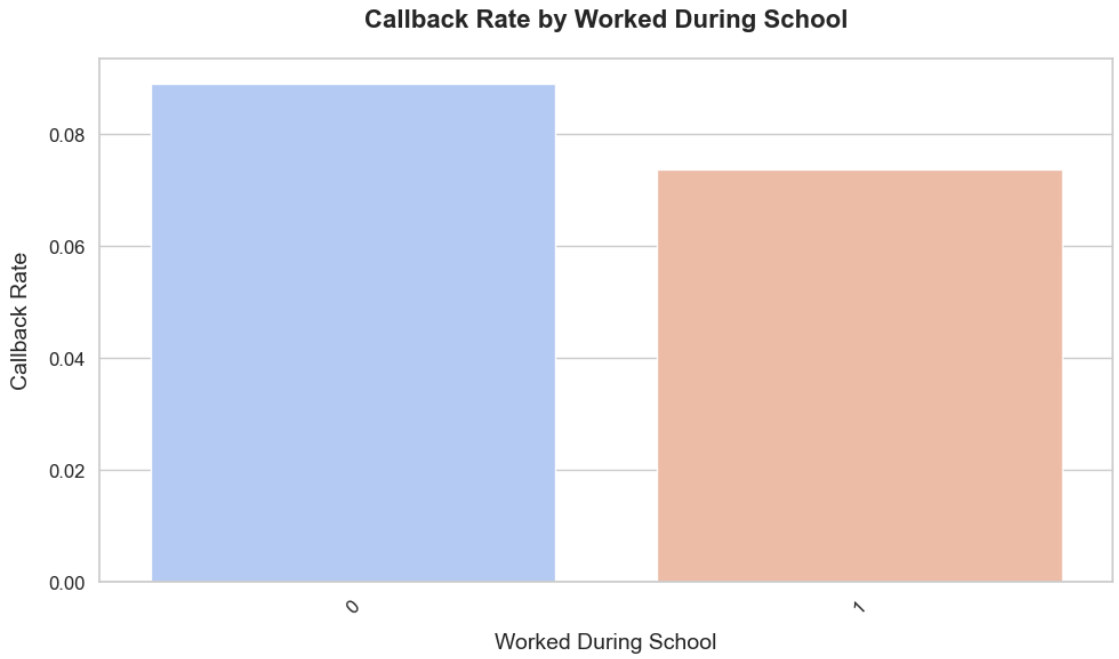

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



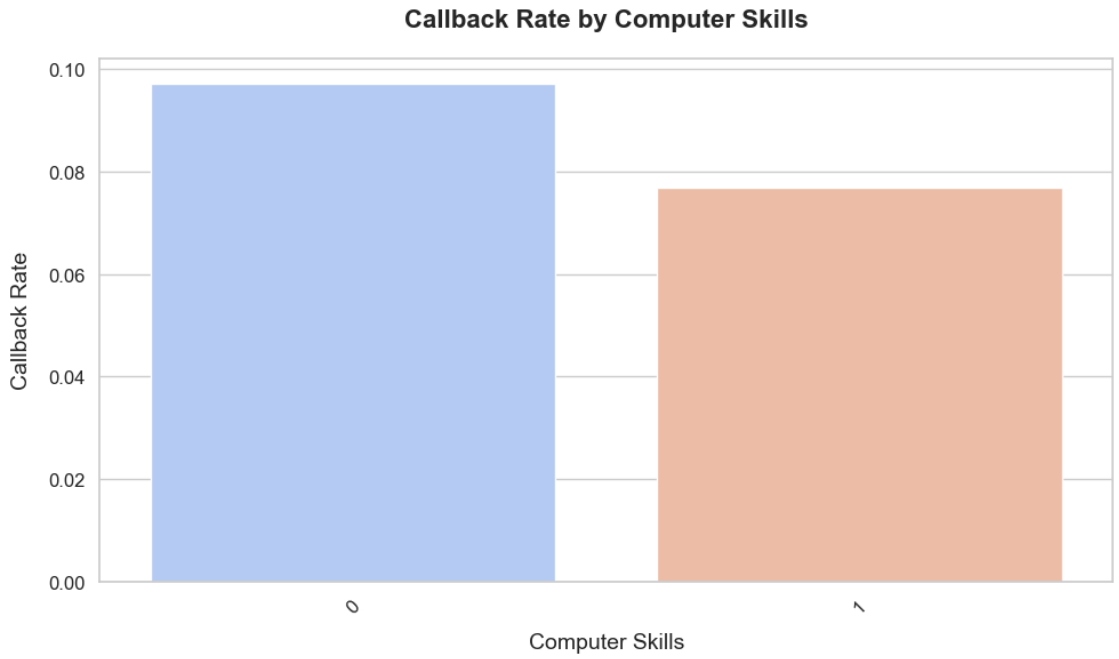

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



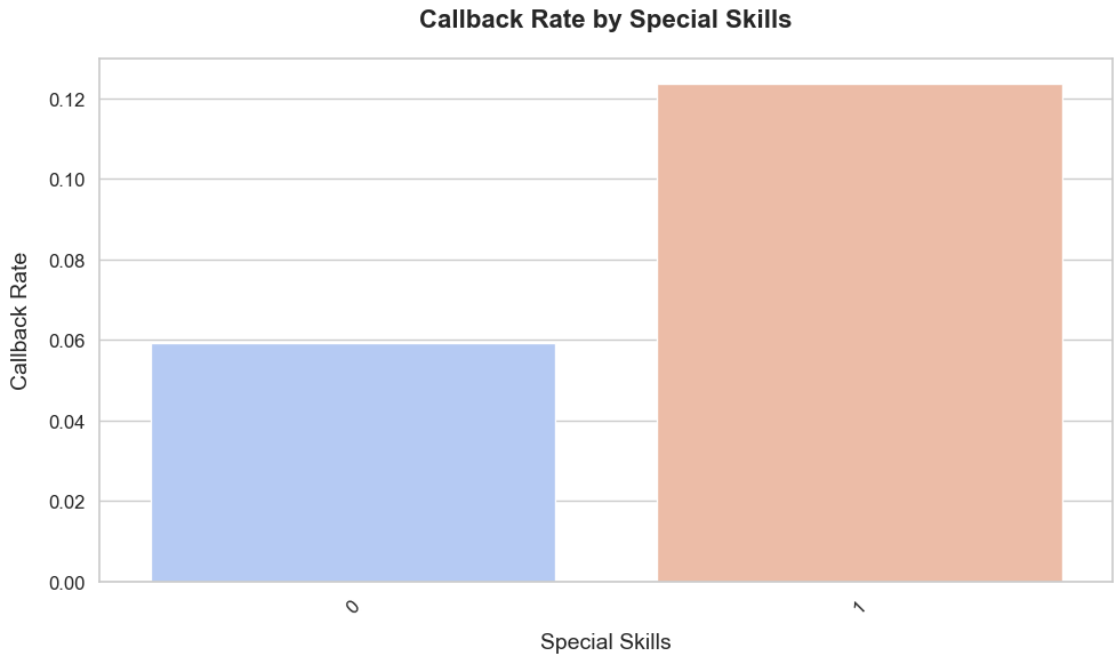

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



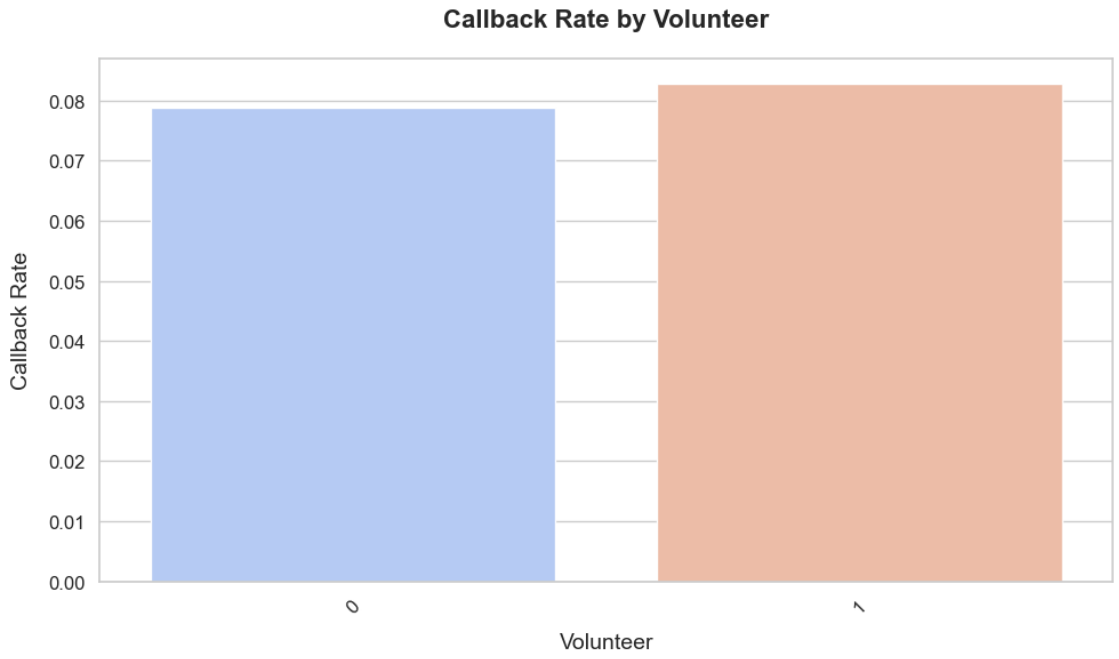

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



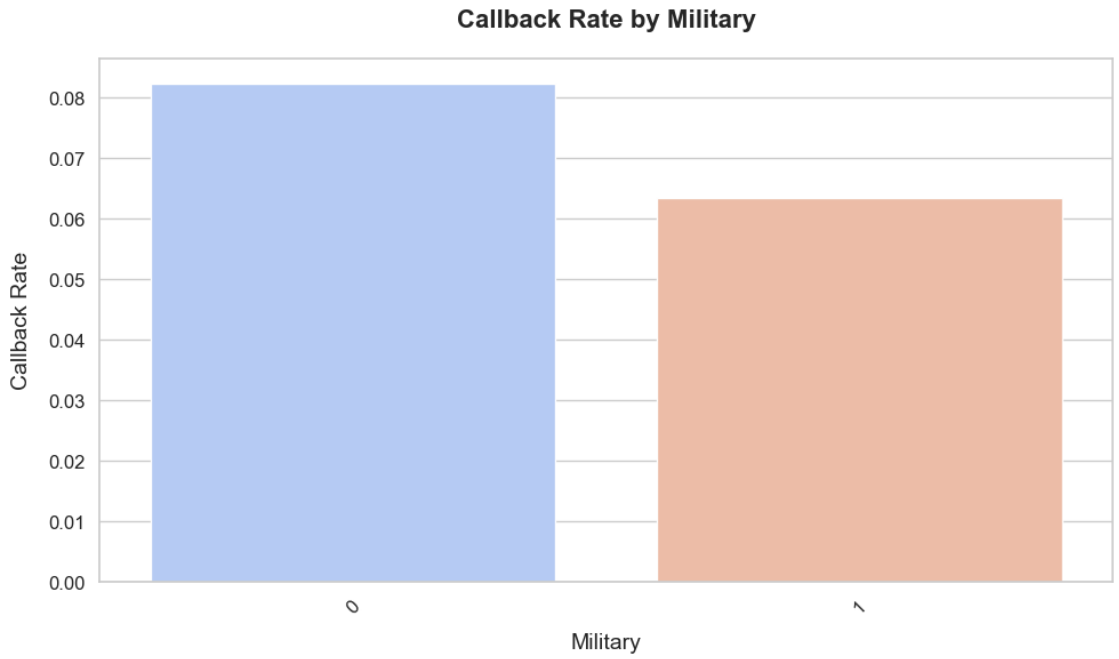

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



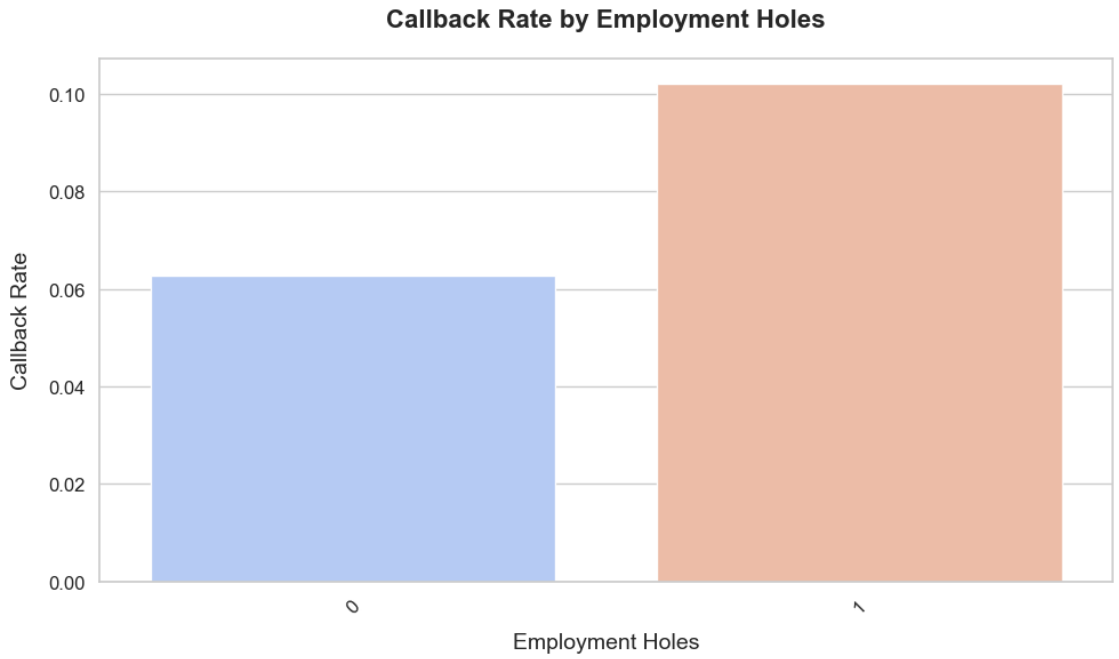

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



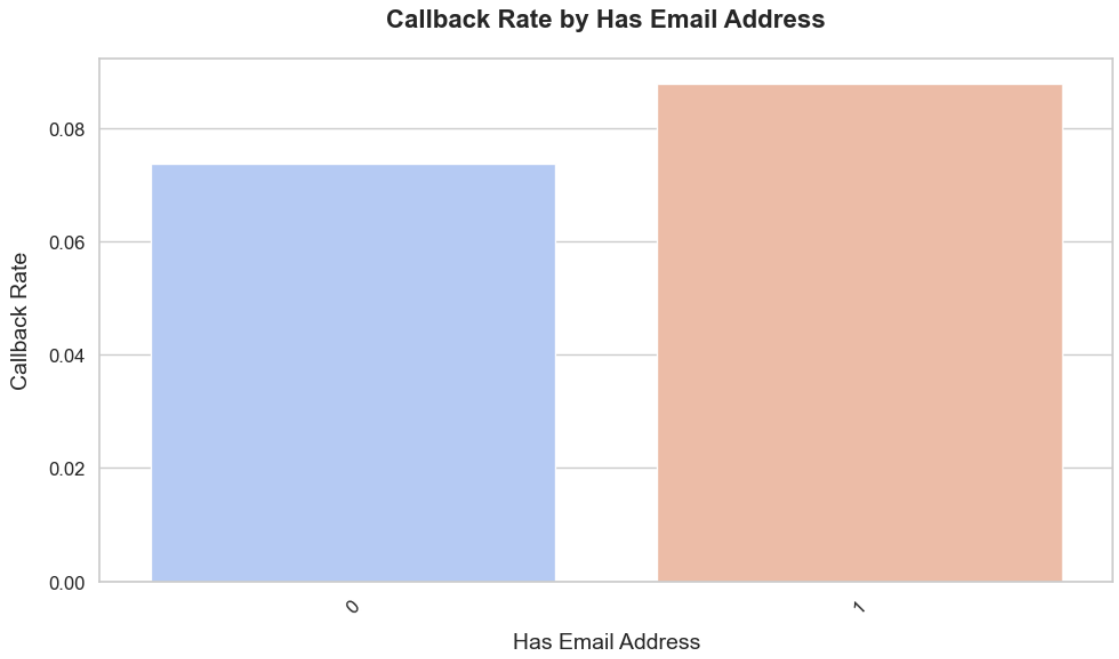

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



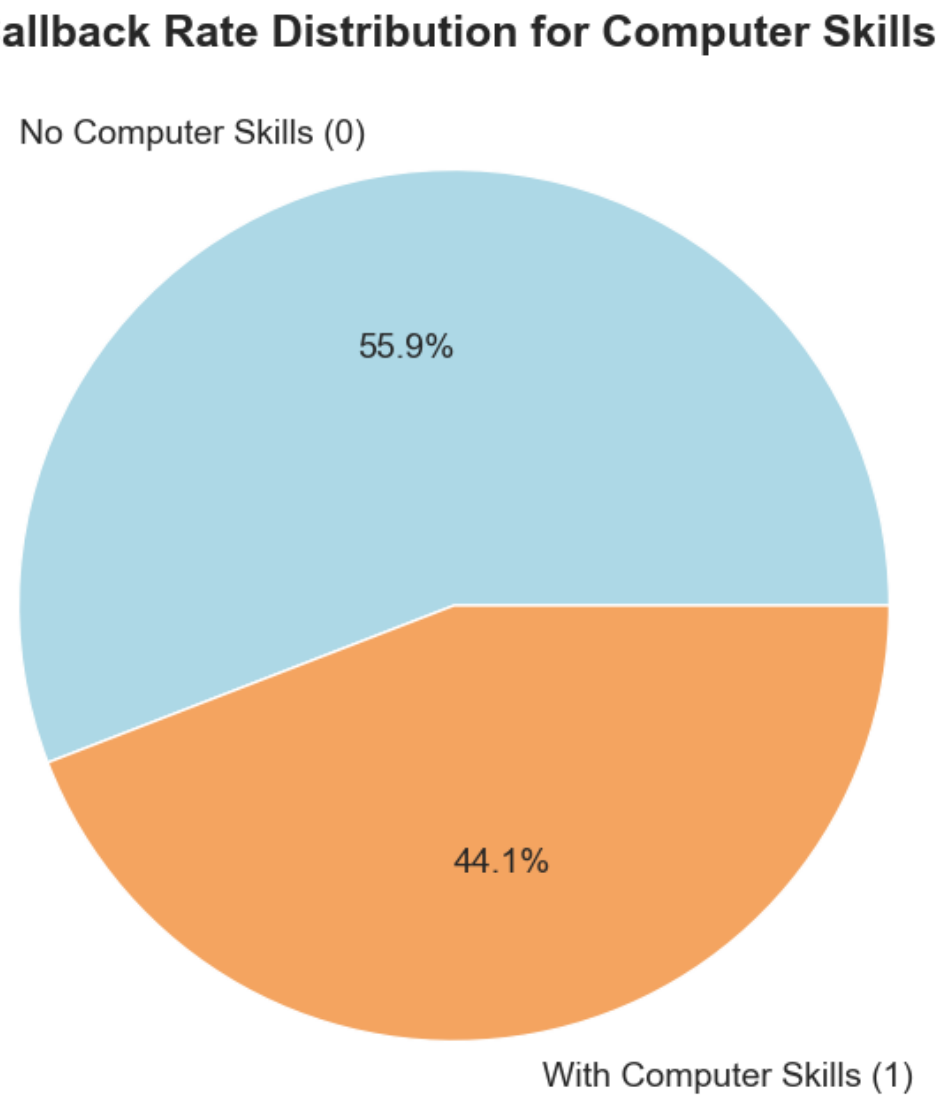

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```




```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=feature, y='received_callback', data=df, ci=None,
palette='coolwarm')
```



[17]:
```
# Pie chart for callback rates by computer skills with legend
def pie_chart_plot(feature, df):
    callback_counts = df.groupby(feature)['received_callback'].mean()
    labels = ['No Computer Skills (0)', 'With Computer Skills (1)']

    # Create pie chart
    callback_counts.plot(kind='pie', labels=labels, autopct='%1.1f%%',␣
 ↪colors=['#ADD8E6', '#F4A460'], figsize=(8, 8), textprops={'fontsize': 14})

    # Title and subtitle
    plt.title(f'Callback Rate Distribution for {feature.replace("_", " ").
 ↪title()}', fontsize=18, fontweight='bold')
    plt.ylabel('')  # Remove default y-label

    plt.show()

# Call the function
pie_chart_plot('computer_skills', df)
```



[18]:
```
from sklearn.model_selection import train_test_split

# Get unique values for all columns
unique_values_all_columns = {col: df[col].unique() for col in df.columns}
for col, unique_values in unique_values_all_columns.items():
    print(f"Unique values in column {col}: {unique_values}")
```

```
Unique values in column job_ad_id: [ 384  385  386 …  381 1344  382]
Unique values in column job_city: ['Chicago' 'Boston']
Unique values in column job_industry: ['manufacturing' 'other_service'
'wholesale_and_retail_trade'
 'business_and_personal_service' 'finance_insurance_real_estate'
```

```
 'transportation_communication']
Unique values in column job_type: ['supervisor' 'secretary' 'sales_rep'
'retail_sales' 'manager' 'clerical']
Unique values in column job_fed_contractor: [nan  0.  1.]
Unique values in column job_equal_opp_employer: [1 0]
Unique values in column job_ownership: ['unknown' 'nonprofit' 'private'
'public']
Unique values in column job_req_any: [1 0]
Unique values in column job_req_communication: [0 1]
Unique values in column job_req_education: [0 1]
Unique values in column job_req_min_experience: ['5' 'some' nan '3' '2' '1' '8'
'7' '0.5' '10' '0' '4' '6']
Unique values in column job_req_computer: [1 0]
Unique values in column job_req_organization: [0 1]
Unique values in column job_req_school: ['none_listed' 'some_college' 'college'
'high_school_grad']
Unique values in column received_callback: [0 1]
Unique values in column firstname: ['Allison' 'Kristen' 'Lakisha' 'Latonya'
'Carrie' 'Jay' 'Jill' 'Kenya'
 'Tyrone' 'Aisha' 'Geoffrey' 'Matthew' 'Tamika' 'Leroy' 'Todd' 'Greg'
 'Keisha' 'Brad' 'Laurie' 'Meredith' 'Anne' 'Emily' 'Latoya' 'Ebony'
 'Brendan' 'Hakim' 'Jamal' 'Neil' 'Tremayne' 'Brett' 'Darnell' 'Sarah'
 'Jermaine' 'Tanisha' 'Rasheed' 'Kareem']
Unique values in column race: ['white' 'black']
Unique values in column gender: ['f' 'm']
Unique values in column years_college: [4 3 1 2 0]
Unique values in column college_degree: [1 0]
Unique values in column honors: [0 1]
Unique values in column worked_during_school: [0 1]
Unique values in column years_experience: [ 6 22  5 21  3  8  4  2  7  9 13 19
12 11 10 23  1 14 18 26 15 25 16 20
 17 44]
Unique values in column computer_skills: [1 0]
Unique values in column special_skills: [0 1]
Unique values in column volunteer: [0 1]
Unique values in column military: [0 1]
Unique values in column employment_holes: [1 0]
Unique values in column has_email_address: [0 1]
Unique values in column resume_quality: ['low' 'high']
```

[19]:
```
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder, OrdinalEncoder
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
```

```
y = df['received_callback']
X = df.drop(['received_callback', 'firstname','job_ad_id'], axis=1)

random_state = 42

X_train, X_other, y_train, y_other = train_test_split(X,y,train_size = 0.
 ↪6,random_state=random_state)

X_val, X_test, y_val, y_test = train_test_split(X_other,y_other,train_size = 0.
 ↪5,random_state=random_state)

# collect which encoder to use on each feature
# needs to be done manually
ordinal_ftrs =␣
 ↪['job_req_min_experience','job_req_school','years_college','years_experience','resume_quali
ordinal_job_req_min = ['unknown','0','0.
 ↪5','1','2','3','4','5','some','6','7','8','10']
ordinal_job_req_school =␣
 ↪['none_listed','high_school_grad','some_college','college']
ordinal_years_college = [0,1,2,3,4]
ordinal_years_experience = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,␣
 ↪16, 17, 18, 19, 20, 21, 22, 23, 25, 26, 44]
ordinal_resume_quality = ['low','high']

onehot_ftrs =␣
 ↪['job_city','job_industry','job_type','job_fed_contractor','job_equal_opp_employer','job_ow
              ␣
 ↪'job_req_communication','job_req_education','job_req_computer','job_req_organization','race
               ␣
 ↪'worked_during_school','computer_skills','special_skills','volunteer','military','employmen

imputer = SimpleImputer(strategy='constant', fill_value='unknown')

# collect all the encoders
preprocessor = ColumnTransformer(
    transformers=[
        ('impute_ord', Pipeline(steps=[
            ('imputer', imputer),  # First impute
            ('ordinal', OrdinalEncoder(categories=[ordinal_job_req_min,␣
 ↪ordinal_job_req_school, ordinal_years_college,
                                                   ordinal_years_experience,␣
 ↪ordinal_resume_quality]))
        ]), ordinal_ftrs),
        ('onehot', OneHotEncoder(sparse_output=False, handle_unknown='ignore'),␣
 ↪onehot_ftrs)
    ])
```

```
clf = Pipeline(steps=[('preprocessor', preprocessor)]) # for now we only␣
 ↪preprocess
                                                       # later on we will add␣
 ↪other steps here

X_train_prep = clf.fit_transform(X_train)
X_val_prep = clf.transform(X_val)
X_test_prep = clf.transform(X_test)

ordinal_feature_names = ordinal_ftrs
onehot_feature_names = clf.named_steps['preprocessor'].
 ↪named_transformers_['onehot'].get_feature_names_out(onehot_ftrs)
all_feature_names = list(ordinal_feature_names) + list(onehot_feature_names)
X_train_prep_df = pd.DataFrame(X_train_prep, columns=all_feature_names)

print(X_train.shape)
print(X_train_prep.shape)

print(X_train_prep_df.head())
```

```
(2922, 27)
(2922, 60)
   job_req_min_experience  job_req_school  years_college  years_experience  \
0                     0.0             0.0            2.0               3.0
1                     0.0             0.0            4.0               6.0
2                     0.0             0.0            4.0               6.0
3                     0.0             0.0            4.0              10.0
4                     0.0             0.0            4.0               6.0

   resume_quality  job_city_Boston  job_city_Chicago  \
0             0.0              0.0               1.0
1             1.0              0.0               1.0
2             0.0              1.0               0.0
3             1.0              0.0               1.0
4             0.0              1.0               0.0

   job_industry_business_and_personal_service  \
0                                         0.0
1                                         1.0
2                                         0.0
3                                         0.0
4                                         0.0

   job_industry_finance_insurance_real_estate  job_industry_manufacturing  \
0                                         0.0                         1.0
1                                         0.0                         0.0
```

```
2                                          0.0                        0.0
3                                          0.0                        0.0
4                                          0.0                        0.0

   …  special_skills_0  special_skills_1  volunteer_0  volunteer_1  \
0  …               0.0               1.0          1.0          0.0
1  …               1.0               0.0          0.0          1.0
2  …               1.0               0.0          1.0          0.0
3  …               1.0               0.0          0.0          1.0
4  …               1.0               0.0          1.0          0.0

   military_0  military_1  employment_holes_0  employment_holes_1  \
0         1.0         0.0                 1.0                 0.0
1         1.0         0.0                 0.0                 1.0
2         1.0         0.0                 0.0                 1.0
3         1.0         0.0                 1.0                 0.0
4         1.0         0.0                 0.0                 1.0

   has_email_address_0  has_email_address_1
0                  1.0                  0.0
1                  0.0                  1.0
2                  1.0                  0.0
3                  0.0                  1.0
4                  1.0                  0.0

[5 rows x 60 columns]
```

[ ]:
```python
# stratified and K Fold splitting

from sklearn.model_selection import StratifiedKFold
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder, OrdinalEncoder
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer

df = pd.read_csv("resume.csv")

# assign X and y
y = df['received_callback']
X = df.drop(['received_callback', 'firstname','job_ad_id'], axis=1)

# set random state
random_state = 42

# split test set to use at the end
```

```
X_other, X_test, y_other, y_test = train_test_split(X,y,test_size = 0.
 ↪2,stratify=y,random_state=random_state)
print('test balance:',np.unique(y_test,return_counts=True))

# do StratifiedKFold split on other
kf = StratifiedKFold(n_splits=4,shuffle=True,random_state=random_state)
for train_index, val_index in kf.split(X_other,y_other):
    print('new fold')
    X_train = X_other.iloc[train_index]
    y_train = y_other.iloc[train_index]
    X_val = X_other.iloc[val_index]
    y_val = y_other.iloc[val_index]
    print(np.unique(y_train,return_counts=True))
    print(np.unique(y_val,return_counts=True))

ordinal_ftrs =␣
 ↪['job_req_min_experience','job_req_school','years_college','years_experience','resume_quali
ordinal_job_req_min = ['unknown','0','0.
 ↪5','1','2','3','4','5','some','6','7','8','10']
ordinal_job_req_school =␣
 ↪['none_listed','high_school_grad','some_college','college']
ordinal_years_college = [0,1,2,3,4]
ordinal_years_experience = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,␣
 ↪16, 17, 18, 19, 20, 21, 22, 23, 25, 26, 44]
ordinal_resume_quality = ['low','high']

onehot_ftrs =␣
 ↪['job_city','job_industry','job_type','job_fed_contractor','job_equal_opp_employer','job_ow
              ␣
 ↪'job_req_communication','job_req_education','job_req_computer','job_req_organization','race
                ␣
 ↪'worked_during_school','computer_skills','special_skills','volunteer','military','employmen

imputer = SimpleImputer(strategy='constant', fill_value='unknown')

# collect all the encoders
preprocessor = ColumnTransformer(
    transformers=[
        ('impute_ord', Pipeline(steps=[
            ('imputer', imputer),  # First impute
            ('ordinal', OrdinalEncoder(categories=[ordinal_job_req_min,␣
 ↪ordinal_job_req_school, ordinal_years_college,
                                                   ordinal_years_experience,␣
 ↪ordinal_resume_quality]))
        ]), ordinal_ftrs),
```

```
        ('onehot', OneHotEncoder(sparse_output=False, handle_unknown='ignore'),␣
 ↪onehot_ftrs)
    ])

clf = Pipeline(steps=[('preprocessor', preprocessor)]) # for now we only␣
 ↪preprocess
                                                       # later on we will add␣
 ↪other steps here

X_train_prep = clf.fit_transform(X_train)
X_val_prep = clf.transform(X_val)
X_test_prep = clf.transform(X_test)

ordinal_feature_names = ordinal_ftrs
onehot_feature_names = clf.named_steps['preprocessor'].
 ↪named_transformers_['onehot'].get_feature_names_out(onehot_ftrs)
all_feature_names = list(ordinal_feature_names) + list(onehot_feature_names)
X_train_prep_df = pd.DataFrame(X_train_prep, columns=all_feature_names)

print()
print("X_train shape:", X_train.shape)
print("X_train after preprocessing:", X_train_prep.shape)

print()
print(X_train_prep_df.head())
```

```
test balance: (array([0, 1]), array([896,  78]))
new fold
(array([0, 1]), array([2687,  235]))
(array([0, 1]), array([895,  79]))
new fold
(array([0, 1]), array([2687,  235]))
(array([0, 1]), array([895,  79]))
new fold
(array([0, 1]), array([2686,  236]))
(array([0, 1]), array([896,  78]))
new fold
(array([0, 1]), array([2686,  236]))
(array([0, 1]), array([896,  78]))

X_train shape: (2922, 27)
X_train after preprocessing: (2922, 60)

   job_req_min_experience  job_req_school  years_college  years_experience  \
0                     4.0             0.0            4.0               7.0
1                     8.0             0.0            4.0               6.0
2                     4.0             0.0            3.0               5.0
```

```
3                    0.0             0.0            4.0              4.0
4                    0.0             3.0            4.0             13.0

   resume_quality  job_city_Boston  job_city_Chicago  \
0             1.0              0.0               1.0
1             0.0              1.0               0.0
2             0.0              0.0               1.0
3             1.0              1.0               0.0
4             0.0              0.0               1.0

   job_industry_business_and_personal_service  \
0                                         0.0
1                                         1.0
2                                         0.0
3                                         0.0
4                                         0.0

   job_industry_finance_insurance_real_estate  job_industry_manufacturing  \
0                                         0.0                         0.0
1                                         0.0                         0.0
2                                         0.0                         0.0
3                                         0.0                         0.0
4                                         0.0                         0.0

   …  special_skills_0  special_skills_1  volunteer_0  volunteer_1  \
0  …               0.0               1.0          1.0          0.0
1  …               1.0               0.0          1.0          0.0
2  …               1.0               0.0          1.0          0.0
3  …               0.0               1.0          0.0          1.0
4  …               1.0               0.0          1.0          0.0

   military_0  military_1  employment_holes_0  employment_holes_1  \
0         1.0         0.0                 0.0                 1.0
1         1.0         0.0                 0.0                 1.0
2         1.0         0.0                 1.0                 0.0
3         1.0         0.0                 1.0                 0.0
4         1.0         0.0                 1.0                 0.0

   has_email_address_0  has_email_address_1
0                  0.0                  1.0
1                  1.0                  0.0
2                  1.0                  0.0
3                  1.0                  0.0
4                  1.0                  0.0

[5 rows x 60 columns]
```

[3]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import StratifiedKFold, train_test_split,␣
 ↪GridSearchCV
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline, make_pipeline
from sklearn.preprocessing import OneHotEncoder, OrdinalEncoder
from sklearn.impute import SimpleImputer
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.metrics import mean_squared_error, accuracy_score, fbeta_score,␣
 ↪make_scorer

# function for the ML pipeline as outlined above
def MLpipe_KFold_Accuracy(X, y, ML_algo, param_grid):

    # lists to be returned
    test_scores = []
    best_models = []

    models = []

    # preprocessor
    ordinal_ftrs =␣
 ↪['job_req_min_experience','job_req_school','years_college','years_experience','resume_quali
    ordinal_job_req_min = ['unknown','0','0.
 ↪5','1','2','3','4','5','some','6','7','8','10']
    ordinal_job_req_school =␣
 ↪['none_listed','high_school_grad','some_college','college']
    ordinal_years_college = [0,1,2,3,4]
    ordinal_years_experience = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,␣
 ↪15, 16, 17, 18, 19, 20, 21, 22, 23, 25, 26, 44]
    ordinal_resume_quality = ['low','high']

    onehot_ftrs =␣
 ↪['job_city','job_industry','job_type','job_fed_contractor','job_equal_opp_employer','job_ow
                 ␣
 ↪'job_req_communication','job_req_education','job_req_computer','job_req_organization','race
                     ␣
 ↪'worked_during_school','computer_skills','special_skills','volunteer','military','employmen

    imputer = SimpleImputer(strategy='constant', fill_value='unknown')

    # collect all the encoders
```

```
    preprocessor = ColumnTransformer(
        transformers=[
            ('impute_ord', Pipeline(steps=[
                ('imputer', imputer),  # First impute
                ('ordinal', OrdinalEncoder(categories=[ordinal_job_req_min,␣
↪ordinal_job_req_school, ordinal_years_college,
                                                       ordinal_years_experience,␣
↪ordinal_resume_quality]))
            ]), ordinal_ftrs),
            ('onehot', OneHotEncoder(sparse_output=False,␣
↪handle_unknown='ignore'), onehot_ftrs)
        ])

    # 10 loops
    for random_state in range(1,11):

        X_other, X_test, y_other, y_test = train_test_split(X,y,test_size = 0.
↪2,stratify=y,random_state=random_state*42)

        pipe = make_pipeline(preprocessor,ML_algo)

        kf = StratifiedKFold(n_splits=4,shuffle=True,random_state=random_state)

        grid = GridSearchCV(pipe, param_grid=param_grid, scoring =␣
↪make_scorer(fbeta_score, beta=1),
                            cv=kf, return_train_score = True, n_jobs=-1,␣
↪verbose=True)

        grid.fit(X_other, y_other)

        y_test_pred = grid.predict(X_test)
        test_score = accuracy_score(y_test, y_test_pred)

        test_scores.append(test_score)
        best_models.append(grid.best_estimator_)

    print(f"Mean Test Accuracy: {np.mean(test_scores)}")
    print(f"Standard Deviation of Test Accuracy: {np.std(test_scores)}")

    return test_scores, best_models
```

[4]:
```
# read data
df = pd.read_csv("resume.csv")

# assign X and y
y = df['received_callback']
X = df.drop(['received_callback', 'firstname','job_ad_id'], axis=1)
```

[12]:
```
# logistic
print("----------------Logistic Regression-----------------")
param_grid_lr = {'logisticregression__C': [0.1, 1, 10, 100],
                 'logisticregression__penalty': ['l2', 'l1'],
                 'logisticregression__class_weight': ['balanced', None],
                 'logisticregression__solver': ['saga']}
log_reg = LogisticRegression(max_iter=10000)
MLpipe_KFold_Accuracy(X, y, ML_algo=log_reg, param_grid=param_grid_lr)
```

```
----------------Logistic Regression-----------------
Fitting 4 folds for each of 16 candidates, totalling 64 fits

Fitting 4 folds for each of 16 candidates, totalling 64 fits
Fitting 4 folds for each of 16 candidates, totalling 64 fits
Fitting 4 folds for each of 16 candidates, totalling 64 fits
Fitting 4 folds for each of 16 candidates, totalling 64 fits
Fitting 4 folds for each of 16 candidates, totalling 64 fits
Fitting 4 folds for each of 16 candidates, totalling 64 fits
Fitting 4 folds for each of 16 candidates, totalling 64 fits
Fitting 4 folds for each of 16 candidates, totalling 64 fits
Fitting 4 folds for each of 16 candidates, totalling 64 fits
Mean Test Accuracy: 0.919917864476386
Standard Deviation of Test Accuracy: 0.0
```

[12]:
```
([0.919917864476386,
  0.919917864476386,
  0.919917864476386,
  0.919917864476386,
  0.919917864476386,
  0.919917864476386,
  0.919917864476386,
  0.919917864476386,
  0.919917864476386,
  0.919917864476386],
 [Pipeline(steps=[('columntransformer',
                   ColumnTransformer(transformers=[('impute_ord',
                                                    Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
   strategy='constant')),
                                                                    ('ordinal',
OrdinalEncoder(categories=[['unknown',
                            '0',
                            '0.5',
                            '1',
                            '2',
                            '3',
                            '4',
                            '5',
```

```
                 'some',
                 '6',
                 '7',
                 '8',
                 '10'],
                ['none_listed',
                 'high_school_grad',
                 'some_college',
                 'college'],
                [0,
                 1,
                 2,
                 3,
                 4],
                [1,
                 2,
                 3,…
                                                  'job_ownership',
                                                  'job_req_any',
                                                  'job_req_communication',
                                                  'job_req_education',
                                                  'job_req_computer',
                                                  'job_req_organization',
                                                  'race', 'gender',
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                  ('logisticregression',
                   LogisticRegression(C=1, max_iter=10000, solver='saga'))]),
  Pipeline(steps=[('columntransformer',
                   ColumnTransformer(transformers=[('impute_ord',
                                                    Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
   strategy='constant')),
                                                                    ('ordinal',
OrdinalEncoder(categories=[['unknown',
                 '0',
                 '0.5',
                 '1',
                 '2',
                 '3',
                 '4',
                 '5',
```

```
                 'some',
                 '6',
                 '7',
                 '8',
                 '10'],
                ['none_listed',
                 'high_school_grad',
                 'some_college',
                 'college'],
                [0,
                 1,
                 2,
                 3,
                 4],
                [1,
                 2,
                 3,…
                                                  'job_ownership',
                                                  'job_req_any',
                                                  'job_req_communication',
                                                  'job_req_education',
                                                  'job_req_computer',
                                                  'job_req_organization',
                                                  'race', 'gender',
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                  ('logisticregression',
                   LogisticRegression(C=0.1, max_iter=10000, solver='saga'))]),
  Pipeline(steps=[('columntransformer',
                   ColumnTransformer(transformers=[('impute_ord',
                                                    Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
   strategy='constant')),
                                                                    ('ordinal',
OrdinalEncoder(categories=[['unknown',
                 '0',
                 '0.5',
                 '1',
                 '2',
                 '3',
                 '4',
                 '5',
```

```
                 'some',
                 '6',
                 '7',
                 '8',
                 '10'],
                ['none_listed',
                 'high_school_grad',
                 'some_college',
                 'college'],
                [0,
                 1,
                 2,
                 3,
                 4],
                [1,
                 2,
                 3,…
                                                  'job_ownership',
                                                  'job_req_any',
                                                  'job_req_communication',
                                                  'job_req_education',
                                                  'job_req_computer',
                                                  'job_req_organization',
                                                  'race', 'gender',
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                  ('logisticregression',
                   LogisticRegression(C=0.1, max_iter=10000, solver='saga'))]),
  Pipeline(steps=[('columntransformer',
                   ColumnTransformer(transformers=[('impute_ord',
                                                    Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
   strategy='constant')),
                                                                    ('ordinal',
OrdinalEncoder(categories=[['unknown',
                 '0',
                 '0.5',
                 '1',
                 '2',
                 '3',
                 '4',
                 '5',
```

```
                 'some',
                 '6',
                 '7',
                 '8',
                 '10'],
                ['none_listed',
                 'high_school_grad',
                 'some_college',
                 'college'],
                [0,
                 1,
                 2,
                 3,
                 4],
                [1,
                 2,
                 3,…
                                                  'job_ownership',
                                                  'job_req_any',
                                                  'job_req_communication',
                                                  'job_req_education',
                                                  'job_req_computer',
                                                  'job_req_organization',
                                                  'race', 'gender',
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                  ('logisticregression',
                   LogisticRegression(C=0.1, max_iter=10000, solver='saga'))]),
  Pipeline(steps=[('columntransformer',
                   ColumnTransformer(transformers=[('impute_ord',
                                                    Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
   strategy='constant')),
                                                                    ('ordinal',
OrdinalEncoder(categories=[['unknown',
                 '0',
                 '0.5',
                 '1',
                 '2',
                 '3',
                 '4',
                 '5',
```

```
                 'some',
                 '6',
                 '7',
                 '8',
                 '10'],
                ['none_listed',
                 'high_school_grad',
                 'some_college',
                 'college'],
                [0,
                 1,
                 2,
                 3,
                 4],
                [1,
                 2,
                 3,…
                                                  'job_req_any',
                                                  'job_req_communication',
                                                  'job_req_education',
                                                  'job_req_computer',
                                                  'job_req_organization',
                                                  'race', 'gender',
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                  ('logisticregression',
                   LogisticRegression(C=0.1, max_iter=10000, penalty='l1',
                                      solver='saga'))]),
  Pipeline(steps=[('columntransformer',
                   ColumnTransformer(transformers=[('impute_ord',
                                                    Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
   strategy='constant')),
                                                                    ('ordinal',
OrdinalEncoder(categories=[['unknown',
                 '0',
                 '0.5',
                 '1',
                 '2',
                 '3',
                 '4',
                 '5',
```

```
                 'some',
                 '6',
                 '7',
                 '8',
                 '10'],
                ['none_listed',
                 'high_school_grad',
                 'some_college',
                 'college'],
                [0,
                 1,
                 2,
                 3,
                 4],
                [1,
                 2,
                 3,…
                                                  'job_ownership',
                                                  'job_req_any',
                                                  'job_req_communication',
                                                  'job_req_education',
                                                  'job_req_computer',
                                                  'job_req_organization',
                                                  'race', 'gender',
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                  ('logisticregression',
                   LogisticRegression(C=10, max_iter=10000, solver='saga'))]),
  Pipeline(steps=[('columntransformer',
                   ColumnTransformer(transformers=[('impute_ord',
                                                    Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
   strategy='constant')),
                                                                    ('ordinal',
OrdinalEncoder(categories=[['unknown',
                 '0',
                 '0.5',
                 '1',
                 '2',
                 '3',
                 '4',
                 '5',
```

```
                 'some',
                 '6',
                 '7',
                 '8',
                 '10'],
                ['none_listed',
                 'high_school_grad',
                 'some_college',
                 'college'],
                [0,
                 1,
                 2,
                 3,
                 4],
                [1,
                 2,
                 3,…
                                                  'job_ownership',
                                                  'job_req_any',
                                                  'job_req_communication',
                                                  'job_req_education',
                                                  'job_req_computer',
                                                  'job_req_organization',
                                                  'race', 'gender',
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                  ('logisticregression',
                   LogisticRegression(C=0.1, max_iter=10000, solver='saga'))]),
  Pipeline(steps=[('columntransformer',
                   ColumnTransformer(transformers=[('impute_ord',
                                                    Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
   strategy='constant')),
                                                                    ('ordinal',
OrdinalEncoder(categories=[['unknown',
                 '0',
                 '0.5',
                 '1',
                 '2',
                 '3',
                 '4',
                 '5',
```

```
                 'some',
                 '6',
                 '7',
                 '8',
                 '10'],
                ['none_listed',
                 'high_school_grad',
                 'some_college',
                 'college'],
                [0,
                 1,
                 2,
                 3,
                 4],
                [1,
                 2,
                 3,…
                                                  'job_ownership',
                                                  'job_req_any',
                                                  'job_req_communication',
                                                  'job_req_education',
                                                  'job_req_computer',
                                                  'job_req_organization',
                                                  'race', 'gender',
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                  ('logisticregression',
                   LogisticRegression(C=0.1, max_iter=10000, solver='saga'))]),
  Pipeline(steps=[('columntransformer',
                   ColumnTransformer(transformers=[('impute_ord',
                                                    Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
   strategy='constant')),
                                                                    ('ordinal',
OrdinalEncoder(categories=[['unknown',
                 '0',
                 '0.5',
                 '1',
                 '2',
                 '3',
                 '4',
                 '5',
```

```
                 'some',
                 '6',
                 '7',
                 '8',
                 '10'],
                ['none_listed',
                 'high_school_grad',
                 'some_college',
                 'college'],
                [0,
                 1,
                 2,
                 3,
                 4],
                [1,
                 2,
                 3,…
                                                  'job_ownership',
                                                  'job_req_any',
                                                  'job_req_communication',
                                                  'job_req_education',
                                                  'job_req_computer',
                                                  'job_req_organization',
                                                  'race', 'gender',
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                  ('logisticregression',
                   LogisticRegression(C=0.1, max_iter=10000, solver='saga'))]),
  Pipeline(steps=[('columntransformer',
                   ColumnTransformer(transformers=[('impute_ord',
                                                    Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
   strategy='constant')),
                                                                    ('ordinal',
OrdinalEncoder(categories=[['unknown',
                 '0',
                 '0.5',
                 '1',
                 '2',
                 '3',
                 '4',
                 '5',
```

```
                 'some',
                 '6',
                 '7',
                 '8',
                 '10'],
                ['none_listed',
                 'high_school_grad',
                 'some_college',
                 'college'],
                [0,
                 1,
                 2,
                 3,
                 4],
                [1,
                 2,
                 3,…
                                                    'job_ownership',
                                                    'job_req_any',
                                                    'job_req_communication',
                                                    'job_req_education',
                                                    'job_req_computer',
                                                    'job_req_organization',
                                                    'race', 'gender',
                                                    'college_degree', 'honors',
                                                    'worked_during_school',
                                                    'computer_skills',
                                                    'special_skills',
                                                    'volunteer', 'military',
                                                    'employment_holes',
                                                    'has_email_address'])])),
                  ('logisticregression',
                   LogisticRegression(C=0.1, max_iter=10000, solver='saga'))])])
```

[ ]:
```
# random forest classifier
print("-------------Random Forest--------------")
rf_clf = RandomForestClassifier(random_state=42)
param_grid = {
    'randomforestclassifier__n_estimators': [50, 100, 200, 500],
    'randomforestclassifier__max_depth': [1, 5, 10, 30, 50],
    'randomforestclassifier__max_samples': [0.5, 0.75, 1.0],
    'randomforestclassifier__min_samples_split': [2, 5, 10],
    'randomforestclassifier__class_weight': [None, 'balanced']
    }
print(MLpipe_KFold_Accuracy(X, y, ML_algo=rf_clf, param_grid=param_grid))
```

```
-------------Random Forest--------------
Fitting 4 folds for each of 648 candidates, totalling 2592 fits
```

```
/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 648 candidates, totalling 2592 fits

/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 648 candidates, totalling 2592 fits

/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 648 candidates, totalling 2592 fits

/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 648 candidates, totalling 2592 fits

/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 648 candidates, totalling 2592 fits

/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 648 candidates, totalling 2592 fits

/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 648 candidates, totalling 2592 fits

/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 648 candidates, totalling 2592 fits

/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 648 candidates, totalling 2592 fits

/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,
```

```
Mean Test Accuracy: 0.9198151950718685
Standard Deviation of Test Accuracy: 0.0007186858316221678
([0.9188911704312115, 0.919917864476386, 0.919917864476386, 0.919917864476386,
0.9188911704312115, 0.919917864476386, 0.9209445585215605, 0.919917864476386,
0.9188911704312115, 0.9209445585215605], [Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
                '0',
                '0.5',
                '1',
                '2',
                '3',
                '4',
                '5',
                'some',
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
               [1,
                2,
                3,…
                                                  'job_req_education',
                                                  'job_req_computer',
                                                  'job_req_organization',
                                                  'race', 'gender',
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                ('randomforestclassifier',
                 RandomForestClassifier(max_depth=20, max_samples=1.0,
```

```
                                    min_samples_split=5, n_estimators=500,
                                    random_state=42))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
                '0',
                '0.5',
                '1',
                '2',
                '3',
                '4',
                '5',
                'some',
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
               [1,
                2,
                3,…
                                                  'job_req_education',
                                                  'job_req_computer',
                                                  'job_req_organization',
                                                  'race', 'gender',
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                ('randomforestclassifier',
                 RandomForestClassifier(max_depth=30, max_samples=1.0,
                                    min_samples_split=5, n_estimators=1000,
                                    random_state=42))]),
```

```
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
               '0',
               '0.5',
               '1',
               '2',
               '3',
               '4',
               '5',
               'some',
               '6',
               '7',
               '8',
               '10'],
              ['none_listed',
               'high_school_grad',
               'some_college',
               'college'],
              [0,
               1,
               2,
               3,
               4],
              [1,
               2,
               3,…
                                                   'job_req_communication',
                                                   'job_req_education',
                                                   'job_req_computer',
                                                   'job_req_organization',
                                                   'race', 'gender',
                                                   'college_degree', 'honors',
                                                   'worked_during_school',
                                                   'computer_skills',
                                                   'special_skills',
                                                   'volunteer', 'military',
                                                   'employment_holes',
                                                   'has_email_address'])])),
                ('randomforestclassifier',
                 RandomForestClassifier(max_depth=1, max_samples=0.5,
                                        n_estimators=50, random_state=42))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
```

```
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
               '0',
               '0.5',
               '1',
               '2',
               '3',
               '4',
               '5',
               'some',
               '6',
               '7',
               '8',
               '10'],
              ['none_listed',
               'high_school_grad',
               'some_college',
               'college'],
              [0,
               1,
               2,
               3,
               4],
              [1,
               2,
               3,…
                                                   'job_req_communication',
                                                   'job_req_education',
                                                   'job_req_computer',
                                                   'job_req_organization',
                                                   'race', 'gender',
                                                   'college_degree', 'honors',
                                                   'worked_during_school',
                                                   'computer_skills',
                                                   'special_skills',
                                                   'volunteer', 'military',
                                                   'employment_holes',
                                                   'has_email_address'])])),
                ('randomforestclassifier',
                 RandomForestClassifier(max_depth=1, max_samples=0.5,
                                        n_estimators=50, random_state=42))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
```

```
 strategy='constant')),
                                                                 ('ordinal',
OrdinalEncoder(categories=[['unknown',
                '0',
                '0.5',
                '1',
                '2',
                '3',
                '4',
                '5',
                'some',
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
               [1,
                2,
                3,…
                                                 'job_req_computer',
                                                 'job_req_organization',
                                                 'race', 'gender',
                                                 'college_degree', 'honors',
                                                 'worked_during_school',
                                                 'computer_skills',
                                                 'special_skills',
                                                 'volunteer', 'military',
                                                 'employment_holes',
                                                 'has_email_address'])])),
                ('randomforestclassifier',
                 RandomForestClassifier(criterion='entropy', max_depth=20,
                                        max_samples=0.5, min_samples_split=5,
                                        n_estimators=50, random_state=42))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                 ('ordinal',
OrdinalEncoder(categories=[['unknown',
```

```
                '0',
                '0.5',
                '1',
                '2',
                '3',
                '4',
                '5',
                'some',
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
               [1,
                2,
                3,…
                                                  'job_req_communication',
                                                  'job_req_education',
                                                  'job_req_computer',
                                                  'job_req_organization',
                                                  'race', 'gender',
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                ('randomforestclassifier',
                 RandomForestClassifier(max_depth=1, max_samples=0.5,
                                        n_estimators=50, random_state=42))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
                '0',
                '0.5',
```

```
                '1',
                '2',
                '3',
                '4',
                '5',
                'some',
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
               [1,
                2,
                3,…
                                                  'job_req_communication',
                                                  'job_req_education',
                                                  'job_req_computer',
                                                  'job_req_organization',
                                                  'race', 'gender',
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                ('randomforestclassifier',
                 RandomForestClassifier(max_depth=10, max_samples=0.5,
                                        n_estimators=50, random_state=42))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
                '0',
                '0.5',
                '1',
                '2',
```

```
                '3',
                '4',
                '5',
                'some',
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
               [1,
                2,
                3,…
                                                 'job_req_communication',
                                                 'job_req_education',
                                                 'job_req_computer',
                                                 'job_req_organization',
                                                 'race', 'gender',
                                                 'college_degree', 'honors',
                                                 'worked_during_school',
                                                 'computer_skills',
                                                 'special_skills',
                                                 'volunteer', 'military',
                                                 'employment_holes',
                                                 'has_email_address'])])),
               ('randomforestclassifier',
                RandomForestClassifier(max_depth=1, max_samples=0.5,
                                       n_estimators=50, random_state=42))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
                '0',
                '0.5',
                '1',
                '2',
                '3',
                '4',
```

```
                '5',
                'some',
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
               [1,
                2,
                3,…
                                                  'job_req_communication',
                                                  'job_req_education',
                                                  'job_req_computer',
                                                  'job_req_organization',
                                                  'race', 'gender',
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                ('randomforestclassifier',
                 RandomForestClassifier(max_depth=10, max_samples=1.0,
                                        n_estimators=50, random_state=42))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
                '0',
                '0.5',
                '1',
                '2',
                '3',
                '4',
                '5',
                'some',
```

```
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
               [1,
                2,
                3,…
                                                'job_req_education',
                                                'job_req_computer',
                                                'job_req_organization',
                                                'race', 'gender',
                                                'college_degree', 'honors',
                                                'worked_during_school',
                                                'computer_skills',
                                                'special_skills',
                                                'volunteer', 'military',
                                                'employment_holes',
                                                'has_email_address'])])),
                 ('randomforestclassifier',
                  RandomForestClassifier(max_depth=20, max_samples=0.75,
                                         min_samples_split=5, n_estimators=200,
                                         random_state=42))])])
```

[ ]:
```
# KNN
# No class_weight
print("-----------------KNN------------------")
knn_clf = KNeighborsClassifier()
param_grid = {'kneighborsclassifier__n_neighbors': [1,5,10,50],
              'kneighborsclassifier__weights': ['uniform', 'distance'],
              'kneighborsclassifier__metric': ['minkowski', 'euclidean',␣
 ↪'manhattan'],
              'kneighborsclassifier__p': [1, 2],  # Only relevant for minkowski␣
 ↪metric
              'kneighborsclassifier__algorithm': ['auto', 'ball_tree',␣
 ↪'kd_tree', 'brute'],
              'kneighborsclassifier__leaf_size': [10, 20, 30, 40]}
print(MLpipe_KFold_Accuracy(X, y, ML_algo=knn_clf, param_grid=param_grid))
```

```
-----------------KNN------------------
```

```
Fitting 4 folds for each of 768 candidates, totalling 3072 fits

/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 768 candidates, totalling 3072 fits

/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 768 candidates, totalling 3072 fits

/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 768 candidates, totalling 3072 fits

/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 768 candidates, totalling 3072 fits

/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 768 candidates, totalling 3072 fits

/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 768 candidates, totalling 3072 fits

/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 768 candidates, totalling 3072 fits

/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 768 candidates, totalling 3072 fits

/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 768 candidates, totalling 3072 fits
Mean Test Accuracy: 0.9197125256673511
Standard Deviation of Test Accuracy: 0.00041067761806981015
```

```
([0.919917864476386, 0.919917864476386, 0.919917864476386, 0.919917864476386,
0.919917864476386, 0.9188911704312115, 0.919917864476386, 0.919917864476386,
0.9188911704312115, 0.919917864476386], [Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
                '0',
                '0.5',
                '1',
                '2',
                '3',
                '4',
                '5',
                'some',
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
               [1,
                2,
                3,…
                                                   'job_req_communication',
                                                   'job_req_education',
                                                   'job_req_computer',
                                                   'job_req_organization',
                                                   'race', 'gender',
                                                   'college_degree', 'honors',
                                                   'worked_during_school',
                                                   'computer_skills',
                                                   'special_skills',
                                                   'volunteer', 'military',
                                                   'employment_holes',
                                                   'has_email_address'])])),
                ('kneighborsclassifier',
                 KNeighborsClassifier(algorithm='ball_tree', leaf_size=20,
                                      n_neighbors=10, p=1))]),
```

```
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
                '0',
                '0.5',
                '1',
                '2',
                '3',
                '4',
                '5',
                'some',
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
               [1,
                2,
                3,…
                                                   'job_ownership',
                                                   'job_req_any',
                                                   'job_req_communication',
                                                   'job_req_education',
                                                   'job_req_computer',
                                                   'job_req_organization',
                                                   'race', 'gender',
                                                   'college_degree', 'honors',
                                                   'worked_during_school',
                                                   'computer_skills',
                                                   'special_skills',
                                                   'volunteer', 'military',
                                                   'employment_holes',
                                                   'has_email_address'])])),
                ('kneighborsclassifier',
                 KNeighborsClassifier(leaf_size=10, n_neighbors=50, p=1))]),
Pipeline(steps=[('columntransformer',
```

```
                ColumnTransformer(transformers=[('impute_ord',
                                                 Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                 ('ordinal',
OrdinalEncoder(categories=[['unknown',
               '0',
               '0.5',
               '1',
               '2',
               '3',
               '4',
               '5',
               'some',
               '6',
               '7',
               '8',
               '10'],
              ['none_listed',
               'high_school_grad',
               'some_college',
               'college'],
              [0,
               1,
               2,
               3,
               4],
              [1,
               2,
               3,…
                                                  'job_ownership',
                                                  'job_req_any',
                                                  'job_req_communication',
                                                  'job_req_education',
                                                  'job_req_computer',
                                                  'job_req_organization',
                                                  'race', 'gender',
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                ('kneighborsclassifier',
                 KNeighborsClassifier(leaf_size=10, n_neighbors=50, p=1))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
```

```
                                                Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                 ('ordinal',
OrdinalEncoder(categories=[['unknown',
               '0',
               '0.5',
               '1',
               '2',
               '3',
               '4',
               '5',
               'some',
               '6',
               '7',
               '8',
               '10'],
              ['none_listed',
               'high_school_grad',
               'some_college',
               'college'],
              [0,
               1,
               2,
               3,
               4],
              [1,
               2,
               3,…
                                                 'job_ownership',
                                                 'job_req_any',
                                                 'job_req_communication',
                                                 'job_req_education',
                                                 'job_req_computer',
                                                 'job_req_organization',
                                                 'race', 'gender',
                                                 'college_degree', 'honors',
                                                 'worked_during_school',
                                                 'computer_skills',
                                                 'special_skills',
                                                 'volunteer', 'military',
                                                 'employment_holes',
                                                 'has_email_address'])])),
                ('kneighborsclassifier',
                 KNeighborsClassifier(leaf_size=10, n_neighbors=10, p=1))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
```

```
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                ('ordinal',
OrdinalEncoder(categories=[['unknown',
                '0',
                '0.5',
                '1',
                '2',
                '3',
                '4',
                '5',
                'some',
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
               [1,
                2,
                3,…
                                                'job_ownership',
                                                'job_req_any',
                                                'job_req_communication',
                                                'job_req_education',
                                                'job_req_computer',
                                                'job_req_organization',
                                                'race', 'gender',
                                                'college_degree', 'honors',
                                                'worked_during_school',
                                                'computer_skills',
                                                'special_skills',
                                                'volunteer', 'military',
                                                'employment_holes',
                                                'has_email_address'])])),
                ('kneighborsclassifier',
                 KNeighborsClassifier(leaf_size=10, n_neighbors=50, p=1))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
```

```
 strategy='constant')),
                                                                ('ordinal',
OrdinalEncoder(categories=[['unknown',
                '0',
                '0.5',
                '1',
                '2',
                '3',
                '4',
                '5',
                'some',
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
               [1,
                2,
                3,…
                                                 'job_ownership',
                                                 'job_req_any',
                                                 'job_req_communication',
                                                 'job_req_education',
                                                 'job_req_computer',
                                                 'job_req_organization',
                                                 'race', 'gender',
                                                 'college_degree', 'honors',
                                                 'worked_during_school',
                                                 'computer_skills',
                                                 'special_skills',
                                                 'volunteer', 'military',
                                                 'employment_holes',
                                                 'has_email_address'])])),
                ('kneighborsclassifier',
                 KNeighborsClassifier(leaf_size=10, n_neighbors=10, p=1))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
```

```
                                                                ('ordinal',
OrdinalEncoder(categories=[['unknown',
                            '0',
                            '0.5',
                            '1',
                            '2',
                            '3',
                            '4',
                            '5',
                            'some',
                            '6',
                            '7',
                            '8',
                            '10'],
                           ['none_listed',
                            'high_school_grad',
                            'some_college',
                            'college'],
                           [0,
                            1,
                            2,
                            3,
                            4],
                           [1,
                            2,
                            3,…
                                                        'job_ownership',
                                                        'job_req_any',
                                                        'job_req_communication',
                                                        'job_req_education',
                                                        'job_req_computer',
                                                        'job_req_organization',
                                                        'race', 'gender',
                                                        'college_degree', 'honors',
                                                        'worked_during_school',
                                                        'computer_skills',
                                                        'special_skills',
                                                        'volunteer', 'military',
                                                        'employment_holes',
                                                        'has_email_address'])])),
                            ('kneighborsclassifier',
                             KNeighborsClassifier(leaf_size=10, n_neighbors=50, p=1))]),
Pipeline(steps=[('columntransformer',
                             ColumnTransformer(transformers=[('impute_ord',
                                                              Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                ('ordinal',
```

```
OrdinalEncoder(categories=[['unknown',
                '0',
                '0.5',
                '1',
                '2',
                '3',
                '4',
                '5',
                'some',
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
               [1,
                2,
                3,…
                                                    'job_ownership',
                                                    'job_req_any',
                                                    'job_req_communication',
                                                    'job_req_education',
                                                    'job_req_computer',
                                                    'job_req_organization',
                                                    'race', 'gender',
                                                    'college_degree', 'honors',
                                                    'worked_during_school',
                                                    'computer_skills',
                                                    'special_skills',
                                                    'volunteer', 'military',
                                                    'employment_holes',
                                                    'has_email_address'])])),
                ('kneighborsclassifier',
                 KNeighborsClassifier(leaf_size=10, n_neighbors=10))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
```

```
                '0',
                '0.5',
                '1',
                '2',
                '3',
                '4',
                '5',
                'some',
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
               [1,
                2,
                3,…
                                                  'job_ownership',
                                                  'job_req_any',
                                                  'job_req_communication',
                                                  'job_req_education',
                                                  'job_req_computer',
                                                  'job_req_organization',
                                                  'race', 'gender',
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                ('kneighborsclassifier',
                 KNeighborsClassifier(leaf_size=10, n_neighbors=10))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
                '0',
```

```
                 '0.5',
                 '1',
                 '2',
                 '3',
                 '4',
                 '5',
                 'some',
                 '6',
                 '7',
                 '8',
                 '10'],
                ['none_listed',
                 'high_school_grad',
                 'some_college',
                 'college'],
                [0,
                 1,
                 2,
                 3,
                 4],
                [1,
                 2,
                 3,…
                                                    'job_req_communication',
                                                    'job_req_education',
                                                    'job_req_computer',
                                                    'job_req_organization',
                                                    'race', 'gender',
                                                    'college_degree', 'honors',
                                                    'worked_during_school',
                                                    'computer_skills',
                                                    'special_skills',
                                                    'volunteer', 'military',
                                                    'employment_holes',
                                                    'has_email_address'])])),
                 ('kneighborsclassifier',
                  KNeighborsClassifier(algorithm='ball_tree', leaf_size=10,
                                       n_neighbors=10, p=1))])])
```

```
/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,
```

[ ]:

[ ]:
```
from xgboost import XGBClassifier
from collections import Counter
```

```
counter = Counter(y)
n_negative = counter[0]
n_positive = counter[1]

scale_pos_weight = n_negative / n_positive
print("Suggested scale_pos_weight:", scale_pos_weight)
# Should I use y_train or y to find the scale_pos_weight?

# Define the parameter grid for XGBoost
param_grid_xgb = {
    'xgbclassifier__n_estimators': [50, 100, 200],
    'xgbclassifier__learning_rate': [0.01, 0.05, 0.1],
    'xgbclassifier__max_depth': [3, 5, 10],
    'xgbclassifier__colsample_bytree': [0.75, 0.9, 1],
    'xgbclassifier__subsample': [0.75, 0.9, 1],
    'xgbclassifier__min_child_weight': [],
    'xgbclassifier__reg_alpha': [0, 0.01, 0.1, 1],
    'xgbclassifier__reg_lambda': [1, 1.5, 2],
    'xgbclassifier__scale_pos_weight': [scale_pos_weight * 0.5,␣
 ↪scale_pos_weight, scale_pos_weight * 1.5]
}

# Initialize the XGBoost classifier
xgb_clf = XGBClassifier(eval_metric='auc', random_state=42)

# Run the pipeline with XGBoost
print("----------------XGBoost-----------------")
xgb_test_scores, xgb_best_models = MLpipe_KFold_Accuracy(
    X, y, ML_algo=xgb_clf, param_grid=param_grid_xgb
)

# Display results
print(f"XGBoost Test Scores: {xgb_test_scores}")
print(f"Mean Test Accuracy: {np.mean(xgb_test_scores)}")
print(f"Standard Deviation of Test Accuracy: {np.std(xgb_test_scores)}")
```

```
----------------XGBoost-----------------
Fitting 4 folds for each of 3888 candidates, totalling 15552 fits

/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 3888 candidates, totalling 15552 fits
Fitting 4 folds for each of 3888 candidates, totalling 15552 fits
Fitting 4 folds for each of 3888 candidates, totalling 15552 fits
Fitting 4 folds for each of 3888 candidates, totalling 15552 fits
Fitting 4 folds for each of 3888 candidates, totalling 15552 fits
```

```
/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 3888 candidates, totalling 15552 fits
Fitting 4 folds for each of 3888 candidates, totalling 15552 fits

/opt/anaconda3/envs/data1030/lib/python3.12/site-packages/numpy/ma/core.py:2820:
RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,

Fitting 4 folds for each of 3888 candidates, totalling 15552 fits
Fitting 4 folds for each of 3888 candidates, totalling 15552 fits
Mean Test Accuracy: 0.9196098562628336
Standard Deviation of Test Accuracy: 0.0016586749919305253
XGBoost Test Scores: [0.9188911704312115, 0.917864476386037, 0.919917864476386,
0.919917864476386, 0.9168377823408624, 0.9209445585215605, 0.9219712525667351,
0.9219712525667351, 0.917864476386037, 0.919917864476386]
Mean Test Accuracy: 0.9196098562628336
Standard Deviation of Test Accuracy: 0.0016586749919305253
```

[19]:
```
print(xgb_best_models)
```

```
[Pipeline(steps=[('columntransformer',
                  ColumnTransformer(transformers=[('impute_ord',
                                                   Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                   ('ordinal',
OrdinalEncoder(categories=[['unknown',
                            '0',
                            '0.5',
                            '1',
                            '2',
                            '3',
                            '4',
                            '5',
                            'some',
                            '6',
                            '7',
                            '8',
                            '10'],
                           ['none_listed',
                            'high_school_grad',
                            'some_college',
                            'college'],
                           [0,
                            1,
                            2,
                            3,
```

```
                4],
               [1,
                2,
                3,…
                                feature_types=None, gamma=None, grow_policy=None,
                                importance_type=None,
                                interaction_constraints=None, learning_rate=0.05,
                                max_bin=None, max_cat_threshold=None,
                                max_cat_to_onehot=None, max_delta_step=None,
                                max_depth=10, max_leaves=None,
                                min_child_weight=None, missing=nan,
                                monotone_constraints=None, multi_strategy=None,
                                n_estimators=50, n_jobs=None,
                                num_parallel_tree=None, random_state=42, …))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
                '0',
                '0.5',
                '1',
                '2',
                '3',
                '4',
                '5',
                'some',
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
               [1,
                2,
                3,…
                                feature_types=None, gamma=None, grow_policy=None,
                                importance_type=None,
                                interaction_constraints=None, learning_rate=0.05,
```

```
                              max_bin=None, max_cat_threshold=None,
                              max_cat_to_onehot=None, max_delta_step=None,
                              max_depth=5, max_leaves=None,
                              min_child_weight=None, missing=nan,
                              monotone_constraints=None, multi_strategy=None,
                              n_estimators=200, n_jobs=None,
                              num_parallel_tree=None, random_state=42, …))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
                '0',
                '0.5',
                '1',
                '2',
                '3',
                '4',
                '5',
                'some',
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
               [1,
                2,
                3,…
                              feature_types=None, gamma=None, grow_policy=None,
                              importance_type=None,
                              interaction_constraints=None, learning_rate=0.2,
                              max_bin=None, max_cat_threshold=None,
                              max_cat_to_onehot=None, max_delta_step=None,
                              max_depth=3, max_leaves=None,
                              min_child_weight=None, missing=nan,
                              monotone_constraints=None, multi_strategy=None,
                              n_estimators=50, n_jobs=None,
                              num_parallel_tree=None, random_state=42, …))]),
```

```
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
               '0',
               '0.5',
               '1',
               '2',
               '3',
               '4',
               '5',
               'some',
               '6',
               '7',
               '8',
               '10'],
              ['none_listed',
               'high_school_grad',
               'some_college',
               'college'],
              [0,
               1,
               2,
               3,
               4],
              [1,
               2,
               3,…
                              feature_types=None, gamma=None, grow_policy=None,
                              importance_type=None,
                              interaction_constraints=None, learning_rate=0.05,
                              max_bin=None, max_cat_threshold=None,
                              max_cat_to_onehot=None, max_delta_step=None,
                              max_depth=3, max_leaves=None,
                              min_child_weight=None, missing=nan,
                              monotone_constraints=None, multi_strategy=None,
                              n_estimators=100, n_jobs=None,
                              num_parallel_tree=None, random_state=42, …))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
```

```
                '0',
                '0.5',
                '1',
                '2',
                '3',
                '4',
                '5',
                'some',
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
               [1,
                2,
                3,…
                               feature_types=None, gamma=None, grow_policy=None,
                               importance_type=None,
                               interaction_constraints=None, learning_rate=0.2,
                               max_bin=None, max_cat_threshold=None,
                               max_cat_to_onehot=None, max_delta_step=None,
                               max_depth=5, max_leaves=None,
                               min_child_weight=None, missing=nan,
                               monotone_constraints=None, multi_strategy=None,
                               n_estimators=50, n_jobs=None,
                               num_parallel_tree=None, random_state=42, …))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
                '0',
                '0.5',
                '1',
                '2',
                '3',
                '4',
                '5',
```

```
                'some',
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
               [1,
                2,
                3,…
                               feature_types=None, gamma=None, grow_policy=None,
                               importance_type=None,
                               interaction_constraints=None, learning_rate=0.2,
                               max_bin=None, max_cat_threshold=None,
                               max_cat_to_onehot=None, max_delta_step=None,
                               max_depth=3, max_leaves=None,
                               min_child_weight=None, missing=nan,
                               monotone_constraints=None, multi_strategy=None,
                               n_estimators=100, n_jobs=None,
                               num_parallel_tree=None, random_state=42, …))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
                '0',
                '0.5',
                '1',
                '2',
                '3',
                '4',
                '5',
                'some',
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
```

```
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
               [1,
                2,
                3,…
                               feature_types=None, gamma=None, grow_policy=None,
                               importance_type=None,
                               interaction_constraints=None, learning_rate=0.1,
                               max_bin=None, max_cat_threshold=None,
                               max_cat_to_onehot=None, max_delta_step=None,
                               max_depth=5, max_leaves=None,
                               min_child_weight=None, missing=nan,
                               monotone_constraints=None, multi_strategy=None,
                               n_estimators=50, n_jobs=None,
                               num_parallel_tree=None, random_state=42, …))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
                '0',
                '0.5',
                '1',
                '2',
                '3',
                '4',
                '5',
                'some',
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
```

```
              [1,
               2,
               3,…
                                feature_types=None, gamma=None, grow_policy=None,
                                importance_type=None,
                                interaction_constraints=None, learning_rate=0.2,
                                max_bin=None, max_cat_threshold=None,
                                max_cat_to_onehot=None, max_delta_step=None,
                                max_depth=3, max_leaves=None,
                                min_child_weight=None, missing=nan,
                                monotone_constraints=None, multi_strategy=None,
                                n_estimators=100, n_jobs=None,
                                num_parallel_tree=None, random_state=42, …))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
               '0',
               '0.5',
               '1',
               '2',
               '3',
               '4',
               '5',
               'some',
               '6',
               '7',
               '8',
               '10'],
              ['none_listed',
               'high_school_grad',
               'some_college',
               'college'],
              [0,
               1,
               2,
               3,
               4],
              [1,
               2,
               3,…
                                feature_types=None, gamma=None, grow_policy=None,
                                importance_type=None,
                                interaction_constraints=None, learning_rate=0.2,
                                max_bin=None, max_cat_threshold=None,
```

```
                              max_cat_to_onehot=None, max_delta_step=None,
                              max_depth=5, max_leaves=None,
                              min_child_weight=None, missing=nan,
                              monotone_constraints=None, multi_strategy=None,
                              n_estimators=50, n_jobs=None,
                              num_parallel_tree=None, random_state=42, …))]),
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(transformers=[('impute_ord',
                                                  Pipeline(steps=[('imputer',
SimpleImputer(fill_value='unknown',
 strategy='constant')),
                                                                  ('ordinal',
OrdinalEncoder(categories=[['unknown',
                '0',
                '0.5',
                '1',
                '2',
                '3',
                '4',
                '5',
                'some',
                '6',
                '7',
                '8',
                '10'],
               ['none_listed',
                'high_school_grad',
                'some_college',
                'college'],
               [0,
                1,
                2,
                3,
                4],
               [1,
                2,
                3,…
                              feature_types=None, gamma=None, grow_policy=None,
                              importance_type=None,
                              interaction_constraints=None, learning_rate=0.05,
                              max_bin=None, max_cat_threshold=None,
                              max_cat_to_onehot=None, max_delta_step=None,
                              max_depth=5, max_leaves=None,
                              min_child_weight=None, missing=nan,
                              monotone_constraints=None, multi_strategy=None,
                              n_estimators=200, n_jobs=None,
                              num_parallel_tree=None, random_state=42, …))])]
```

0.0.1 Maybe logistic is the best?

[ ]: