DATA1030Project

December 15, 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Warning: Looks like you're using an outdated `kagglehub` version, please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
```

[49]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv("/Users/fruit/Desktop/DATA1030-Fall2024/
 ↪Key-Resume-Attributes-Impacting-Job-Callbacks/data/resume.csv")
```

[10]:
```
#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
```

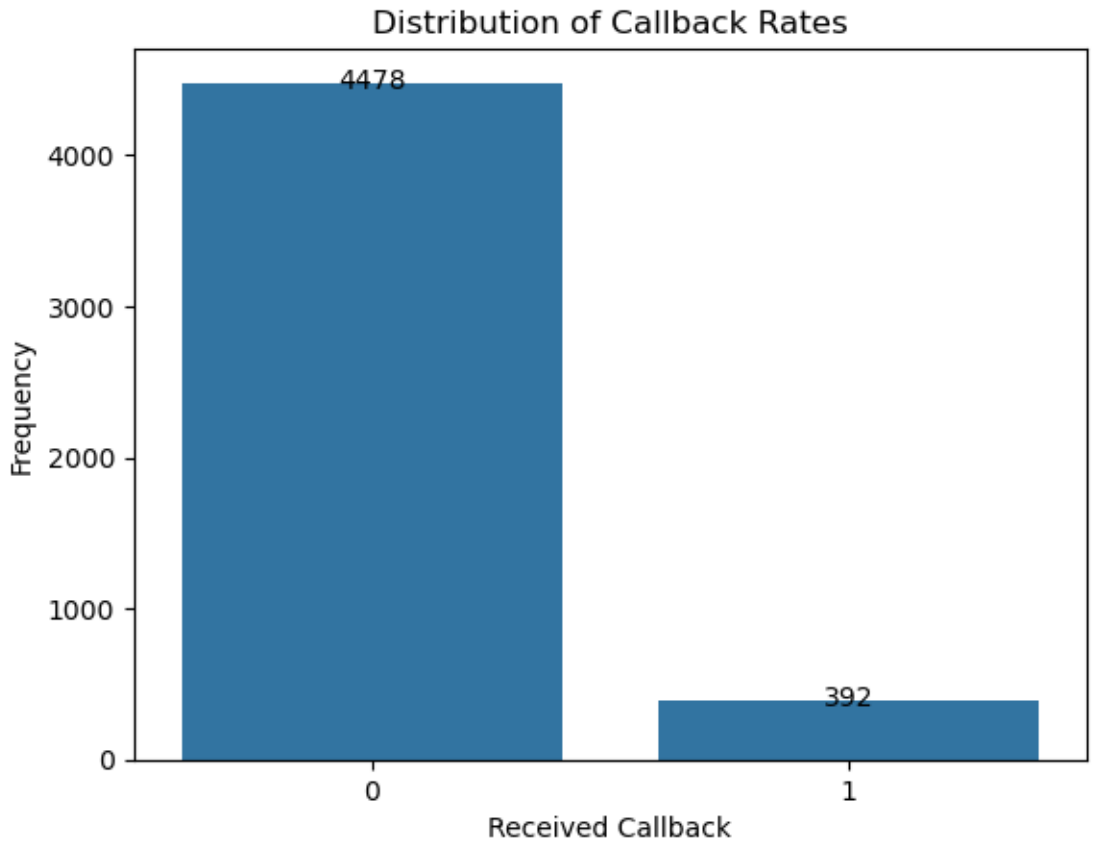
[9]:
```
df.columns
```

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

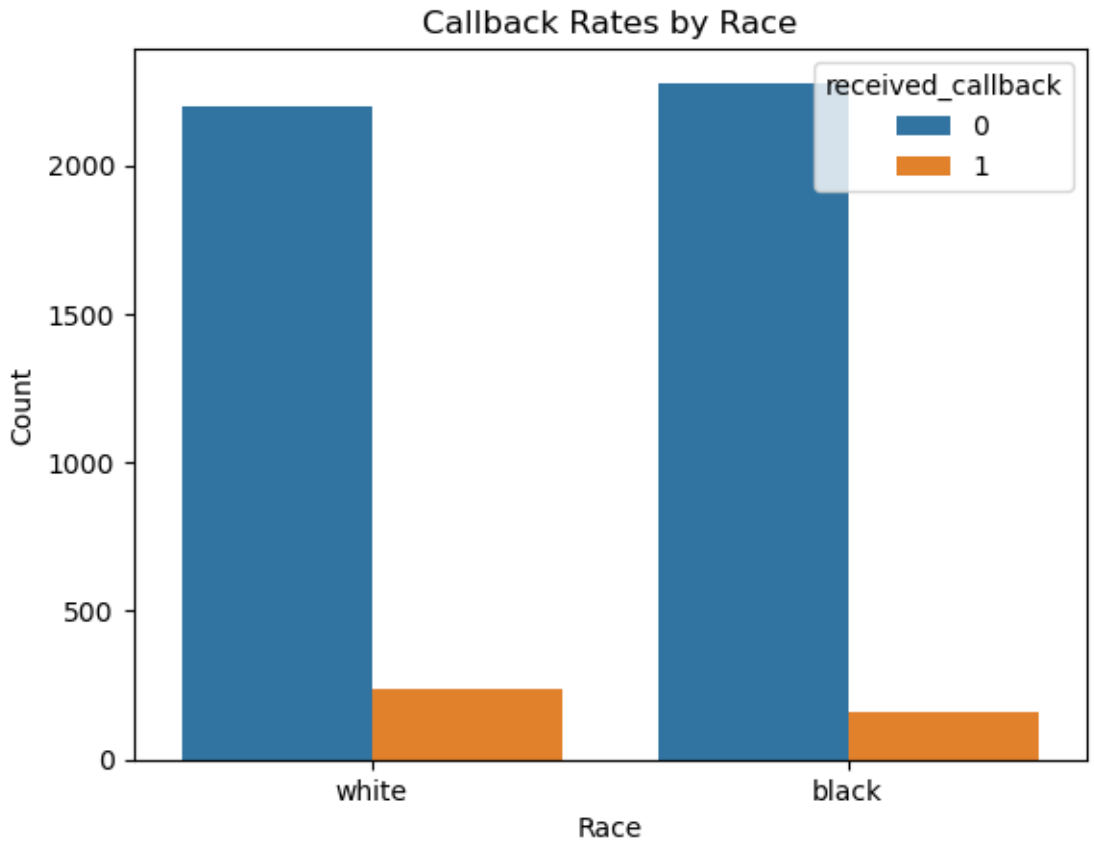
```
       'job_req_any', 'job_req_communication', 'job_req_education',
       'job_req_min_experience', 'job_req_computer', 'job_req_organization',
       'job_req_school', 'received_callback', 'firstname', 'race', 'gender',
       'years_college', 'college_degree', 'honors', 'worked_during_school',
       '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()
```

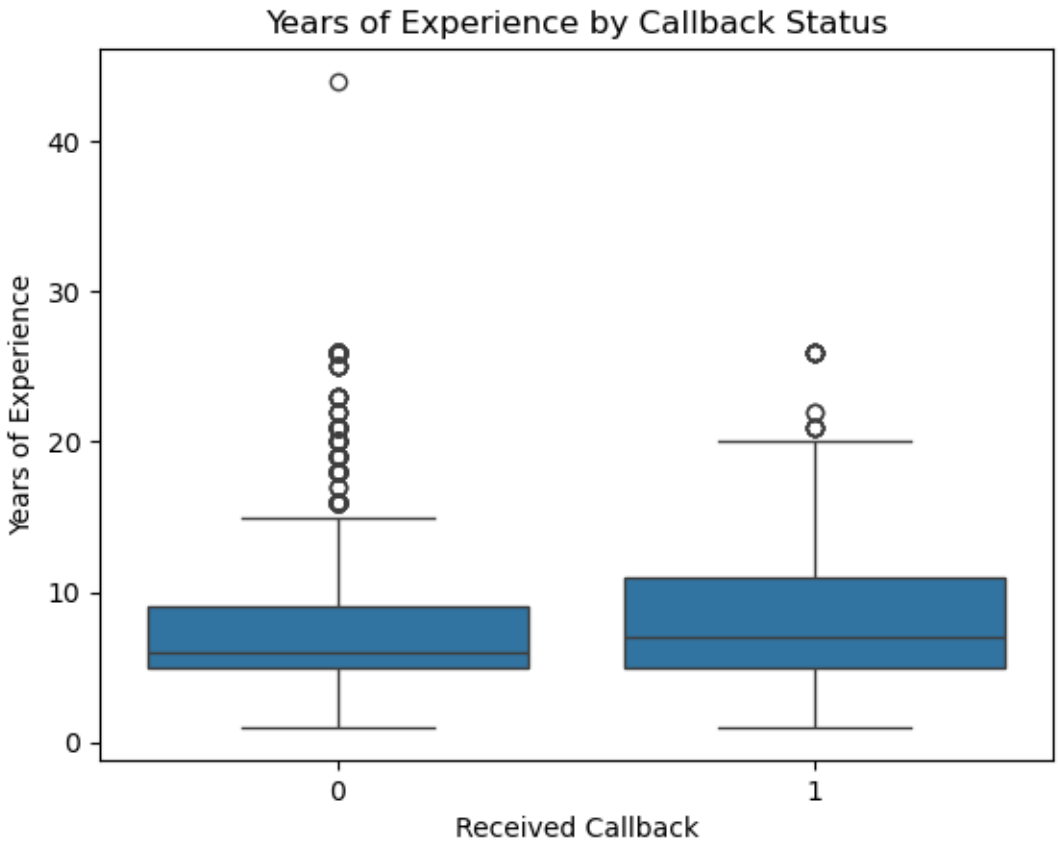
[50]:
```
# Plot a histogram of callback rates

# Plot the count plot
ax = sns.countplot(x=df["received_callback"])
plt.xlabel("Received Callback")
plt.ylabel("Frequency")
plt.title("Distribution of Callback Rates")

# Add counts above the bars
for p in ax.patches:
    ax.text(
        p.get_x() + p.get_width() / 2.,  # x-coordinate of the bar's center
        p.get_height() + 10,  # y-coordinate just above the bar
        int(p.get_height()),  # the count as an integer
        ha="center",  # horizontal alignment
        va="center",  # vertical alignment
        fontsize=10  # font size
    )

plt.savefig(f"/Users/fruit/Desktop/DATA1030-Fall2024/
 ↪Key-Resume-Attributes-Impacting-Job-Callbacks/figures/
 ↪target_variable_distribution.png", dpi=800)
plt.show()
```



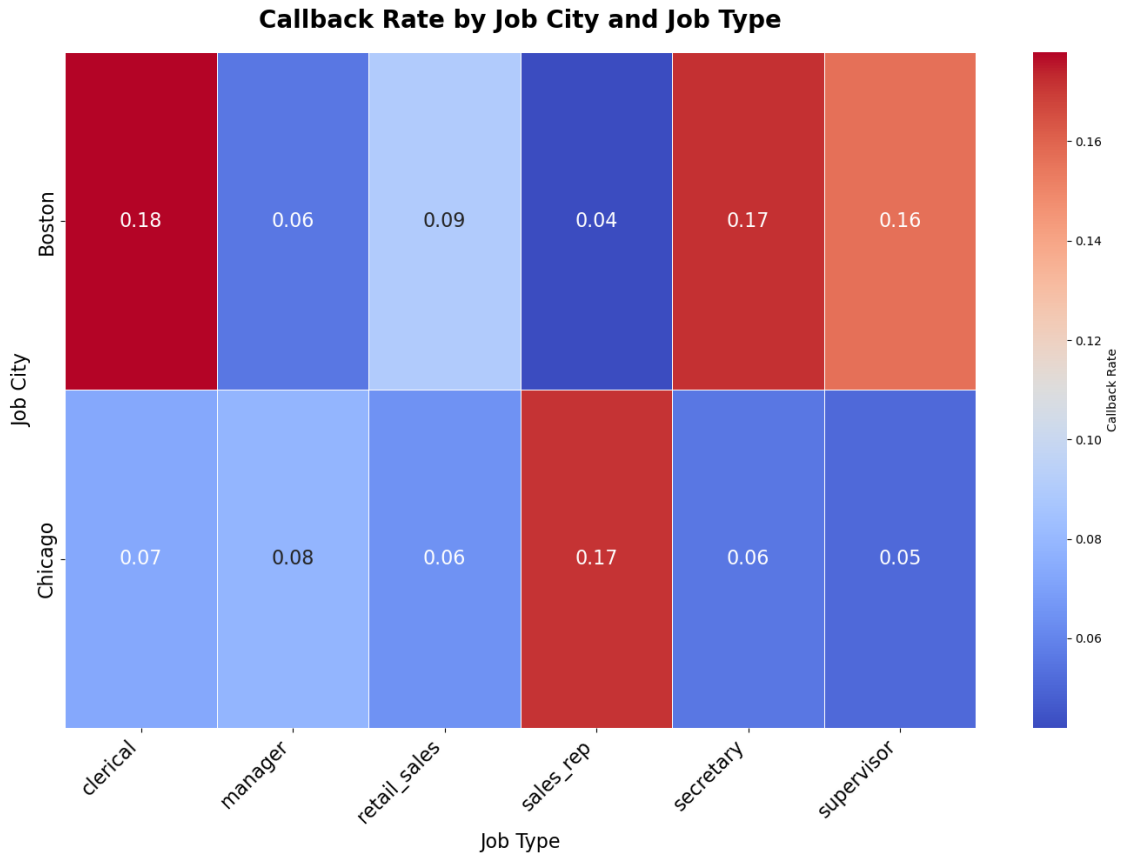
[12]:
```
sns.countplot(x="race", hue="received_callback", data=df)
plt.xlabel("Race")
plt.ylabel("Count")
plt.title("Callback Rates by Race")
plt.show()
```



[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()
```



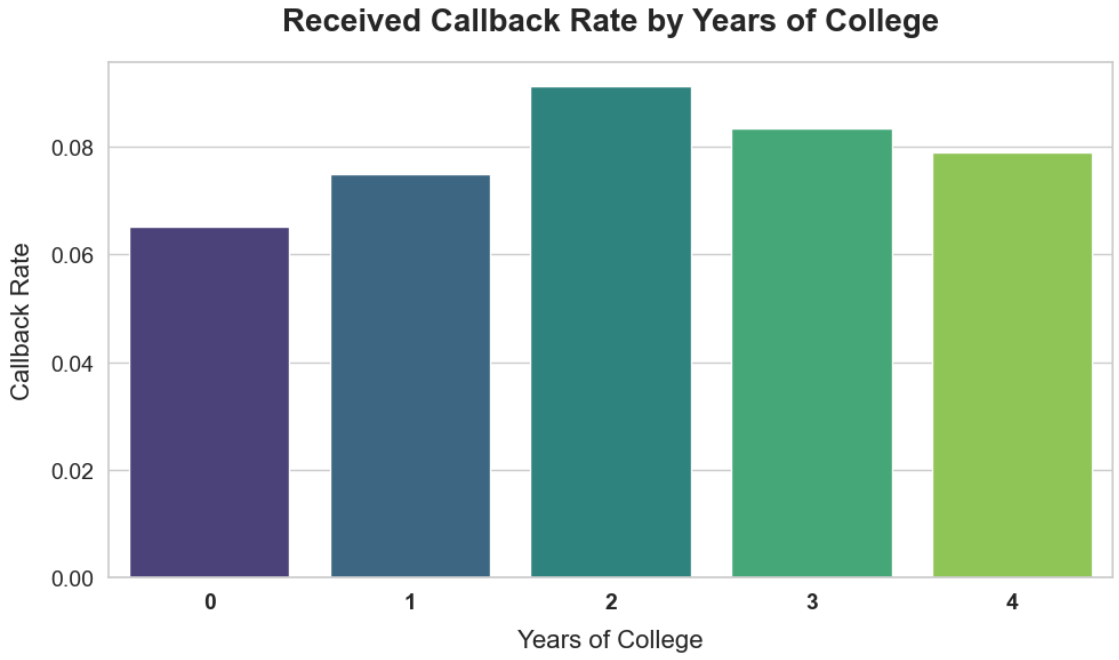
[51]:
```
heatmap_data = df.pivot_table(index='job_city', columns='job_type',␣
 ↪values='received_callback', aggfunc='mean')

# 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.savefig(f"/Users/fruit/Desktop/DATA1030-Fall2024/
 ↪Key-Resume-Attributes-Impacting-Job-Callbacks/figures/
 ↪callback_rate_by_job_city_and_type.png", dpi=800)

plt.show()
```



[52]:
```
# 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.savefig(f"/Users/fruit/Desktop/DATA1030-Fall2024/
  ↪Key-Resume-Attributes-Impacting-Job-Callbacks/figures/
  ↪callback_rate_by_years_of_college.png", dpi=800)
plt.show()
```

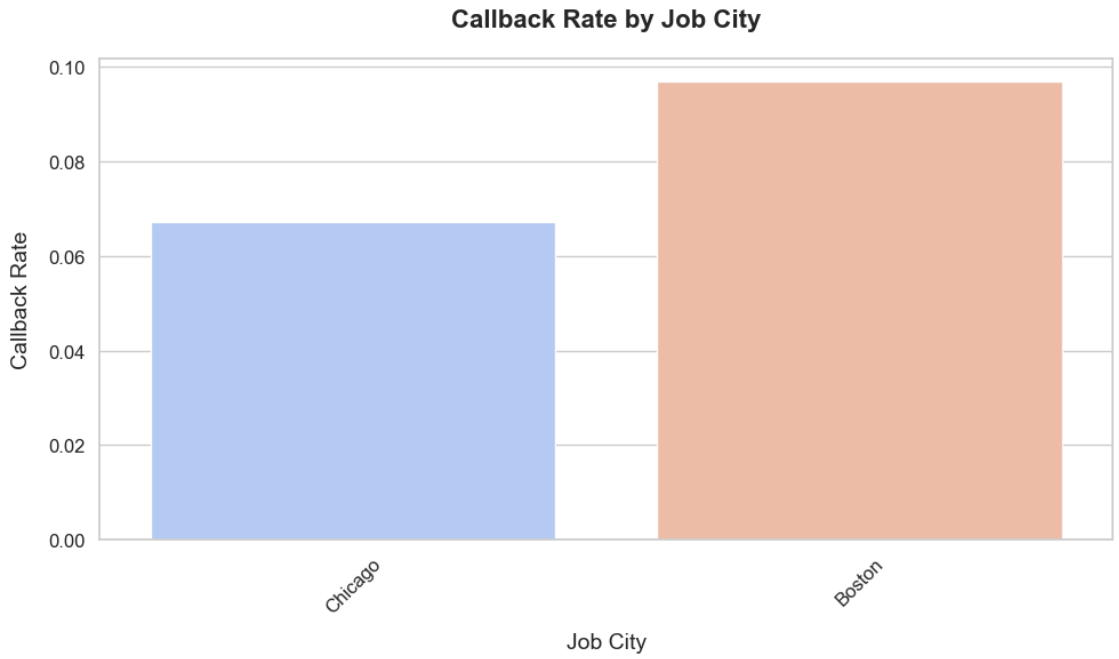
```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36825/1379185811.py:6
: FutureWarning:

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_36825/1379185811.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')
```



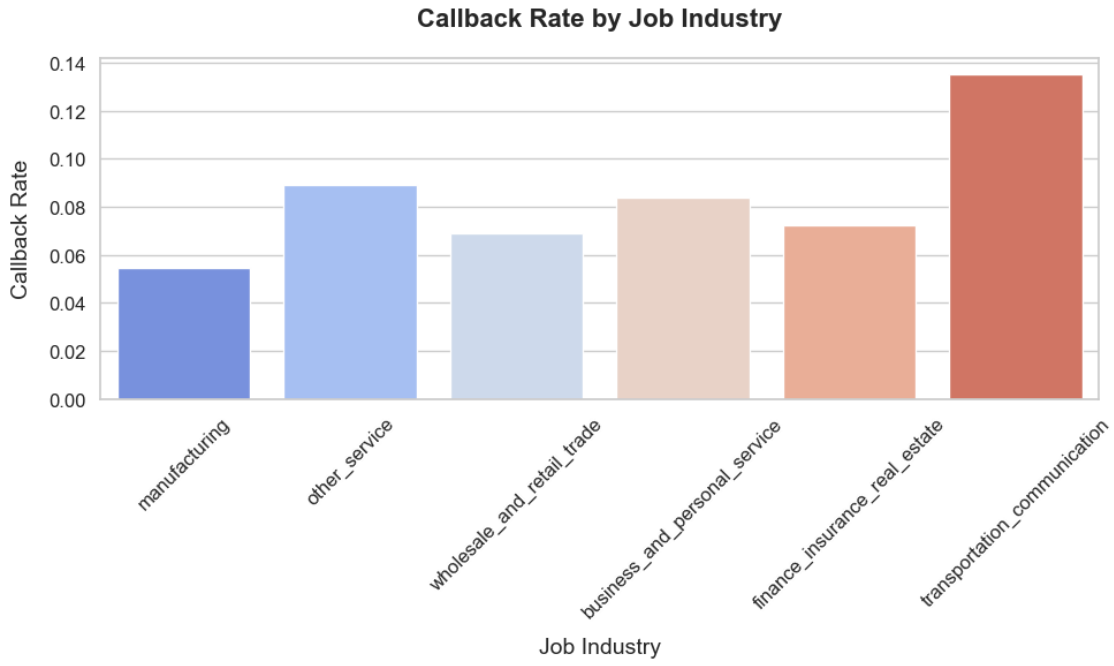
[16]:
```
# List of categorical features (excluding the target 'received_callback')
categorical_features = ['job_city', 'job_industry', 'job_type',␣
  ↪'job_fed_contractor', 'job_equal_opp_employer',
```

```
                        '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()
```

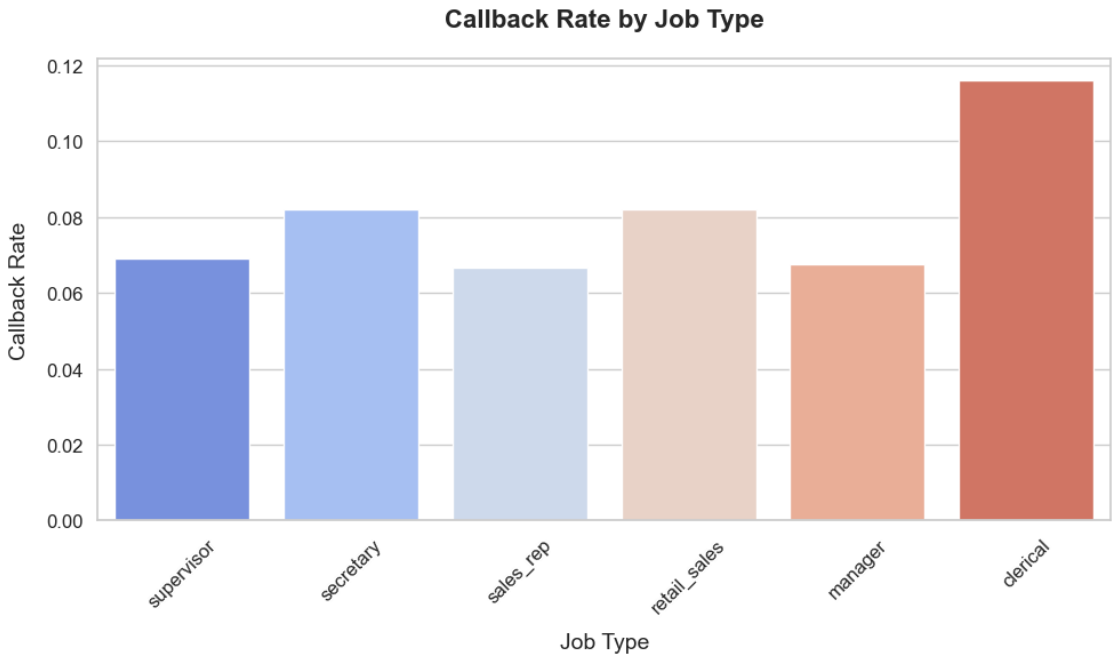
```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```



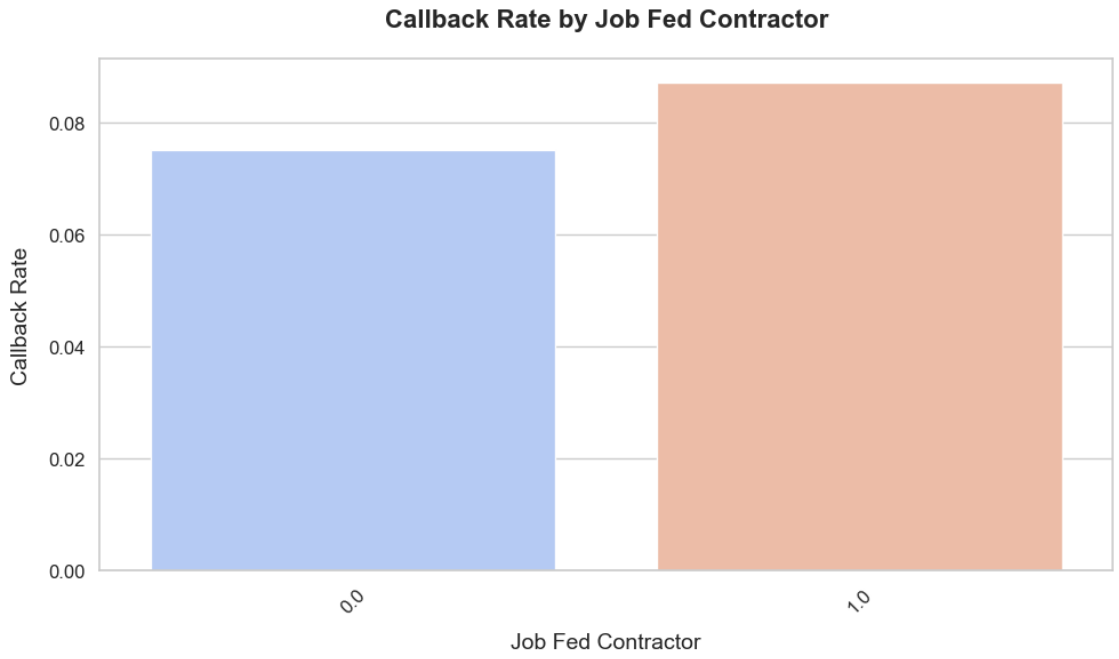

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```



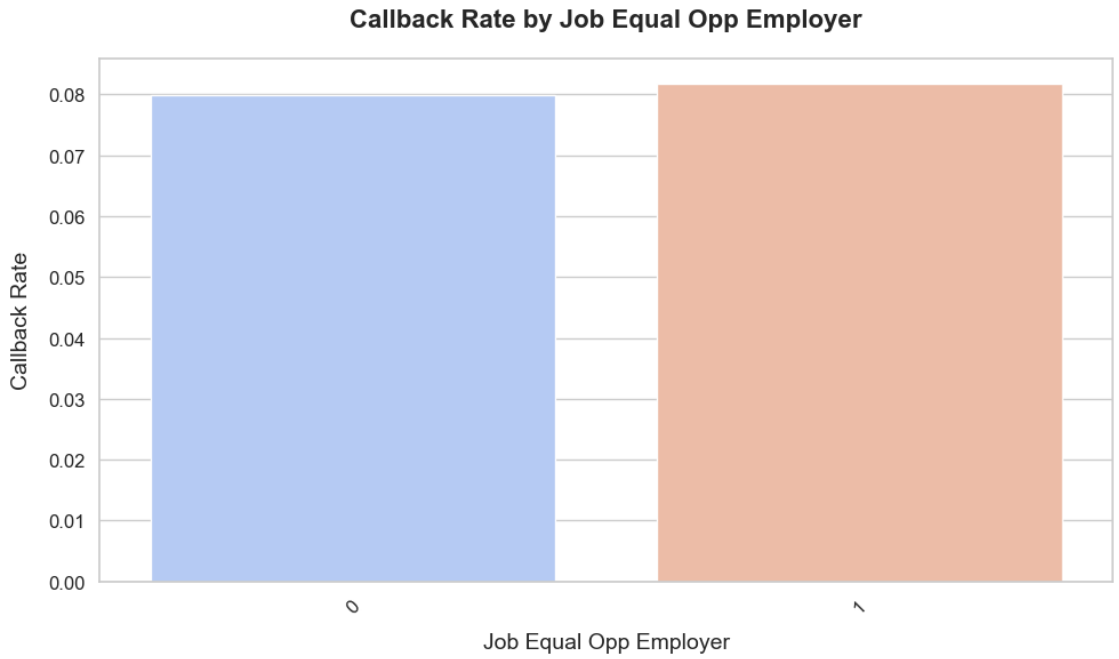

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```



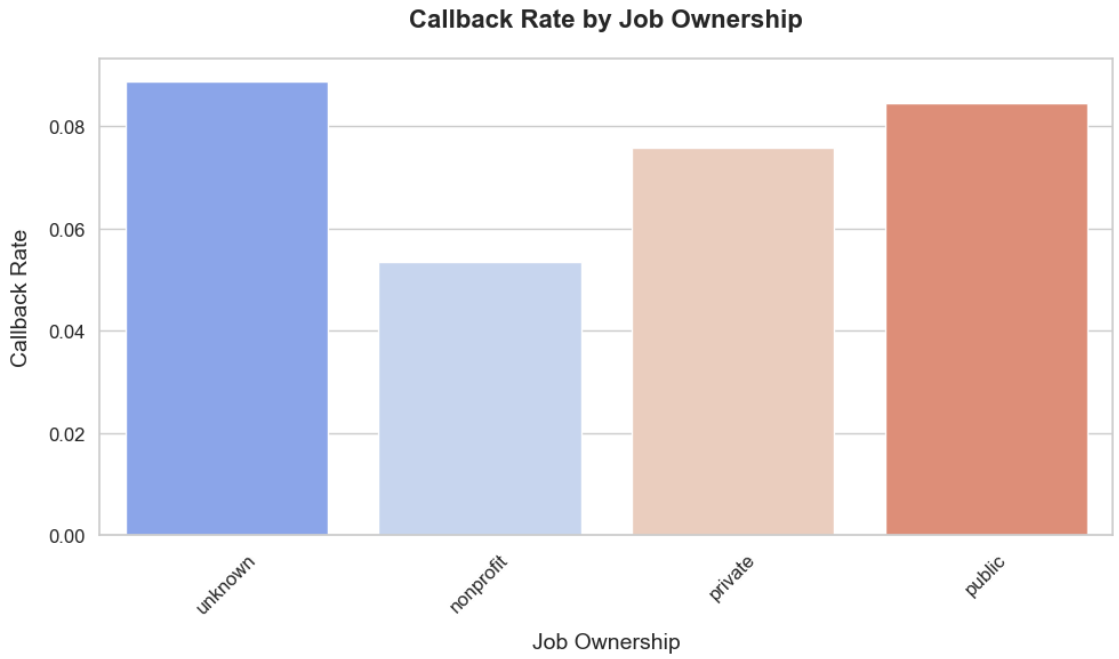

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```



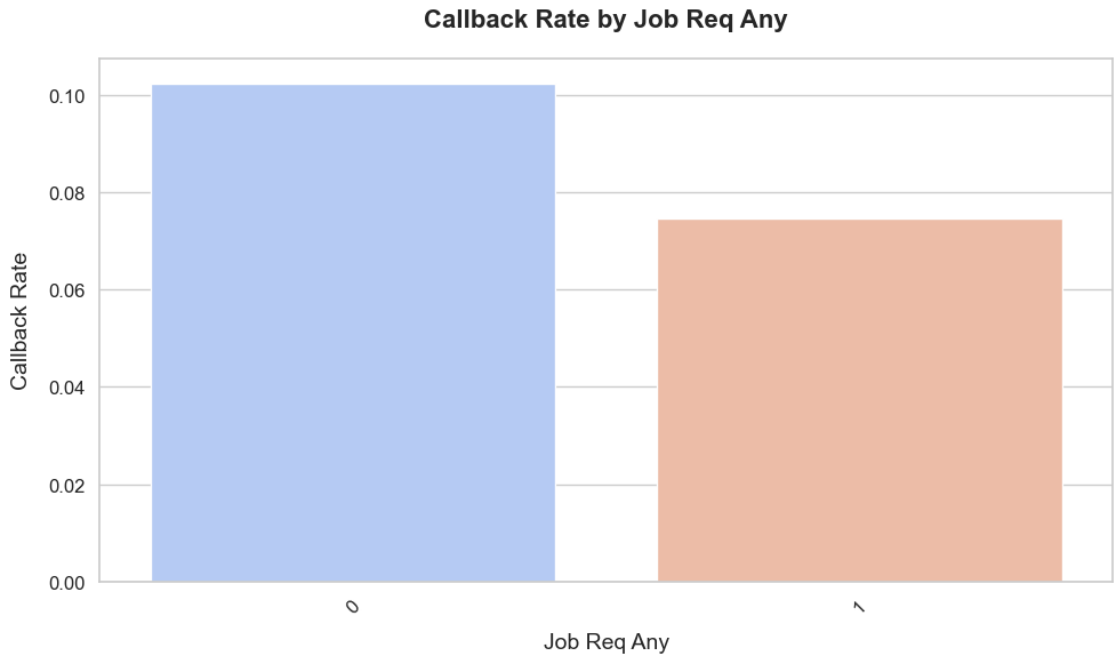

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```



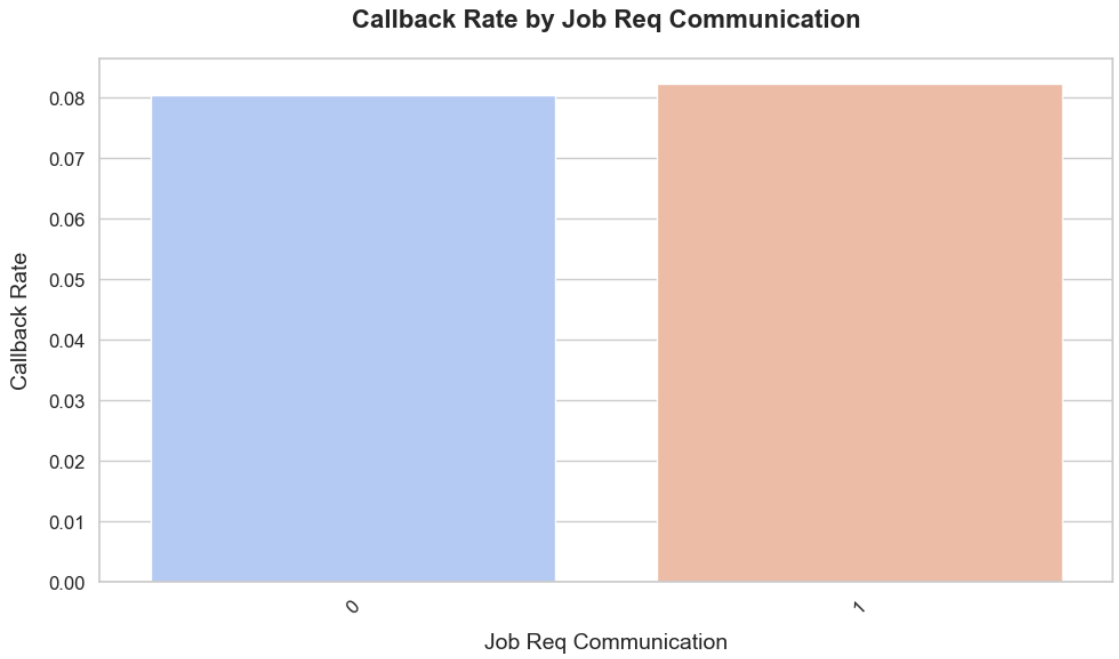

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```



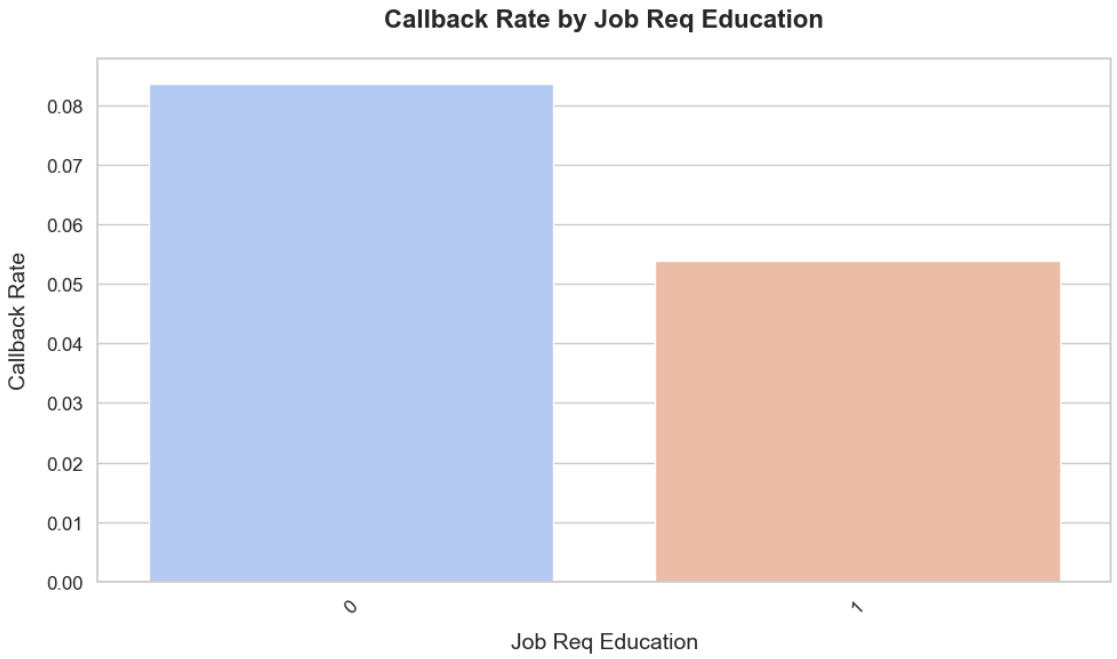

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```



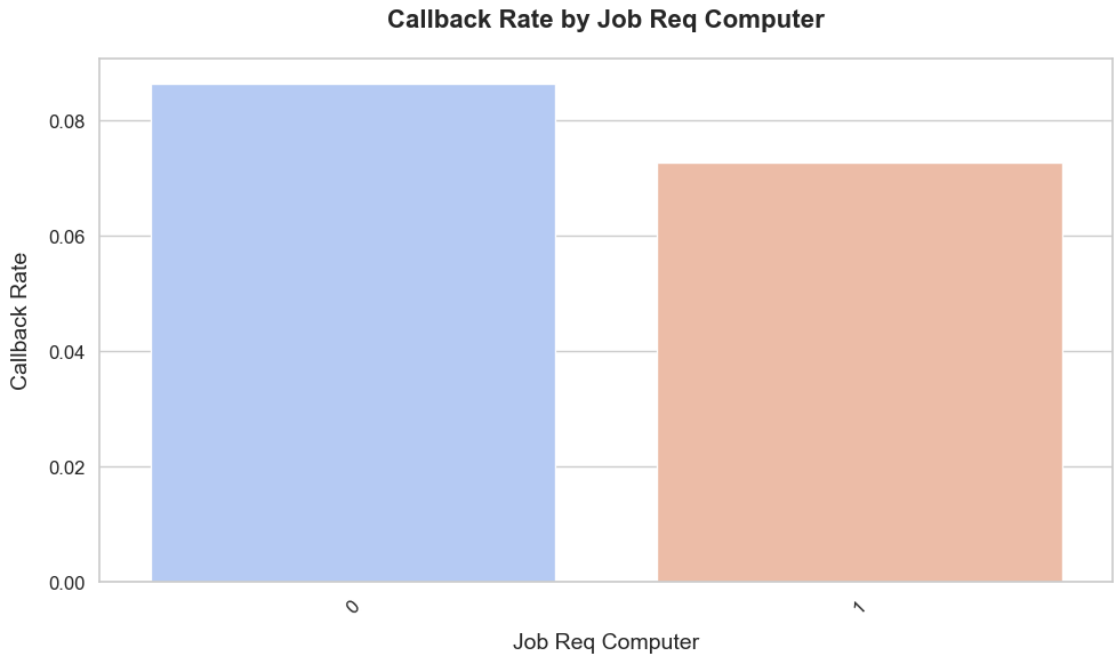

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```



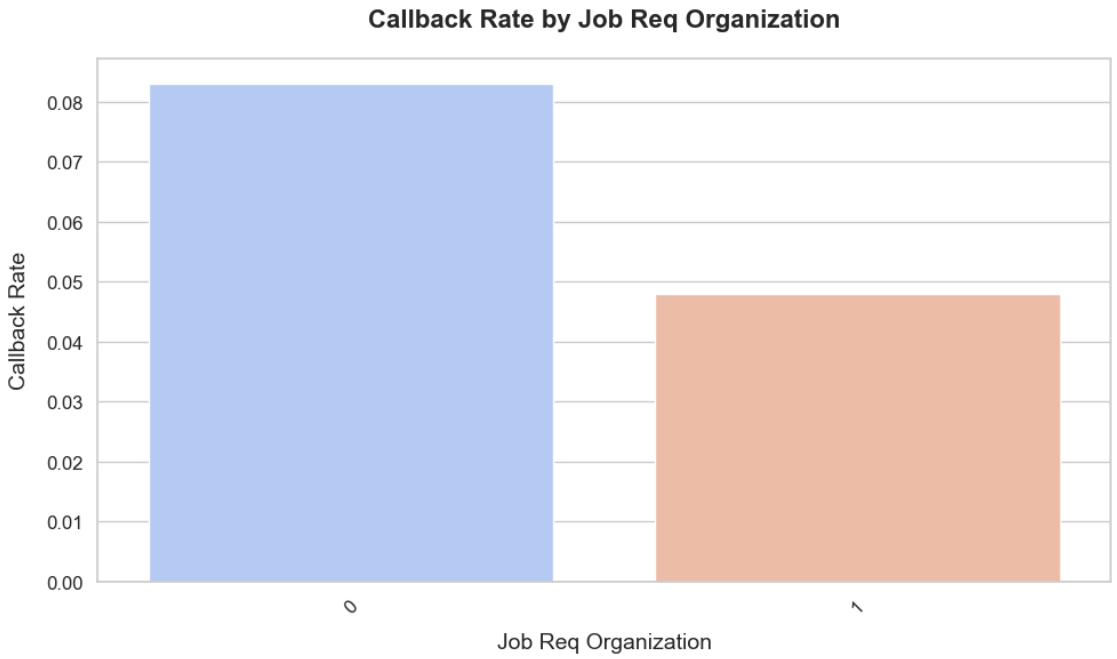

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```



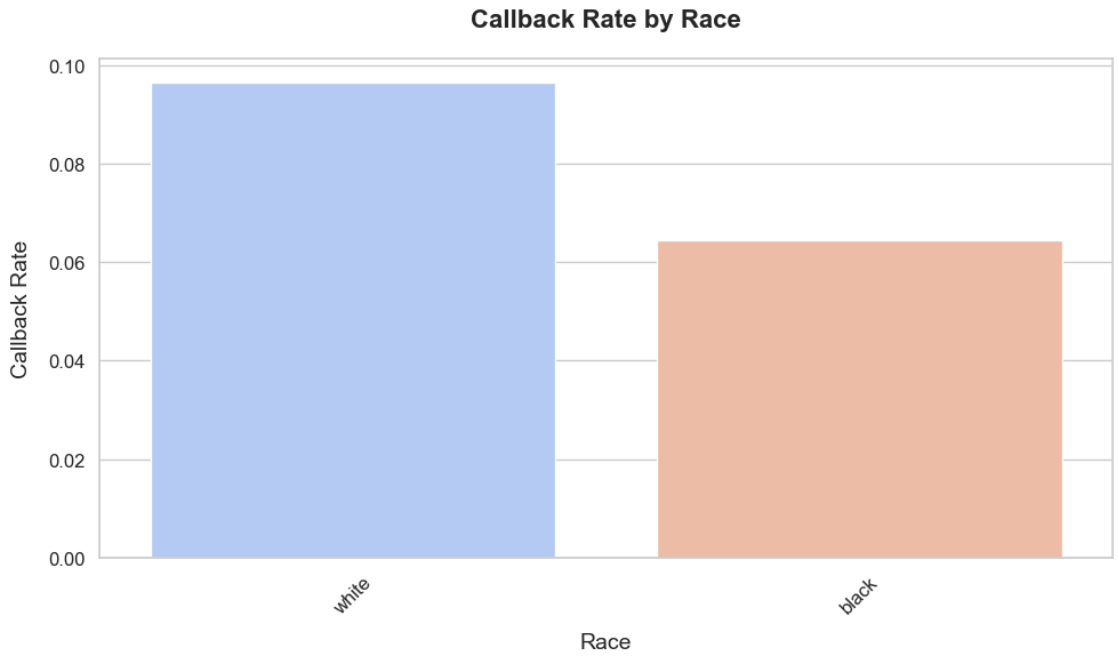

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```



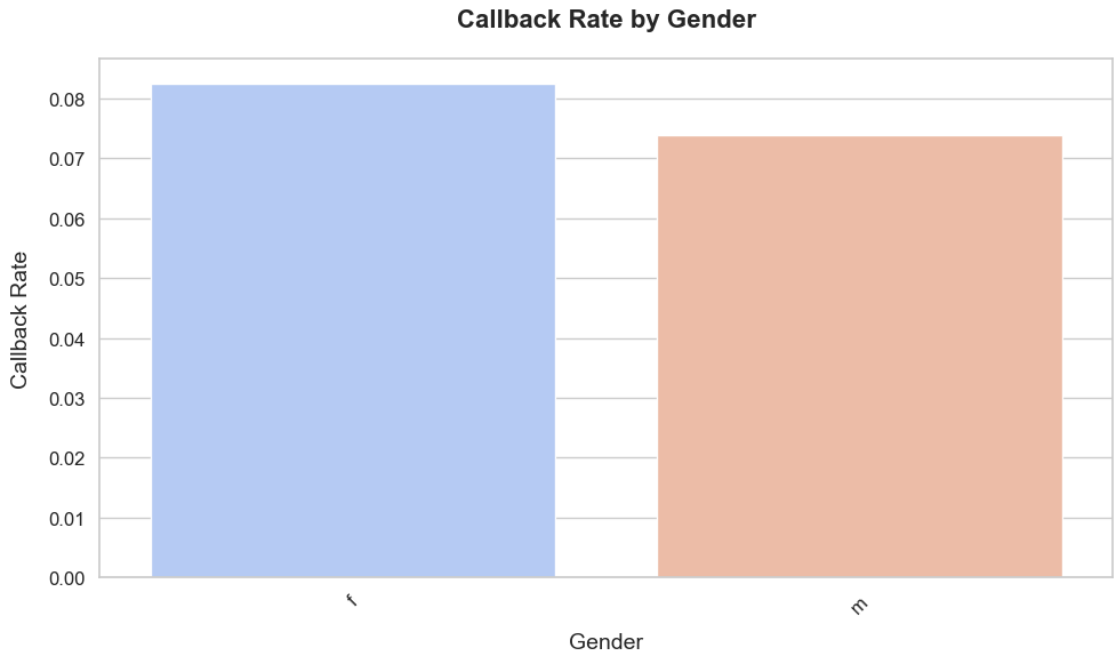

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```



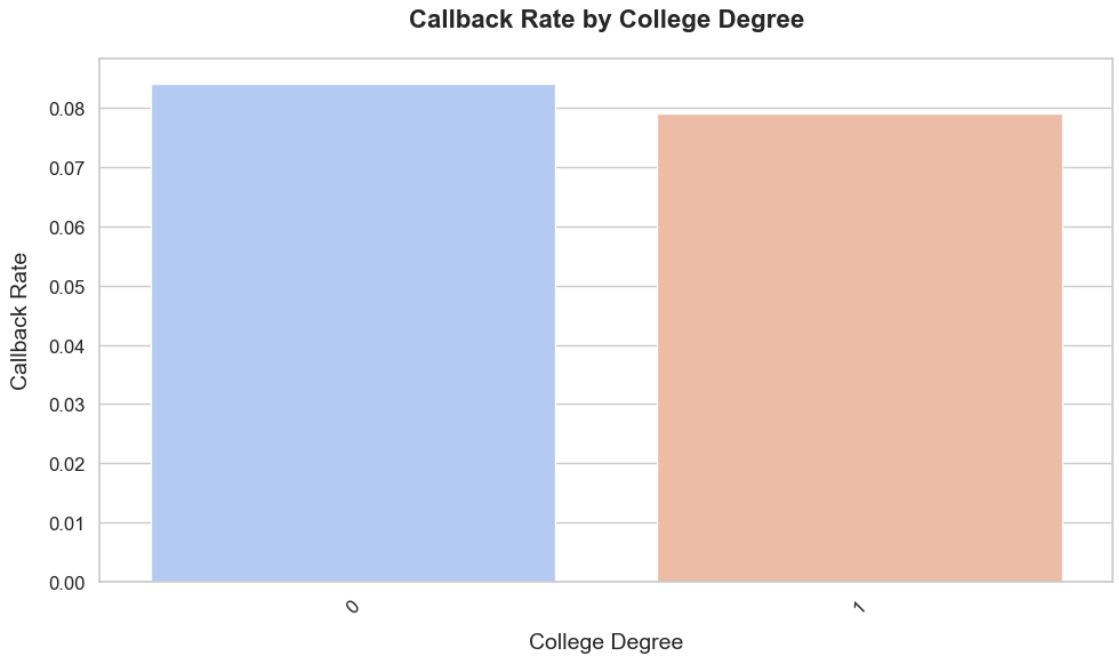

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```



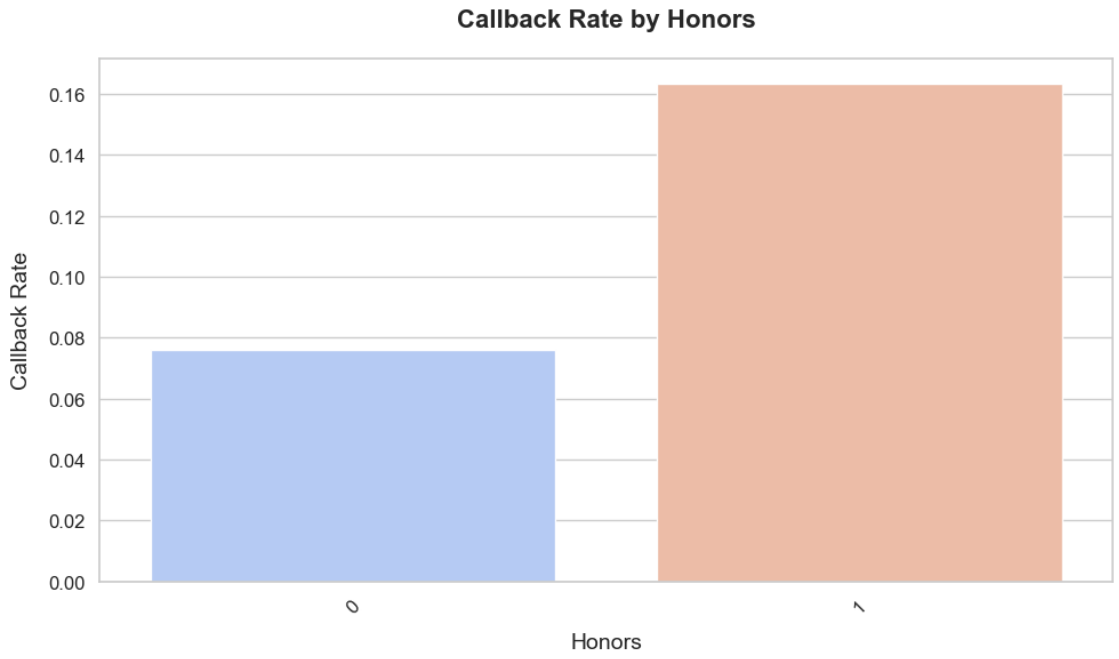

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```



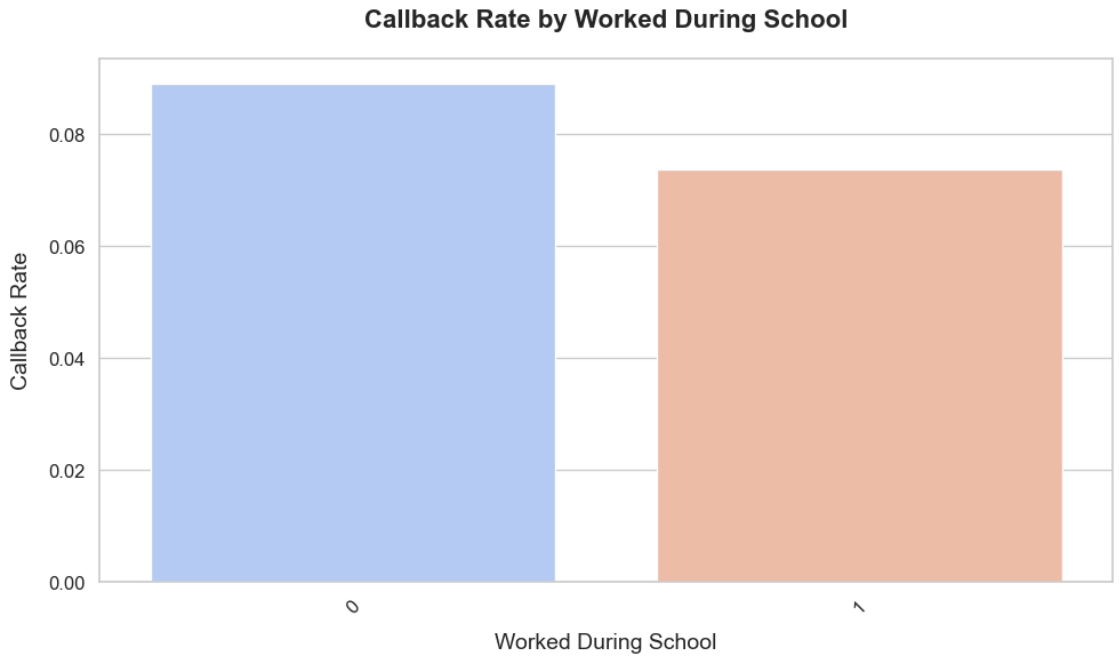

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```



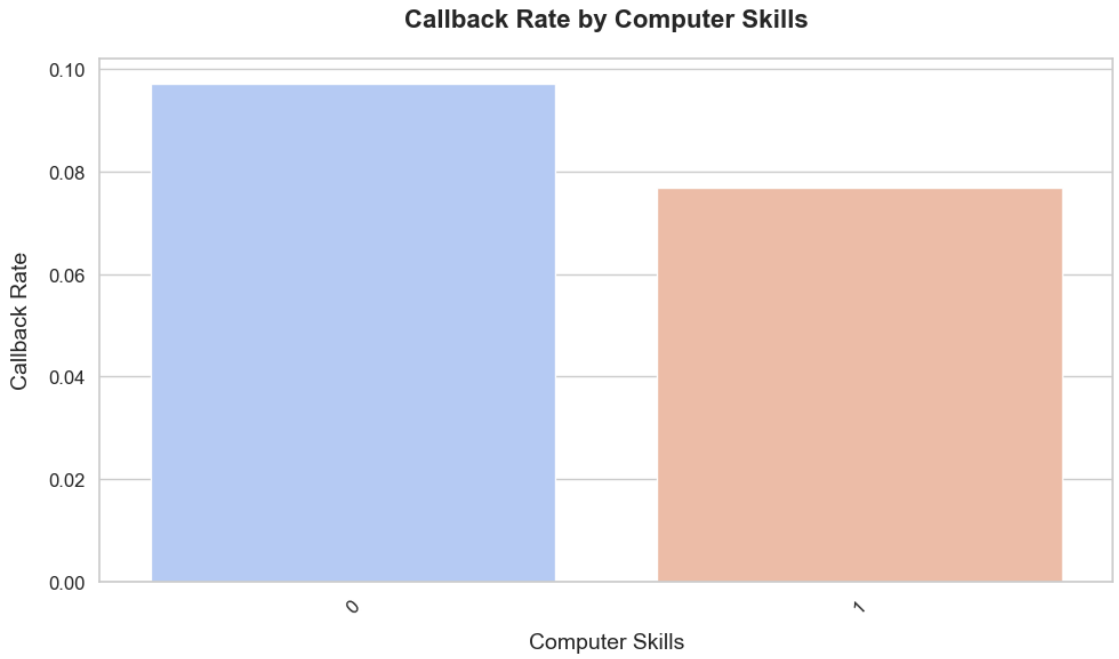

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```



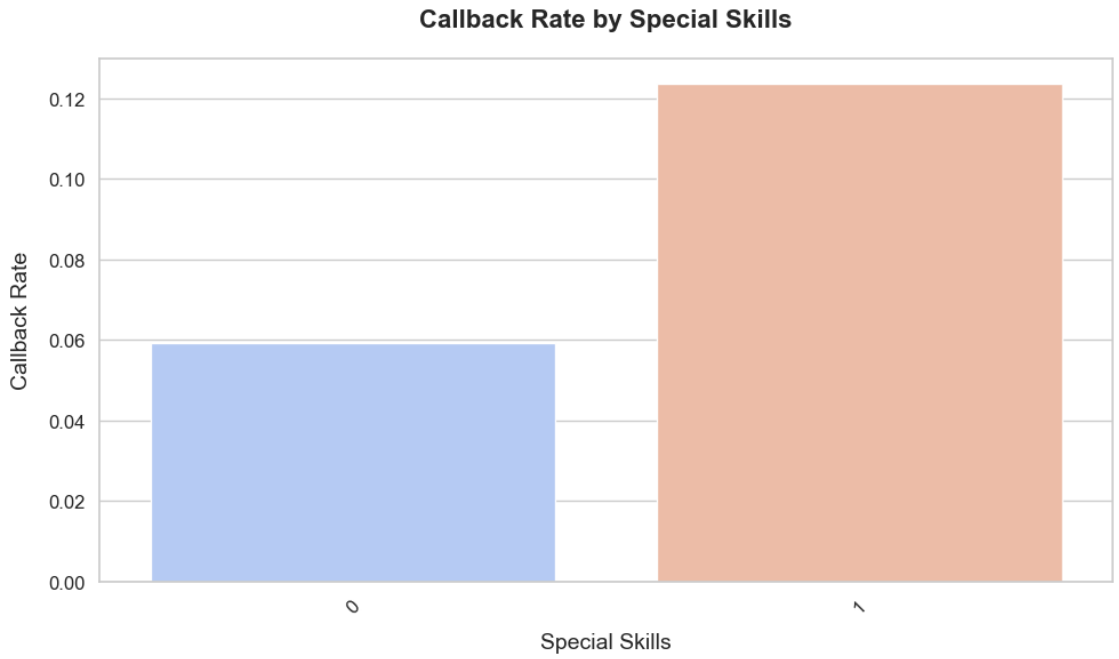

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```



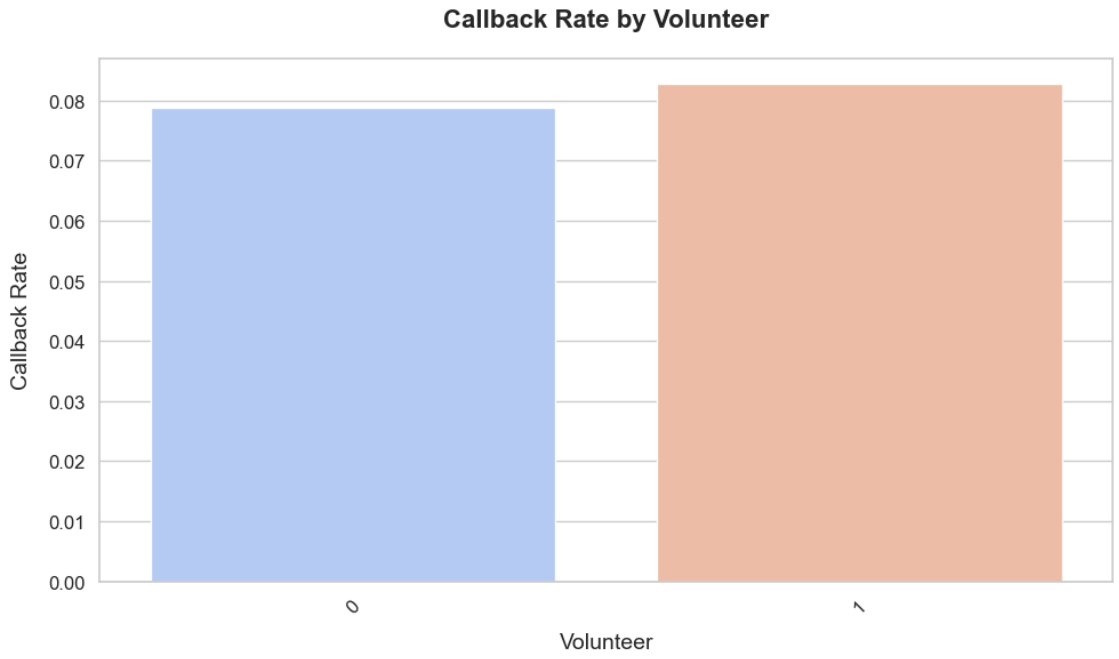

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```



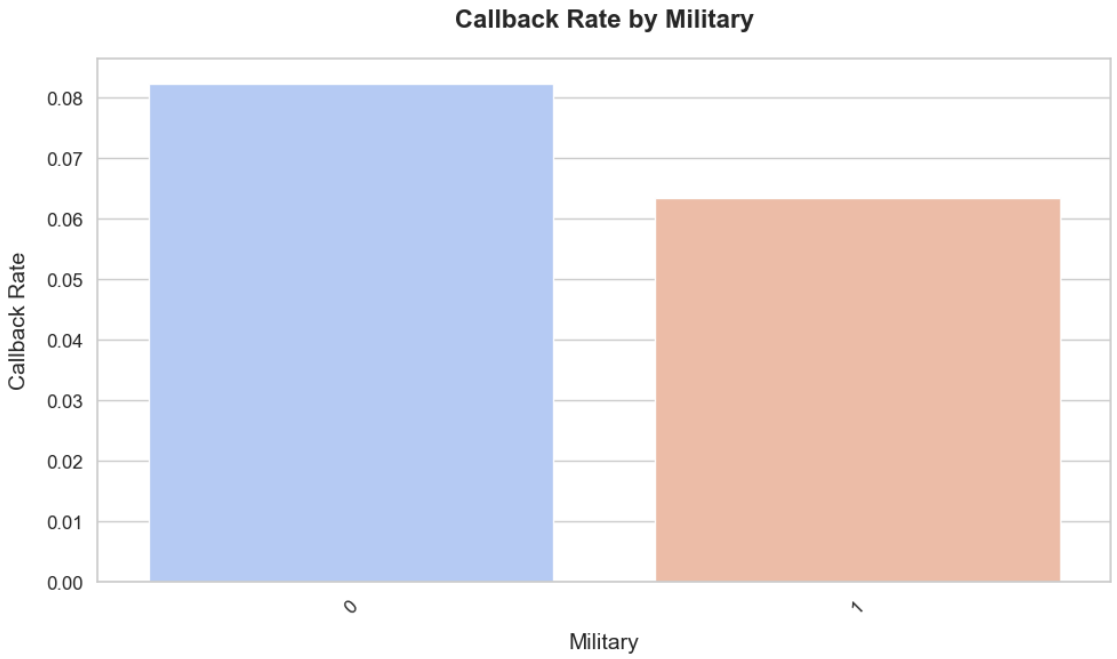

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```



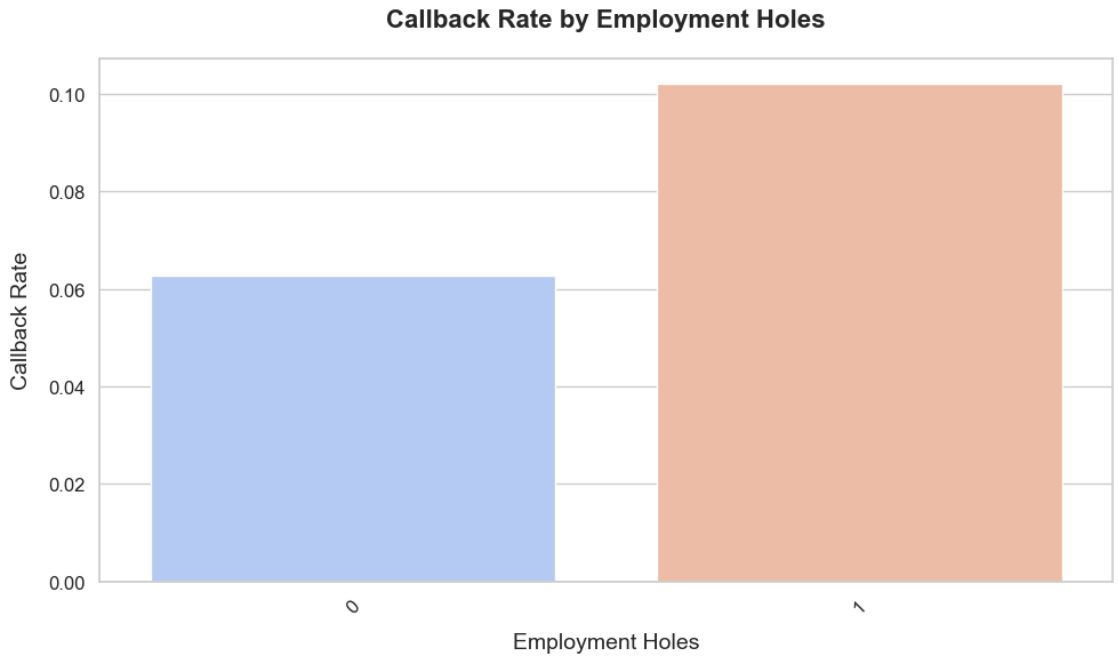

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```



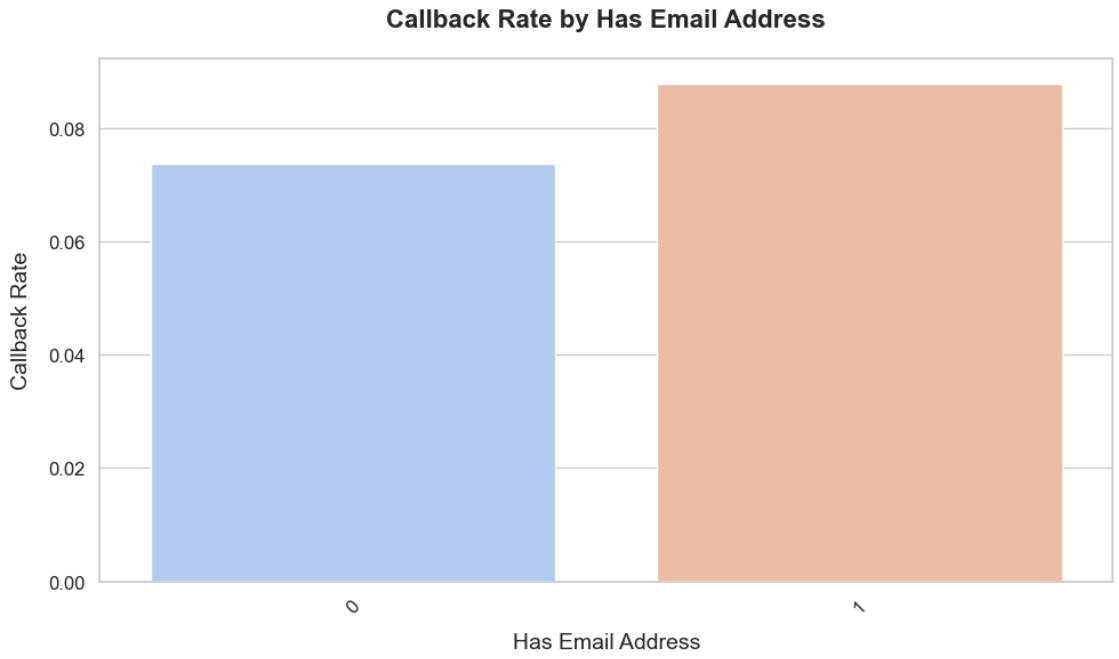

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```



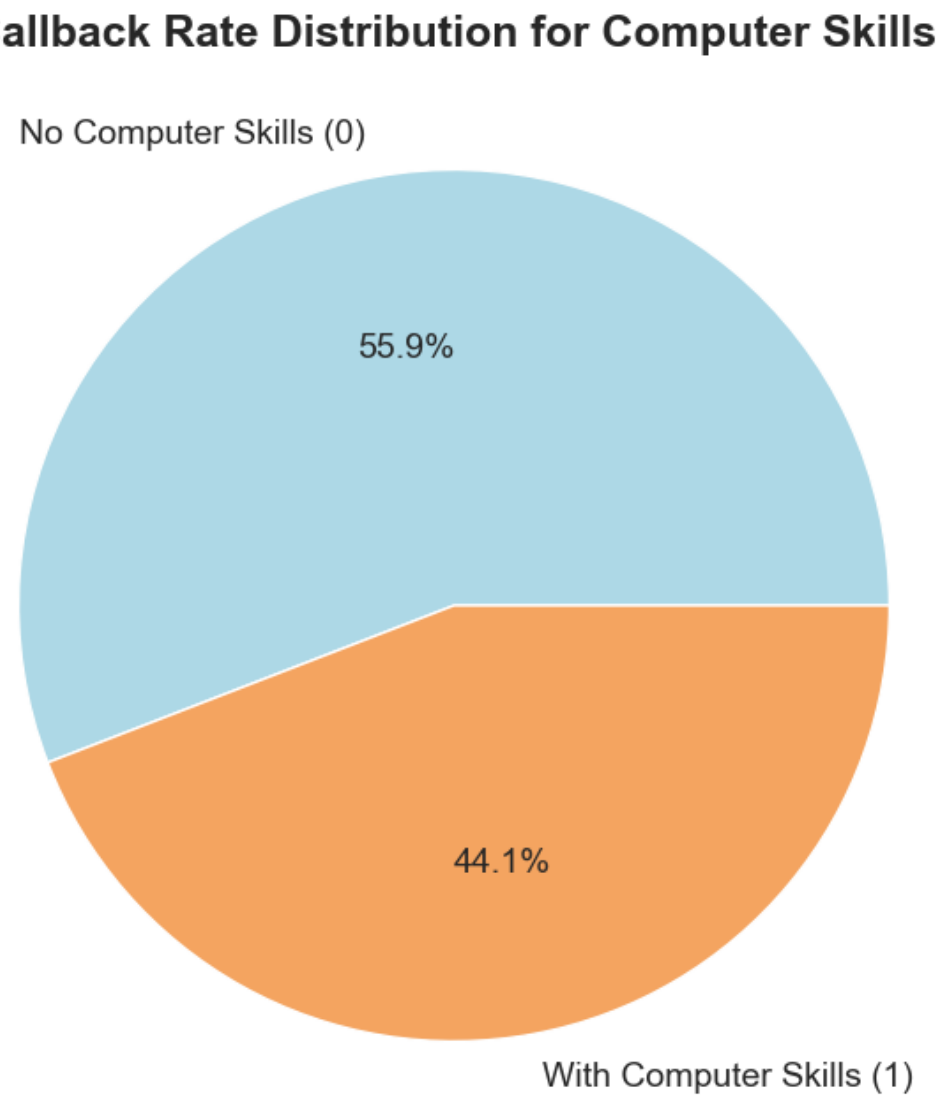

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36327/1823601790.py:1
3: FutureWarning:

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')
```

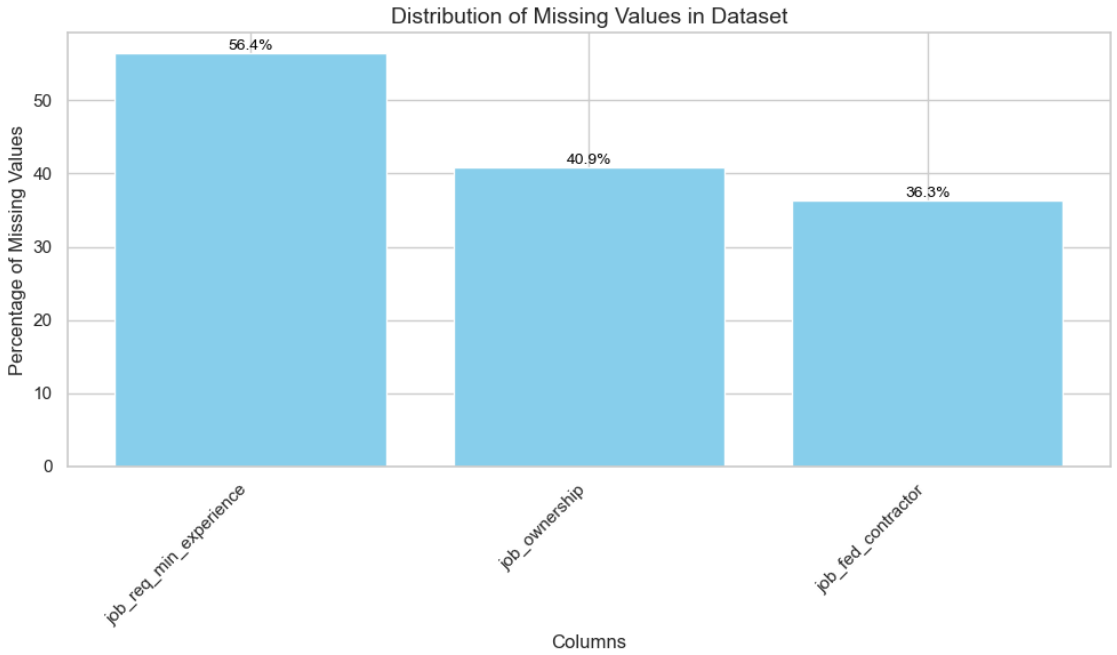


[17]:
```
# Pie chart for callback rates by computer skills with legend
def pie_chart_plot(feature, df):
    callback_counts = df.groupby(feature)['received_callback'].mean()
    labels = ['No Computer Skills (0)', 'With Computer Skills (1)']

    # 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)
```



[65]:
```
# Define placeholders for missing values
placeholders = ['unknown', 'UNK', 'Unknown']

# Replace placeholders with pd.NA
df_cleaned = df.replace(placeholders, pd.NA)

# Calculate missing values
missing_values = df_cleaned.isnull().sum()
missing_values = missing_values[missing_values > 0]  # Filter only columns with␣
↪missing values
```

```
missing_percentage = (missing_values / len(df_cleaned)) * 100  # Convert to␣
 ↪percentage

# Sort the missing values for better visualization
missing_percentage = missing_percentage.sort_values(ascending=False)

# Plot the missing values
plt.figure(figsize=(10, 6))
bars = plt.bar(missing_percentage.index, missing_percentage.values,␣
 ↪color='skyblue')
plt.xticks(rotation=45, ha="right")
plt.title("Distribution of Missing Values in Dataset", fontsize=14)
plt.ylabel("Percentage of Missing Values")
plt.xlabel("Columns")

# Add the percentage values on top of each bar
for bar, perc in zip(bars, missing_percentage.values):
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width() / 2, height, f"{perc:.1f}%",
             ha='center', va='bottom', fontsize=10, color='black')

plt.tight_layout()
plt.savefig("/Users/fruit/Desktop/DATA1030-Fall2024/
 ↪Key-Resume-Attributes-Impacting-Job-Callbacks/figures/missing_value_dist.
 ↪png", dpi=800)
plt.show()
```



[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']
```

[46]:
```
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]
```

[7]:
```
# 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]
```

[26]:
```
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, StandardScaler
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, f1_score,␣
 ↪make_scorer, confusion_matrix, ConfusionMatrixDisplay
import pickle
from pprint import pprint
import shap
from IPython.core.display import display
import matplotlib.pyplot as plt
from xgboost import XGBClassifier

# 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 = []
    model_name = ML_algo.__class__.__name__

    # ----------------------Preprocessing----------------------------#
    # 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)
       ])

   nr_runs = 10
   ftr_names = X.columns
   scores = np.zeros((len(ftr_names), nr_runs))

   #---------------------------------Initialize Grid Search␣
↪CV---------------------------------#
   # 10 loops
   last_y_test = None
   last_y_test_pred = None

   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, StandardScaler(), ML_algo)

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

```
        grid = GridSearchCV(pipe, param_grid=param_grid, scoring =␣
↪make_scorer(f1_score),
                            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 = f1_score(y_test, y_test_pred)

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

        last_y_test = y_test
        last_y_test_pred = y_test_pred

                # Collect predictions and true values for this random state
        if random_state == 1:
            all_results = pd.DataFrame()  # Initialize an empty DataFrame on␣
↪the first loop

        # Create a DataFrame for the current random state
        results_df = pd.DataFrame({
            "Random State": [random_state] * len(y_test),  # Add the current␣
↪random state
            "True Class": y_test.values,
            "Predicted Class": y_test_pred
        })

        # Append the results to the master DataFrame
        all_results = pd.concat([all_results, results_df], ignore_index=True)

    #----------------------------------Permutation␣
↪Importance----------------------------------#
        # Permutation Importance

        for i in range(len(ftr_names)):
            #print('shuffling ' + str(ftr_names[i]))
            acc_scores = []
            for j in range(nr_runs):
                # Shuffle the feature
                X_test_shuffled = X_test.copy()
                X_test_shuffled[ftr_names[i]] = np.random.
↪permutation(X_test[ftr_names[i]].values)
                # Calculate and store the F1 score
```

```
                preprocessor = grid.best_estimator_.
↪named_steps['columntransformer']
                X_test_shuffled_transformed = preprocessor.
↪transform(X_test_shuffled)

                # Predict using the model, which expects preprocessed data
                model = grid.best_estimator_.named_steps[ML_algo.__class__.
↪__name__.lower()]
                y_pred_shuffled = model.predict(X_test_shuffled_transformed)

                # Calculate F1 score for the shuffled data
                acc_scores.append(f1_score(y_test, y_pred_shuffled))
            # Store mean accuracy score for this run
            scores[i, random_state - 1] = np.mean(acc_scores)


    print(f"Mean Test F1 score: {np.mean(test_scores)}")
    print(f"Standard Deviation of Test F1 score: {np.std(test_scores)}")

    # Plotting Permutation Importance
    sorted_indices = np.argsort(np.mean(scores, axis=1))[::-1]  # Sort indices␣
↪by descending mean score

    top_n = min(20, len(sorted_indices))  # Determine the top N features
    top_indices = sorted_indices[:top_n]

    plt.figure(figsize=(8, 8))
    plt.boxplot(scores[top_indices].T, vert=False, labels=np.
↪array(ftr_names)[top_indices])
    plt.axvline(np.mean(test_scores), color='red', linestyle='--', label='Test␣
↪Score')
    plt.title(f"Top {top_n} Permutation Importances ({ML_algo.__class__.
↪__name__})")
    plt.xlabel("F1 Score with Permuted Feature")
    plt.legend()
    plt.tight_layout(pad=3)
    plt.savefig(f"/Users/fruit/Desktop/DATA1030-Fall2024/
↪Key-Resume-Attributes-Impacting-Job-Callbacks/figures/
↪top_{top_n}_permutation_plot_{model_name}.png", dpi=800)
    plt.show()

    # After the loop, save all results to a CSV file
    results_csv_path = f"/Users/fruit/Desktop/DATA1030-Fall2024/
↪Key-Resume-Attributes-Impacting-Job-Callbacks/results/
↪all_predicted_vs_true_{model_name}.csv"
    all_results.to_csv(results_csv_path, index=False)
```

```
    print(f"All predicted and true classes for 10 random states saved to:␣
↪{results_csv_path}")

    # ---------------------- Plot Confusion Matrix ---------------------- #
    # We use the last run test set and predictions stored
    if last_y_test is not None and last_y_test_pred is not None:
        cm = confusion_matrix(last_y_test, last_y_test_pred)
        disp = ConfusionMatrixDisplay(confusion_matrix=cm)
        disp.plot()
        plt.title(f"Confusion Matrix ({ML_algo.__class__.__name__})")
        plt.savefig(f"/Users/fruit/Desktop/DATA1030-Fall2024/
↪Key-Resume-Attributes-Impacting-Job-Callbacks/figures/
↪confusion_matrix_{model_name}", dpi=800)
        plt.show()

    #-----------------------------------Shap-----------------------------------#
    # As XGBoost Classifier is best model
    if isinstance(ML_algo, XGBClassifier):
        shap.initjs()
        model_name = ML_algo.__class__.__name__.lower()
        best_model = grid.best_estimator_.named_steps[model_name]
        preprocessed_X_test = grid.best_estimator_.
↪named_steps['columntransformer'].transform(X_test)

        subset_size = min(len(preprocessed_X_test), 100)
        explainer = shap.TreeExplainer(best_model)
        shap_values = explainer.shap_values(preprocessed_X_test)
        # For binary classification, take the positive class if shap_values is␣
↪a list
        if isinstance(shap_values, list) and len(shap_values) > 1:
            shap_values = shap_values[1]

        processed_feature_names = grid.best_estimator_.
↪named_steps['columntransformer'].get_feature_names_out()

        # SHAP Summary Plot (Global)
        plt.figure(figsize=(10, 6))
        shap.summary_plot(shap_values, preprocessed_X_test,␣
↪feature_names=processed_feature_names, show=False)
        plt.tight_layout()
        plt.savefig(f"/Users/fruit/Desktop/DATA1030-Fall2024/
↪Key-Resume-Attributes-Impacting-Job-Callbacks/figures/SHAP Summary Plot -␣
↪{model_name}", dpi=800)
        plt.show()

    # ---------------------- shap feature importance plot ------------------ #
```

```
    # Calculate mean absolute SHAP values

        mean_abs_shap_values = np.mean(np.abs(shap_values), axis=0)

        # Sort features by importance
        sorted_indices = np.argsort(mean_abs_shap_values)[::-1]
        sorted_feature_names = np.array(processed_feature_names)[sorted_indices]
        sorted_shap_values = mean_abs_shap_values[sorted_indices]

        # Limit to top 20 features
        top_n = min(20, len(sorted_feature_names))
        sorted_feature_names = sorted_feature_names[:top_n]
        sorted_shap_values = sorted_shap_values[:top_n]

        # Create feature importance bar plot
        plt.figure(figsize=(12, 8))
        y_pos = np.arange(len(sorted_feature_names))
        colors = plt.cm.Blues(np.linspace(0.4, 1, len(y_pos)))

        # Horizontal bar plot
        plt.barh(y_pos, sorted_shap_values, color=colors)
        plt.yticks(y_pos, sorted_feature_names)
        plt.gca().invert_yaxis()  # Invert y-axis to have the highest␣
↪importance at the top

        # Add labels and title
        plt.title(f"Top {top_n} Features by SHAP Mean Absolute Values -␣
↪{model_name}", fontsize=16, weight="bold")
        plt.xlabel("Mean |SHAP Value|")
        plt.ylabel("Features")

        # Add data labels
        for i, v in enumerate(sorted_shap_values):
            plt.text(v + 0.01, i, f'{v:.3f}', va='center')

        # Save and display the plot
        plt.tight_layout()
        plt.savefig(f"/Users/fruit/Desktop/DATA1030-Fall2024/
↪Key-Resume-Attributes-Impacting-Job-Callbacks/figures/
↪SHAP_Top_{top_n}_Feature_Importance_{model_name}.png", dpi=800)
        plt.show()

        # ---------------------- SHAP Force Plot ---------------------- #
        # Transform the test set using the pipeline preprocessor
        preprocessor = grid.best_estimator_.named_steps['columntransformer']
        preprocessed_X_test = preprocessor.transform(X_test)
```

```
        # Extract the trained XGBoost model
        xgb_model = grid.best_estimator_.named_steps['xgbclassifier']

        # Create the SHAP explainer for the XGBoost model
        explainer = shap.TreeExplainer(xgb_model)

        # Calculate SHAP values
        shap_values = explainer.shap_values(preprocessed_X_test)

        # If shap_values is a list (e.g., for binary classification), take the␣
↪positive class
        if isinstance(shap_values, list) and len(shap_values) > 1:
            shap_values = shap_values[1]

        # Define indices of instances to explain
        plot_indices = [1, 525, 777]
        plot_indices = [idx for idx in plot_indices if idx <␣
↪len(preprocessed_X_test)]

        # Generate and save SHAP force plots
        for index in plot_indices:
            print(f"Generating force plot for index: {index}")
            expected_value = explainer.expected_value[1] if␣
↪isinstance(explainer.expected_value, list) else explainer.expected_value

            predicted_class = y_test_pred[index]
            print(f"Index {index} - Predicted Class: {predicted_class}, True␣
↪Class: {y_test.iloc[index]}")

            # Generate SHAP force plot for the specific index
            force_plot = shap.force_plot(
                expected_value,
                shap_values[index],
                features=preprocessed_X_test[index],
                feature_names=preprocessor.get_feature_names_out()
            )
            display(force_plot)

            # Save the plot to an HTML file
            file_path = f"/Users/fruit/Desktop/DATA1030-Fall2024/
↪Key-Resume-Attributes-Impacting-Job-Callbacks/figures/force_plot_{index}_xgb.
↪html"
            shap.save_html(file_path, force_plot)
            print(f"Force plot saved to: {file_path}")

        # ---------------------- XGBoost Gain Plot ------------------------ #
        # Extract the best XGBoost model
```

```
        xgb_model = grid.best_estimator_.named_steps['xgbclassifier']

        # Retrieve the transformed feature names
        feature_names = grid.best_estimator_.named_steps['columntransformer'].
↪get_feature_names_out()

        # Retrieve gain-based feature importance from the model
        gain_importance = xgb_model.get_booster().
↪get_score(importance_type='gain')

        # Map feature indices (f0, f1, ...) to actual feature names
        mapped_gain_importance = {feature_names[int(k[1:])]: v for k, v in␣
↪gain_importance.items()}

        # Sort the gain values in descending order
        sorted_gain = sorted(mapped_gain_importance.items(), key=lambda x:␣
↪x[1], reverse=True)

        # Limit to the top 20 features
        top_n = min(20, len(sorted_gain))
        top_features = sorted_gain[:top_n]

        # Prepare data for plotting
        feature_names = [item[0] for item in top_features]
        gain_values = [item[1] for item in top_features]

        # Plot the gain feature importance
        plt.figure(figsize=(12, 8))
        y_pos = np.arange(len(feature_names))
        plt.barh(y_pos, gain_values, color=plt.cm.Greens(np.linspace(0.4, 1,␣
↪len(y_pos))))
        plt.yticks(y_pos, feature_names)
        plt.gca().invert_yaxis()

        # Add labels and title
        plt.title(f"Top {top_n} Features by XGBoost Gain - {model_name}",␣
↪fontsize=16, weight="bold")
        plt.xlabel("Gain Value")
        plt.ylabel("Features")

        # Add value annotations to the bars
        for i, v in enumerate(gain_values):
            plt.text(v + 0.01, i, f'{v:.3f}', va='center')

        # Save and display the plot
        plt.tight_layout()
```

```
        plt.savefig(f"/Users/fruit/Desktop/DATA1030-Fall2024/
↪Key-Resume-Attributes-Impacting-Job-Callbacks/figures/
↪XGBoost_Top_{top_n}_Gain_{model_name}.png", dpi=800)
        plt.show()

        # ---------------------- Coefficient Feature Importance Plot␣
↪------------------ #
        if hasattr(ML_algo, "coef_") and len(ML_algo.coef_) > 0:  # For models␣
↪like Logistic Regression
            # Extract the coefficients and feature names
            coefficients = grid.best_estimator_.named_steps[ML_algo.__class__.
↪__name__.lower()].coef_.flatten()
            feature_names = preprocessor.get_feature_names_out()

            # Calculate absolute values of coefficients and sort
            abs_coefficients = np.abs(coefficients)
            sorted_indices = np.argsort(abs_coefficients)[::-1]
            sorted_feature_names = np.array(feature_names)[sorted_indices]
            sorted_coefficients = abs_coefficients[sorted_indices]

            # Limit to top 15 features
            top_n = min(15, len(sorted_feature_names))
            sorted_feature_names = sorted_feature_names[:top_n]
            sorted_coefficients = sorted_coefficients[:top_n]

            # Create the plot
            plt.figure(figsize=(10, 8))
            y_pos = np.arange(len(sorted_feature_names))
            colors = plt.cm.Purples(np.linspace(0.4, 1, len(y_pos)))

            # Horizontal bar plot
            plt.barh(y_pos, sorted_coefficients, color=colors)
            plt.yticks(y_pos, sorted_feature_names)
            plt.gca().invert_yaxis()  # Invert y-axis to have the highest␣
↪importance at the top

            # Add labels and title
            plt.title(f"Top 15 Features by Absolute Coefficients - {ML_algo.
↪__class__.__name__}", fontsize=16, weight="bold")
            plt.xlabel("Absolute Coefficient Value")
            plt.ylabel("Features")

            # Add data labels
            for i, v in enumerate(sorted_coefficients):
                plt.text(v + 0.01, i, f'{v:.3f}', va='center')
```

```
            # Save and display the plot
            plt.tight_layout()
            plt.savefig(f"/Users/fruit/Desktop/DATA1030-Fall2024/
 ↪Key-Resume-Attributes-Impacting-Job-Callbacks/figures/
 ↪coeff_importance_{model_name}.png", dpi=800)
            plt.show()

    #--------------------------------Save␣
 ↪Model---------------------------------#
    algo_name = ML_algo.__class__.__name__
    save_path = f"/Users/fruit/Desktop/DATA1030-Fall2024/
 ↪Key-Resume-Attributes-Impacting-Job-Callbacks/results/{algo_name}_models.pkl"
    with open(save_path, "wb") as f:
        pickle.dump(best_models, f)
    print(f"All models saved successfully to {save_path}!")

    return test_scores, best_models
```

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36825/3537192852.py:1
6: DeprecationWarning: Importing display from IPython.core.display is deprecated
since IPython 7.14, please import from IPython display
  from IPython.core.display import display
```

[4]:
```
# read data
df = pd.read_csv("/Users/fruit/Desktop/DATA1030-Fall2024/
 ↪Key-Resume-Attributes-Impacting-Job-Callbacks/data/resume.csv")

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

[67]:
```
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)
    ])

# Apply preprocessing pipeline
processed_data = preprocessor.fit_transform(df)

# Get feature names after preprocessing
ordinal_names = ordinal_ftrs
onehot_names = preprocessor.named_transformers_['onehot'].
 ↪get_feature_names_out(onehot_ftrs)
all_feature_names = list(ordinal_names) + list(onehot_names)

# Convert to DataFrame
preprocessed_df = pd.DataFrame(processed_data, columns=all_feature_names)

# Save to CSV
output_path = '/Users/fruit/Desktop/DATA1030-Fall2024/
 ↪Key-Resume-Attributes-Impacting-Job-Callbacks/data/preprocessed_example_data.
 ↪csv'
preprocessed_df.to_csv(output_path, index=False)
print(f"Preprocessed data saved to: {output_path}")
```

```
Preprocessed data saved to: /Users/fruit/Desktop/DATA1030-Fall2024/Key-Resume-
Attributes-Impacting-Job-Callbacks/data/preprocessed_example_data.csv
```

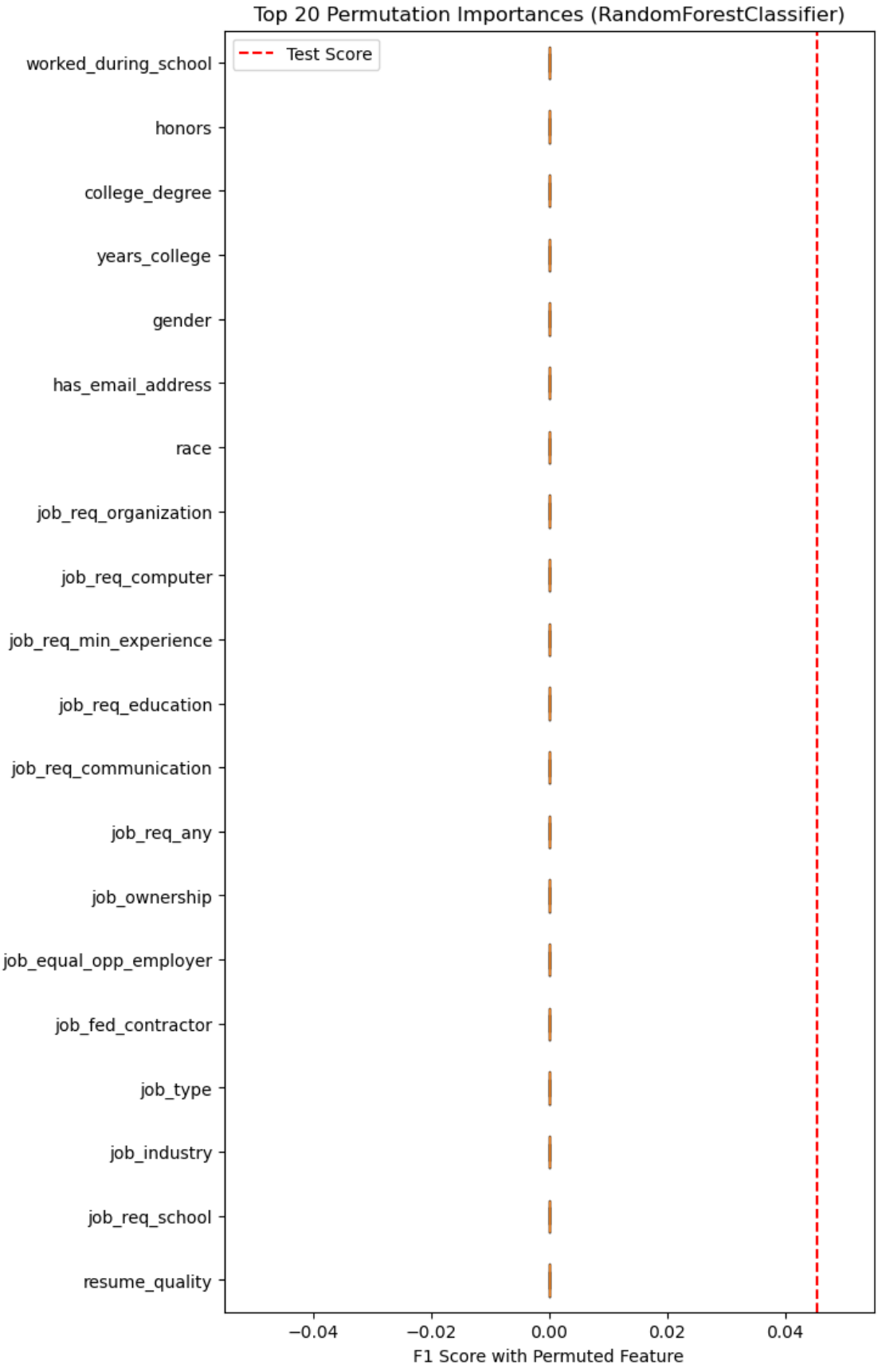
[5]:
```
# Testing
# random forest classifier
print("-------------Random Forest--------------")
rf_clf = RandomForestClassifier(random_state=42)
param_grid = {
    'randomforestclassifier__n_estimators': [50]
```

```
    }
rfc_model = MLpipe_KFold_Accuracy(X, y, ML_algo=rf_clf, param_grid=param_grid)
pprint(rfc_model)

print("------------------SVC--------------------")
param_grid_svc = {
    'svc__C': [1]  # Regularization parameter
}
svc_clf = SVC(probability=True, random_state=42)
svc_model = MLpipe_KFold_Accuracy(X, y, ML_algo=svc_clf,␣
 ↪param_grid=param_grid_svc)
pprint(svc_model)

print("---------------logistic-----------------")
param_grid_lr = {'logisticregression__C': [0.5]}
log_reg = LogisticRegression(max_iter=5000)
log_model = MLpipe_KFold_Accuracy(X, y, ML_algo=log_reg,␣
 ↪param_grid=param_grid_lr)
pprint(log_model)

print("----------------XGBoost-----------------")
param_grid_xgb = {
    'xgbclassifier__n_estimators': [50],
    'xgbclassifier__learning_rate': [0.5]
}
xgb_clf = XGBClassifier(eval_metric='auc', random_state=42, n_jobs=-1)
xgb_model = MLpipe_KFold_Accuracy(X, y, ML_algo=xgb_clf,␣
 ↪param_grid=param_grid_xgb)
print(xgb_model)
```

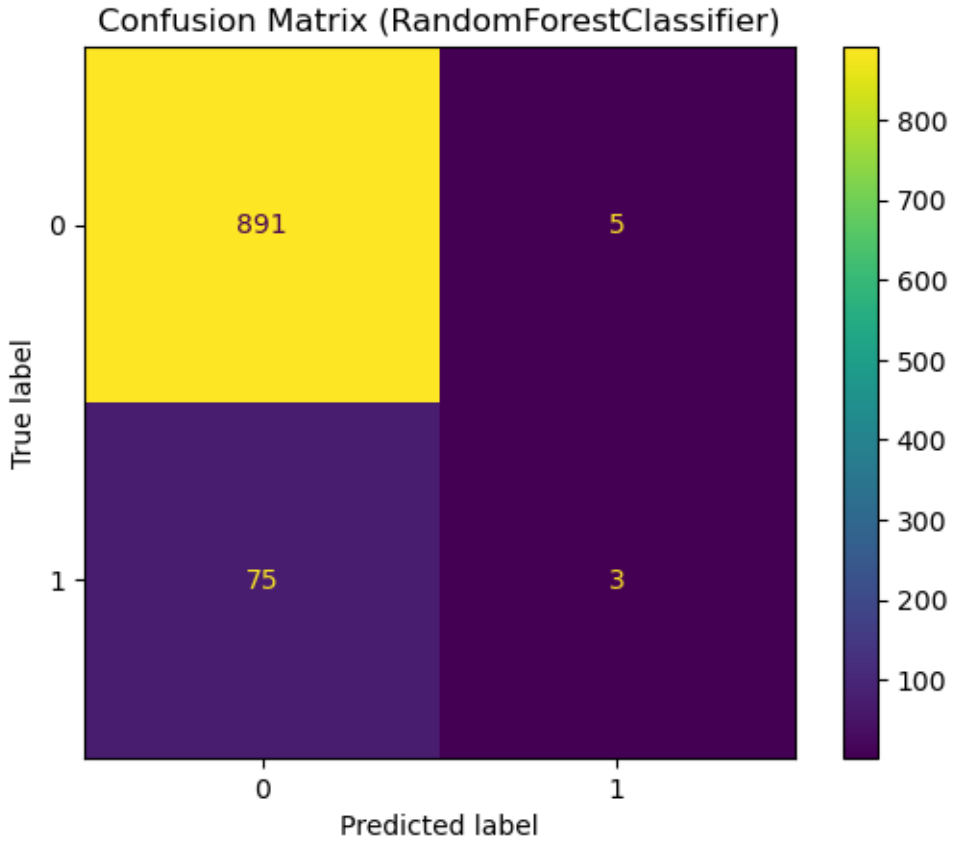
```
-------------Random Forest--------------
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Mean Test F1 score: 0.04538176181983211
Standard Deviation of Test F1 score: 0.02832546746280396
```

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36825/831763975.py:13
2: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has been
renamed 'tick_labels' since Matplotlib 3.9; support for the old name will be
dropped in 3.11.
```

```
  plt.boxplot(scores[top_indices].T, vert=False,
labels=np.array(ftr_names)[top_indices])
```


Top 20 Permutation Importances (RandomForestClassifier)
Test Score
worked_during_school
honors
college_degree
years_college
gender
has_email_address
race
job_req_organization
job_req_computer
job_req_min_experience
job_req_education
job_req_communication
job_req_any
job_ownership
job_equal_opp_employer
job_fed_contractor
job_type
job_industry
job_req_school
resume_quality
−0.04
−0.02
0.00
0.02
0.04
F1 Score with Permuted Feature

```
All predicted and true classes for 10 random states saved to:
/Users/fruit/Desktop/DATA1030-Fall2024/Key-Resume-Attributes-Impacting-Job-
Callbacks/results/all_predicted_vs_true_RandomForestClassifier.csv
```




```
All models saved successfully to /Users/fruit/Desktop/DATA1030-Fall2024/Key-
Resume-Attributes-Impacting-Job-
Callbacks/results/RandomForestClassifier_models.pkl!
([0.022988505747126436,
  0.044444444444444446,
  0.0,
  0.045454545454545456,
  0.047058823529411764,
  0.06896551724137931,
  0.06818181818181818,
  0.08695652173913043,
  0.0,
  0.06976744186046512],
 [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'])])),
                ('standardscaler', StandardScaler()),
                ('randomforestclassifier',
                 RandomForestClassifier(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'])])),
               ('standardscaler', StandardScaler()),
               ('randomforestclassifier',
                RandomForestClassifier(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'])])),
                           ('standardscaler', StandardScaler()),
                           ('randomforestclassifier',
                            RandomForestClassifier(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'])])),
                ('standardscaler', StandardScaler()),
                ('randomforestclassifier',
                 RandomForestClassifier(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'])])),
                ('standardscaler', StandardScaler()),
                ('randomforestclassifier',
                 RandomForestClassifier(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'])])),
                ('standardscaler', StandardScaler()),
                ('randomforestclassifier',
                 RandomForestClassifier(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'])])),
                ('standardscaler', StandardScaler()),
                ('randomforestclassifier',
                 RandomForestClassifier(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'])])),
                ('standardscaler', StandardScaler()),
                ('randomforestclassifier',
                 RandomForestClassifier(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'])])),
                ('standardscaler', StandardScaler()),
                ('randomforestclassifier',
                 RandomForestClassifier(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'])])),
                  ('standardscaler', StandardScaler()),
                  ('randomforestclassifier',
                   RandomForestClassifier(n_estimators=50, random_state=42))])])
------------------SVC--------------------
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Fitting 4 folds for each of 1 candidates, totalling 4 fits
```

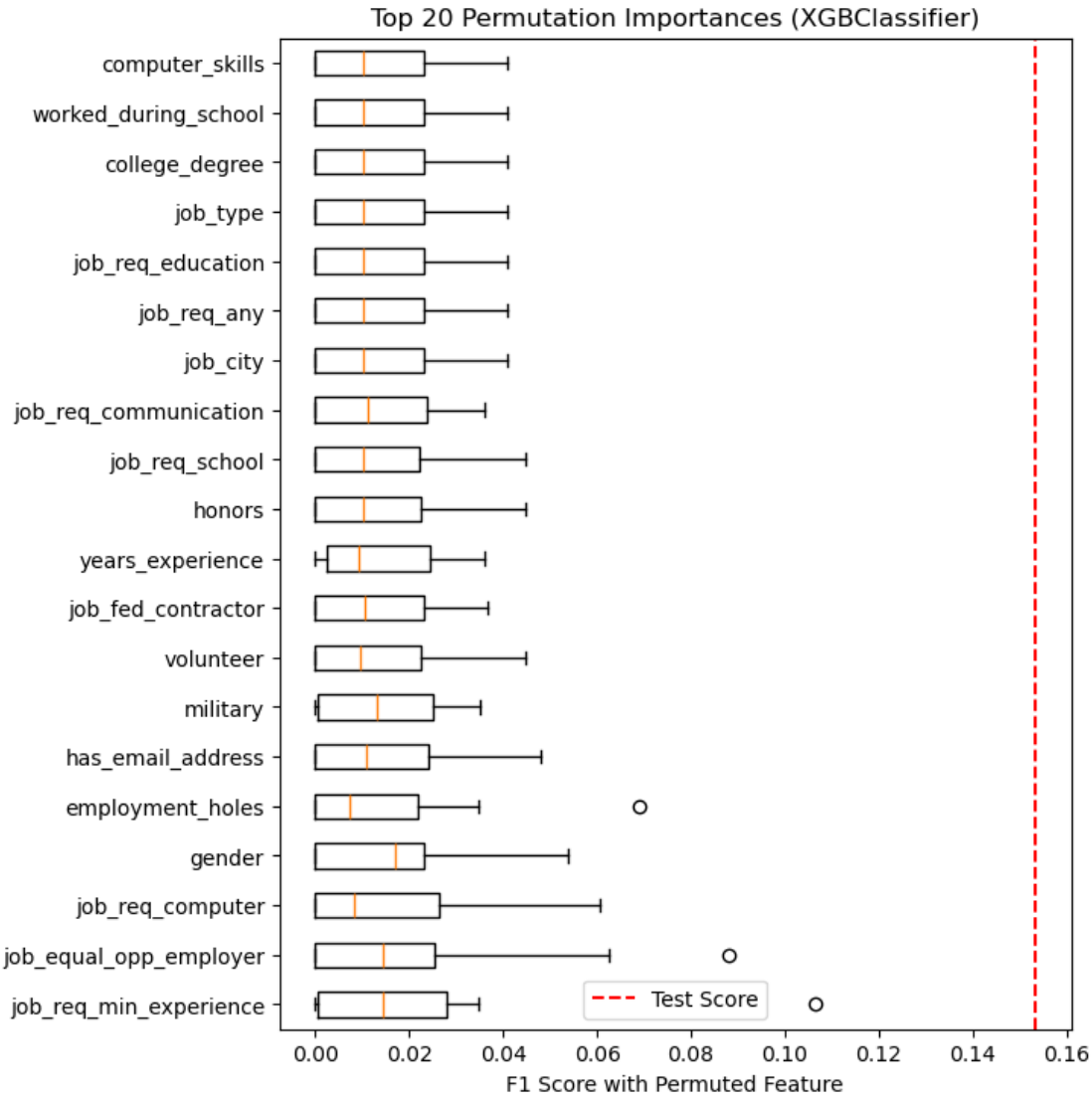
```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
Cell In[5], line 16
     12 param_grid_svc = {
     13     'svc__C': [1]  # Regularization parameter
     14 }
     15 svc_clf = SVC(probability=True, random_state=42)
---> 16 svc_model = ␣
  ↪MLpipe_KFold_Accuracy(X, y, ML_algo=svc_clf, param_grid=param_grid_svc)
     17 pprint(svc_model)
```

```
     19 print("---------------logistic-----------------")

Cell In[3], line 114, in MLpipe_KFold_Accuracy(X, y, ML_algo, param_grid)
    112 # Predict using the model, which expects preprocessed data
    113 model = grid.best_estimator_.named_steps[ML_algo.__class__.__name__.
 ↪lower()]
--> 114 y_pred_shuffled = model.predict(X_test_shuffled_transformed)
    116 # Calculate F1 score for the shuffled data
    117 acc_scores.append(f1_score(y_test, y_pred_shuffled))

File /opt/anaconda3/envs/data1030/lib/python3.12/site-packages/sklearn/svm/_bas .
 ↪py:813, in BaseSVC.predict(self, X)
    811     y = np.argmax(self.decision_function(X), axis=1)
    812 else:
--> 813     y = super().predict(X)
    814 return self.classes_.take(np.asarray(y, dtype=np.intp))

File /opt/anaconda3/envs/data1030/lib/python3.12/site-packages/sklearn/svm/_bas .
 ↪py:430, in BaseLibSVM.predict(self, X)
    428 X = self._validate_for_predict(X)
    429 predict = self._sparse_predict if self._sparse else self._dense_predict
--> 430 return predict(X)

File /opt/anaconda3/envs/data1030/lib/python3.12/site-packages/sklearn/svm/_bas .
 ↪py:449, in BaseLibSVM._dense_predict(self, X)
    441         raise ValueError(
    442             "X.shape[1] = %d should be equal to %d, "
    443             "the number of samples at training time"
    444             % (X.shape[1], self.shape_fit_[0])
    445         )
    447 svm_type = LIBSVM_IMPL.index(self._impl)
--> 449 return libsvm.predict(
    450     X,
    451     self.support_,
    452     self.support_vectors_,
    453     self._n_support,
    454     self._dual_coef_,
    455     self._intercept_,
    456     self._probA,
    457     self._probB,
    458     svm_type=svm_type,
    459     kernel=kernel,
    460     degree=self.degree,
    461     coef0=self.coef0,
    462     gamma=self._gamma,
    463     cache_size=self.cache_size,
    464 )
```

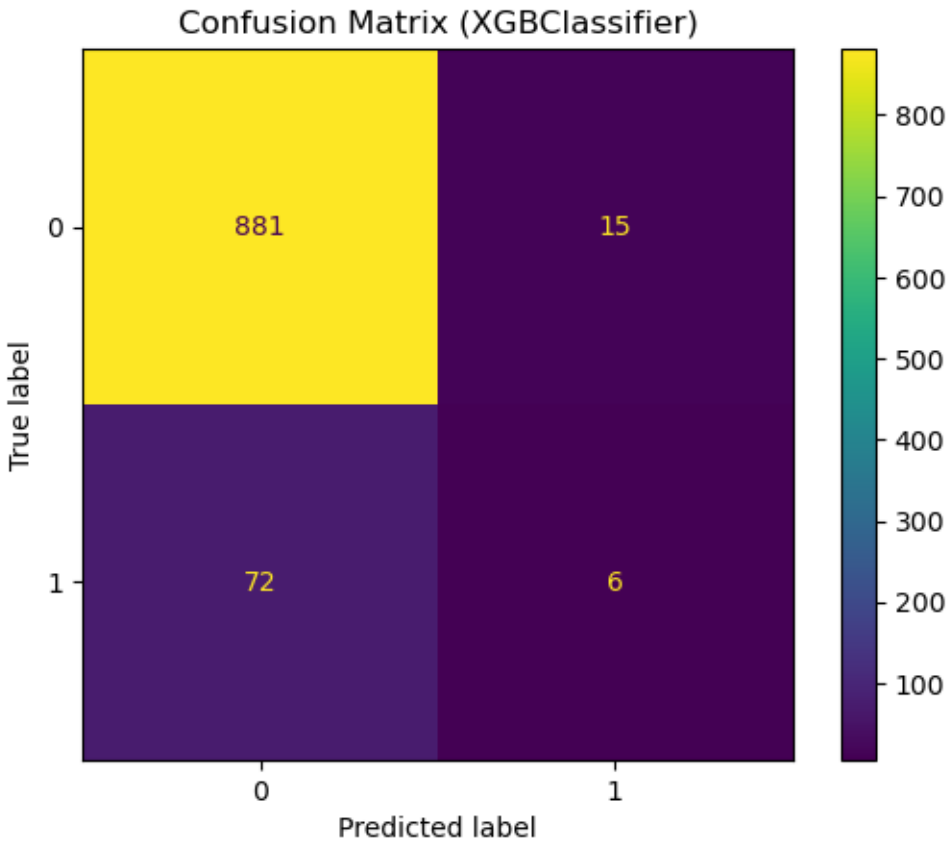
```
KeyboardInterrupt:
```

[12]:
```
print("----------------XGBoost-----------------")
param_grid_xgb = {
    'xgbclassifier__n_estimators': [50],
    'xgbclassifier__learning_rate': [0.5]
}
xgb_clf = XGBClassifier(eval_metric='auc', random_state=42, n_jobs=-1)
xgb_model = MLpipe_KFold_Accuracy(X, y, ML_algo=xgb_clf,␣
 ↪param_grid=param_grid_xgb)
print(xgb_model)
```

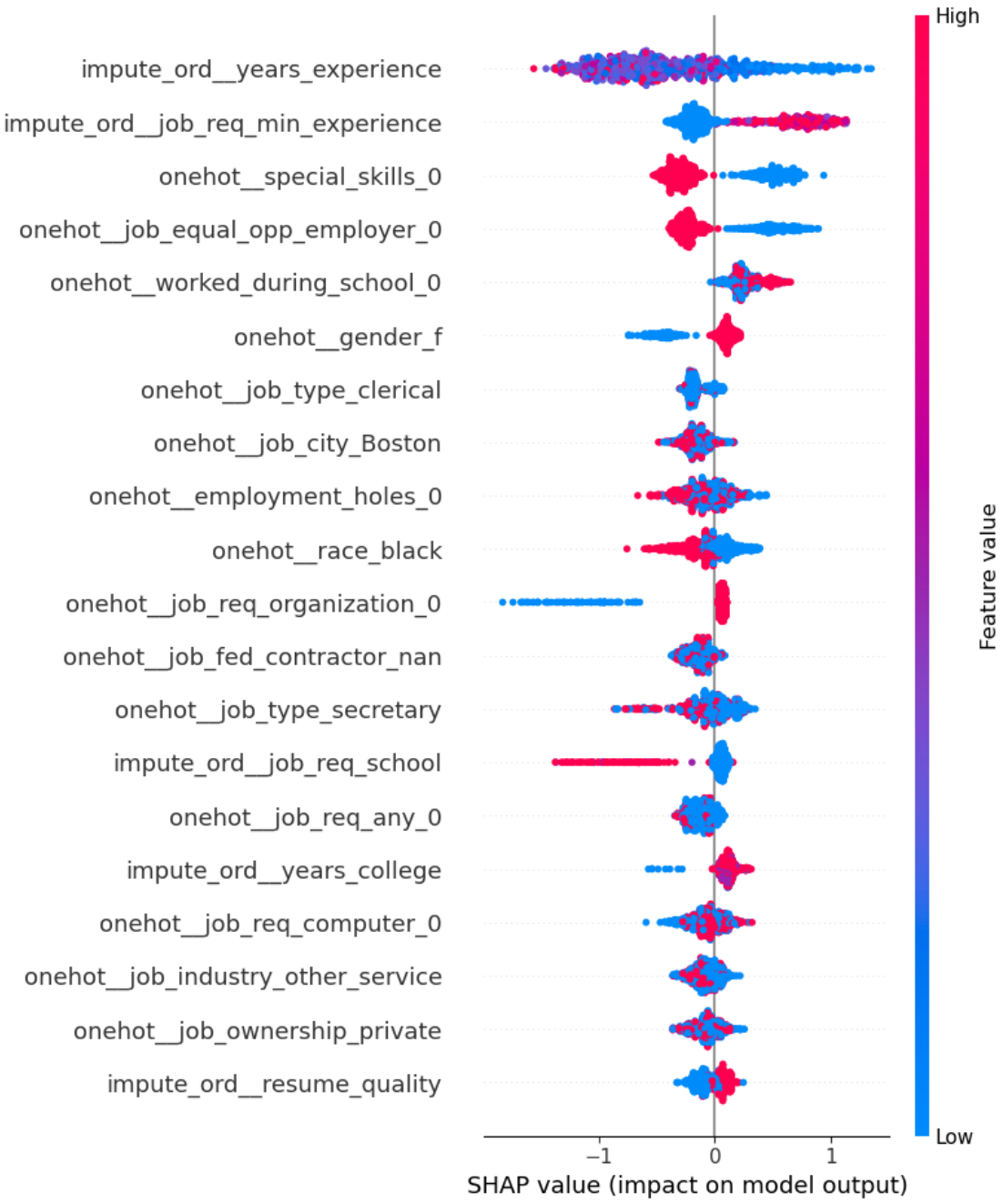
```
----------------XGBoost-----------------
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Baseline Test Score (Unshuffled): 0.909
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Baseline Test Score (Unshuffled): 0.910
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Baseline Test Score (Unshuffled): 0.911
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Baseline Test Score (Unshuffled): 0.921
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Baseline Test Score (Unshuffled): 0.911
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Baseline Test Score (Unshuffled): 0.920
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Baseline Test Score (Unshuffled): 0.909
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Baseline Test Score (Unshuffled): 0.920
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Baseline Test Score (Unshuffled): 0.914
Fitting 4 folds for each of 1 candidates, totalling 4 fits
Baseline Test Score (Unshuffled): 0.911
Mean Test F1 score: 0.15330897526624798
Standard Deviation of Test F1 score: 0.04435325351578238

/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36825/1555730581.py:1
35: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has been
renamed 'tick_labels' since Matplotlib 3.9; support for the old name will be
dropped in 3.11.
  plt.boxplot(scores[top_indices].T, vert=False,
labels=np.array(ftr_names)[top_indices])
```

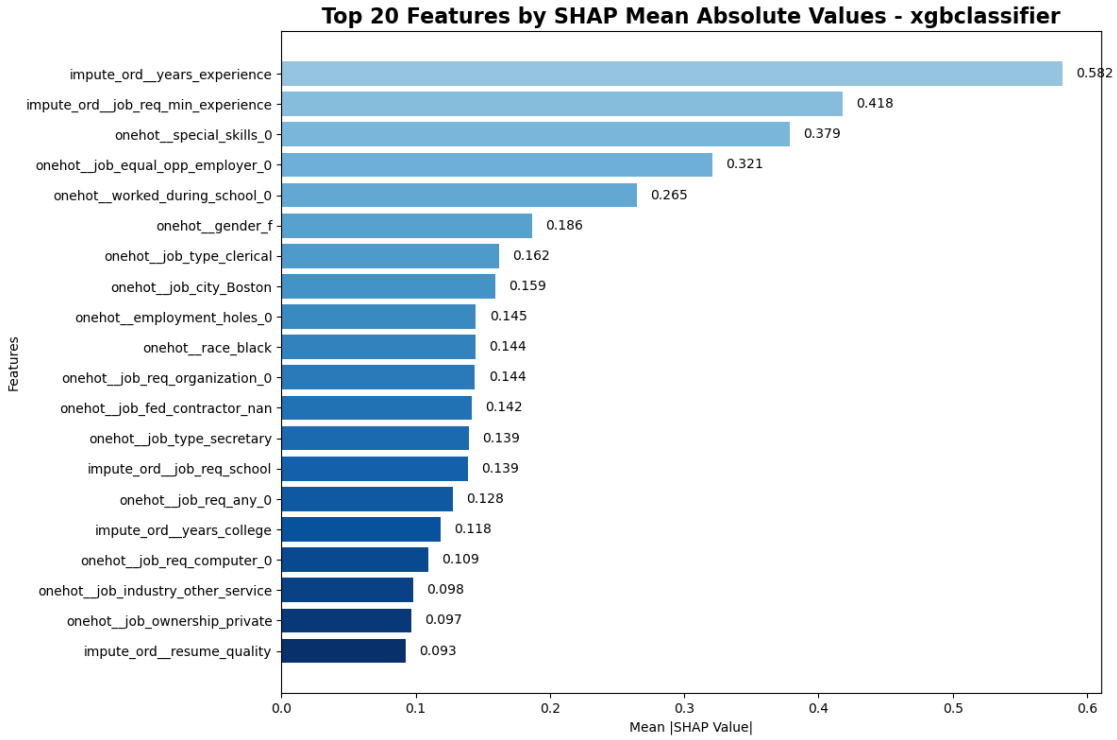


All predicted and true classes for 10 random states saved to:
/Users/fruit/Desktop/DATA1030-Fall2024/Key-Resume-Attributes-Impacting-Job-Callbacks/results/all_predicted_vs_true_XGBClassifier.csv




```
<IPython.core.display.HTML object>
```


impute_ord__years_experience
impute_ord__job_req_min_experience
onehot__special_skills_0
onehot__job_equal_opp_employer_0
onehot__worked_during_school_0
onehot__gender_f
onehot__job_type_clerical
onehot__job_city_Boston
onehot__employment_holes_0
onehot__race_black
onehot__job_req_organization_0
onehot__job_fed_contractor_nan
onehot__job_type_secretary
impute_ord__job_req_school
onehot__job_req_any_0
impute_ord__years_college
onehot__job_req_computer_0
onehot__job_industry_other_service
onehot__job_ownership_private
impute_ord__resume_quality
−1
0
1
SHAP value (impact on model output)
High
Feature value
Low




```
Generating force plot for index: 1
Index 1 - Predicted Class: 0, True Class: 0

<shap.plots._force.AdditiveForceVisualizer at 0x16be963c0>

Force plot saved to: /Users/fruit/Desktop/DATA1030-Fall2024/Key-Resume-
Attributes-Impacting-Job-Callbacks/figures/force_plot_1_xgb.html
Generating force plot for index: 525
Index 525 - Predicted Class: 0, True Class: 0

<shap.plots._force.AdditiveForceVisualizer at 0x16be978f0>

Force plot saved to: /Users/fruit/Desktop/DATA1030-Fall2024/Key-Resume-
Attributes-Impacting-Job-Callbacks/figures/force_plot_525_xgb.html
Generating force plot for index: 777
Index 777 - Predicted Class: 1, True Class: 0

<shap.plots._force.AdditiveForceVisualizer at 0x16bdf6540>

Force plot saved to: /Users/fruit/Desktop/DATA1030-Fall2024/Key-Resume-
Attributes-Impacting-Job-Callbacks/figures/force_plot_777_xgb.html
All models saved successfully to /Users/fruit/Desktop/DATA1030-Fall2024/Key-
Resume-Attributes-Impacting-Job-Callbacks/results/XGBClassifier_models.pkl!
([0.13592233009708737, 0.18518518518518517, 0.08421052631578947,
0.18947368421052632, 0.10309278350515463, 0.1702127659574468,
0.1834862385321101, 0.23529411764705882, 0.125, 0.12121212121212122],
[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=0.5,
                               max_bin=None, max_cat_threshold=None,
                               max_cat_to_onehot=None, max_delta_step=None,
                               max_depth=None, max_leaves=None,
                               min_child_weight=None, missing=nan,
                               monotone_constraints=None, multi_strategy=None,
                               n_estimators=50, n_jobs=-1,
                               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=0.5,
                               max_bin=None, max_cat_threshold=None,
                               max_cat_to_onehot=None, max_delta_step=None,
                               max_depth=None, max_leaves=None,
                               min_child_weight=None, missing=nan,
                               monotone_constraints=None, multi_strategy=None,
                               n_estimators=50, n_jobs=-1,
                               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=0.5,
                               max_bin=None, max_cat_threshold=None,
                               max_cat_to_onehot=None, max_delta_step=None,
                               max_depth=None, max_leaves=None,
                               min_child_weight=None, missing=nan,
                               monotone_constraints=None, multi_strategy=None,
                               n_estimators=50, n_jobs=-1,
                               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=0.5,
                               max_bin=None, max_cat_threshold=None,
                               max_cat_to_onehot=None, max_delta_step=None,
                               max_depth=None, max_leaves=None,
                               min_child_weight=None, missing=nan,
                               monotone_constraints=None, multi_strategy=None,
                               n_estimators=50, n_jobs=-1,
                               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=0.5,
                                max_bin=None, max_cat_threshold=None,
                                max_cat_to_onehot=None, max_delta_step=None,
                                max_depth=None, max_leaves=None,
                                min_child_weight=None, missing=nan,
                                monotone_constraints=None, multi_strategy=None,
                                n_estimators=50, n_jobs=-1,
                                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=0.5,
                                max_bin=None, max_cat_threshold=None,
                                max_cat_to_onehot=None, max_delta_step=None,
```

```
                               max_depth=None, max_leaves=None,
                               min_child_weight=None, missing=nan,
                               monotone_constraints=None, multi_strategy=None,
                               n_estimators=50, n_jobs=-1,
                               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=0.5,
                               max_bin=None, max_cat_threshold=None,
                               max_cat_to_onehot=None, max_delta_step=None,
                               max_depth=None, max_leaves=None,
                               min_child_weight=None, missing=nan,
                               monotone_constraints=None, multi_strategy=None,
                               n_estimators=50, n_jobs=-1,
                               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=0.5,
                               max_bin=None, max_cat_threshold=None,
                               max_cat_to_onehot=None, max_delta_step=None,
                               max_depth=None, max_leaves=None,
                               min_child_weight=None, missing=nan,
                               monotone_constraints=None, multi_strategy=None,
                               n_estimators=50, n_jobs=-1,
                               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=0.5,
                               max_bin=None, max_cat_threshold=None,
                               max_cat_to_onehot=None, max_delta_step=None,
                               max_depth=None, max_leaves=None,
                               min_child_weight=None, missing=nan,
                               monotone_constraints=None, multi_strategy=None,
                               n_estimators=50, n_jobs=-1,
                               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=0.5,
                                max_bin=None, max_cat_threshold=None,
                                max_cat_to_onehot=None, max_delta_step=None,
                                max_depth=None, max_leaves=None,
                                min_child_weight=None, missing=nan,
                                monotone_constraints=None, multi_strategy=None,
                                n_estimators=50, n_jobs=-1,
                                num_parallel_tree=None, random_state=42,
…))])])
```

[17]:
```
from sklearn.metrics import f1_score
from collections import Counter
import numpy as np

counter = Counter(y)
minority_class = counter.most_common()[-1][0]

# Create predictions where all instances are predicted as the minority class
y_baseline_pred = np.full_like(y, minority_class)

# Calculate the baseline F1 score
baseline_f1 = f1_score(y, y_baseline_pred, zero_division=0)

print(f"Baseline F1 score: {baseline_f1}")
```

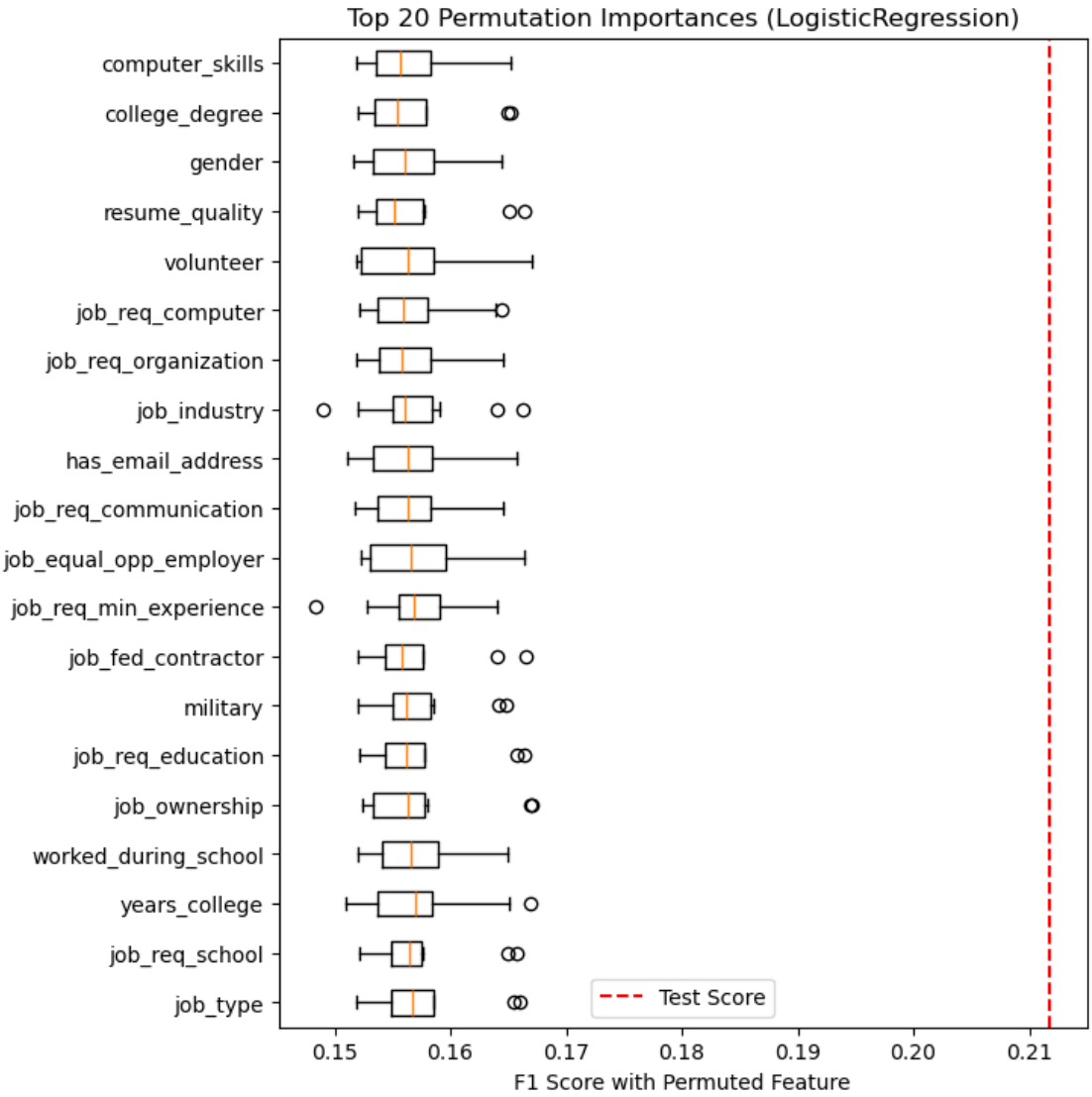
```
Baseline F1 score: 0.14899277841125047
```

[18]:
```
# logistic
print("----------------Logistic Regression-----------------")
param_grid_lr = [
    {
```

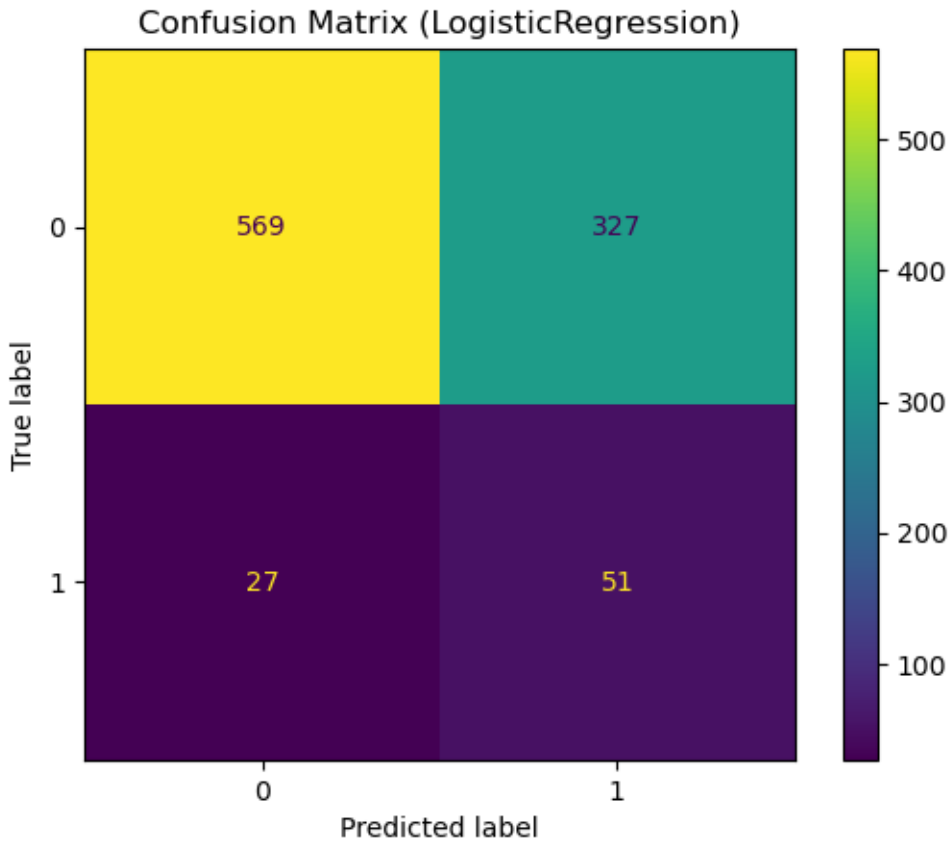
```
        'logisticregression__C': [0.001, 0.01, 0.1, 1, 10],
        'logisticregression__class_weight': ['balanced', None]
    }]
log_reg = LogisticRegression(solver='saga', max_iter=10000)
log_model = MLpipe_KFold_Accuracy(X, y, ML_algo=log_reg,␣
 ↪param_grid=param_grid_lr)
pprint(log_model)
```

```
----------------Logistic Regression-----------------
Fitting 4 folds for each of 10 candidates, totalling 40 fits
Fitting 4 folds for each of 10 candidates, totalling 40 fits
Fitting 4 folds for each of 10 candidates, totalling 40 fits
Fitting 4 folds for each of 10 candidates, totalling 40 fits
Fitting 4 folds for each of 10 candidates, totalling 40 fits
Fitting 4 folds for each of 10 candidates, totalling 40 fits
Fitting 4 folds for each of 10 candidates, totalling 40 fits
Fitting 4 folds for each of 10 candidates, totalling 40 fits
Fitting 4 folds for each of 10 candidates, totalling 40 fits
Fitting 4 folds for each of 10 candidates, totalling 40 fits
Mean Test F1 score: 0.21180921857124374
Standard Deviation of Test F1 score: 0.016249539621002115
```

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36825/3632486508.py:1
33: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has been
renamed 'tick_labels' since Matplotlib 3.9; support for the old name will be
dropped in 3.11.
  plt.boxplot(scores[top_indices].T, vert=False,
labels=np.array(ftr_names)[top_indices])
```



All predicted and true classes for 10 random states saved to:
/Users/fruit/Desktop/DATA1030-Fall2024/Key-Resume-Attributes-Impacting-Job-Callbacks/results/all_predicted_vs_true_LogisticRegression.csv




```
All models saved successfully to /Users/fruit/Desktop/DATA1030-Fall2024/Key-
Resume-Attributes-Impacting-Job-Callbacks/results/LogisticRegression_models.pkl!
([0.22790697674418606,
  0.2154566744730679,
  0.17937219730941703,
  0.22482435597189696,
  0.18636363636363637,
  0.20750551876379691,
  0.22033898305084745,
  0.22695035460992907,
  0.20568927789934355,
  0.2236842105263158],
 [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'])])),
                ('standardscaler', StandardScaler()),
                ('logisticregression',
                 LogisticRegression(C=0.001, class_weight='balanced',
                                    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_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'])])),
                ('standardscaler', StandardScaler()),
                ('logisticregression',
                 LogisticRegression(C=0.01, class_weight='balanced',
                                    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_computer',
                                                  'job_req_organization',
                                                  'race', 'gender',
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                ('standardscaler', StandardScaler()),
                ('logisticregression',
                 LogisticRegression(C=0.001, class_weight='balanced',
                                    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_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'])])),
                ('standardscaler', StandardScaler()),
                ('logisticregression',
                 LogisticRegression(C=0.01, class_weight='balanced',
                                    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_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'])])),
                ('standardscaler', StandardScaler()),
                ('logisticregression',
                 LogisticRegression(C=0.1, class_weight='balanced',
                                    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_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'])])),
                ('standardscaler', StandardScaler()),
                ('logisticregression',
                 LogisticRegression(C=0.1, class_weight='balanced',
                                    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_computer',
                                                  'job_req_organization',
                                                  'race', 'gender',
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                ('standardscaler', StandardScaler()),
                ('logisticregression',
                 LogisticRegression(C=0.001, class_weight='balanced',
                                    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_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'])])),
                ('standardscaler', StandardScaler()),
                ('logisticregression',
                 LogisticRegression(C=0.01, class_weight='balanced',
                                    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,
```

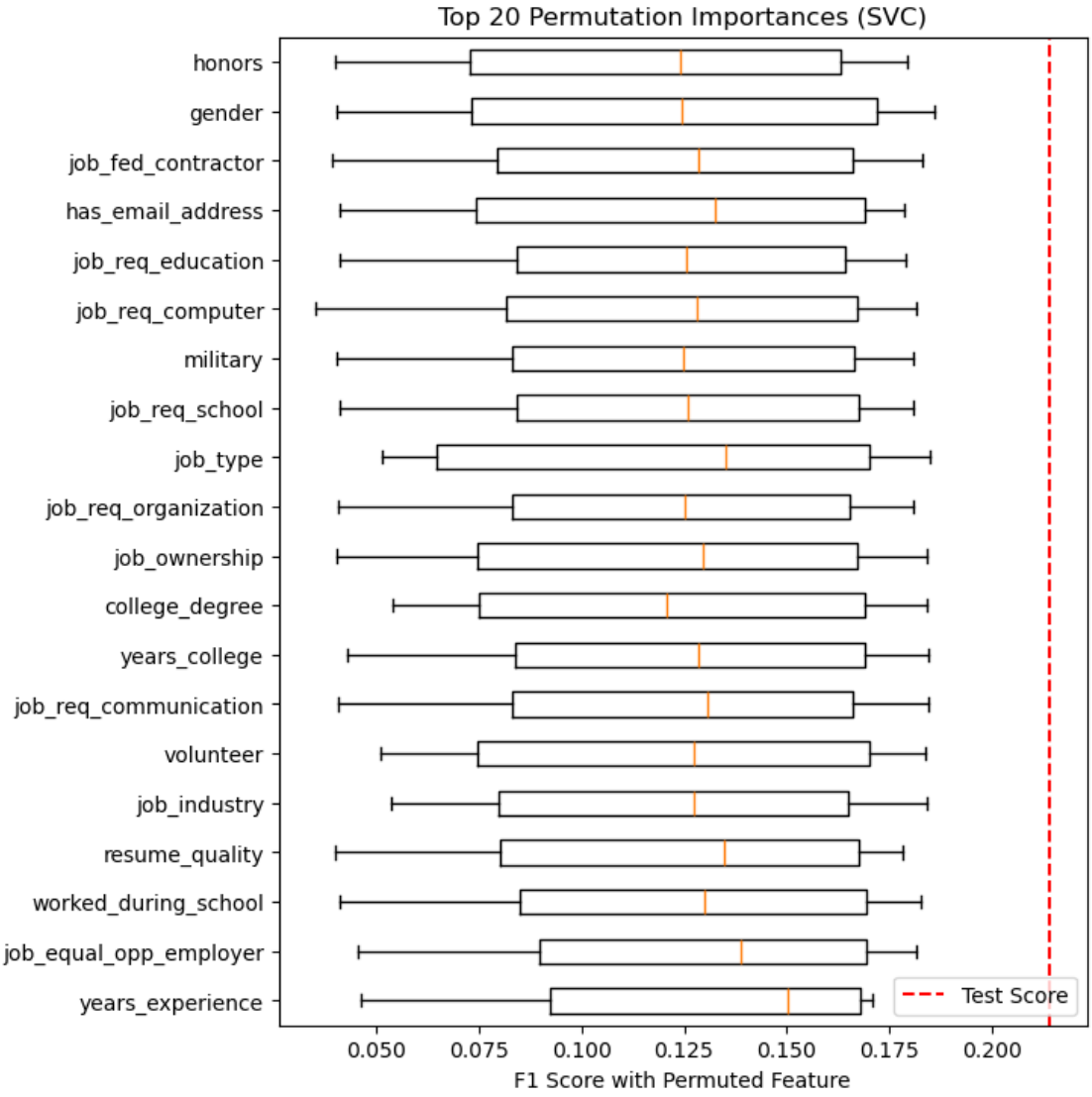
```
                4],
               [1,
                2,
                3,…
                                                 'job_req_computer',
                                                 'job_req_organization',
                                                 'race', 'gender',
                                                 'college_degree', 'honors',
                                                 'worked_during_school',
                                                 'computer_skills',
                                                 'special_skills',
                                                 'volunteer', 'military',
                                                 'employment_holes',
                                                 'has_email_address'])])),
                ('standardscaler', StandardScaler()),
                ('logisticregression',
                 LogisticRegression(C=0.001, class_weight='balanced',
                                    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_computer',
                                                    'job_req_organization',
                                                    'race', 'gender',
                                                    'college_degree', 'honors',
                                                    'worked_during_school',
                                                    'computer_skills',
                                                    'special_skills',
                                                    'volunteer', 'military',
                                                    'employment_holes',
                                                    'has_email_address'])])),
                 ('standardscaler', StandardScaler()),
                 ('logisticregression',
                  LogisticRegression(C=0.001, class_weight='balanced',
                                     max_iter=10000, solver='saga'))])])
```

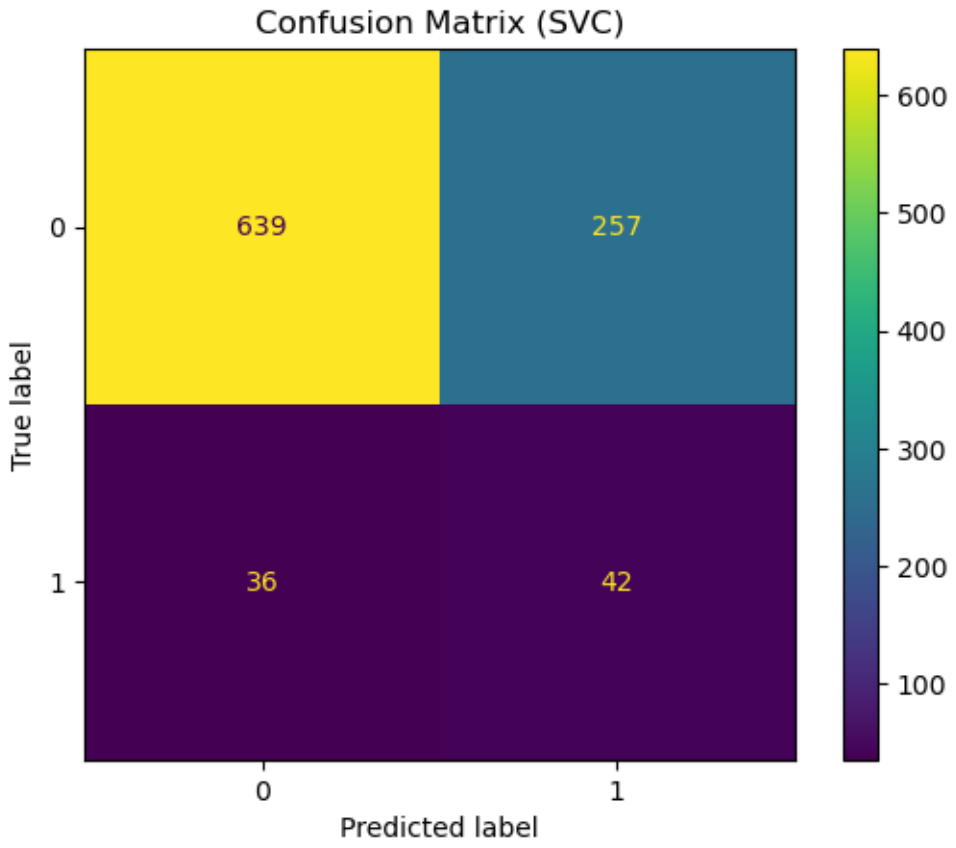
[19]:
```
param_grid_svc = {
    'svc__C': [0.01, 0.1, 1, 10],  # Regularization parameter
    'svc__class_weight': ['balanced', None]  # Adjust weights for class␣
 ↪imbalance
}

# Initialize the SVC classifier
svc_clf = SVC(probability=True, random_state=42)

# Run the pipeline with SVC
print("----------------SVC-----------------")
svc_model = MLpipe_KFold_Accuracy(X, y, ML_algo=svc_clf,␣
 ↪param_grid=param_grid_svc)
pprint(svc_model)
```

```
----------------SVC-----------------
Fitting 4 folds for each of 8 candidates, totalling 32 fits
Fitting 4 folds for each of 8 candidates, totalling 32 fits
Fitting 4 folds for each of 8 candidates, totalling 32 fits
Fitting 4 folds for each of 8 candidates, totalling 32 fits
Fitting 4 folds for each of 8 candidates, totalling 32 fits
Fitting 4 folds for each of 8 candidates, totalling 32 fits
Fitting 4 folds for each of 8 candidates, totalling 32 fits
Fitting 4 folds for each of 8 candidates, totalling 32 fits
Fitting 4 folds for each of 8 candidates, totalling 32 fits
Fitting 4 folds for each of 8 candidates, totalling 32 fits
Mean Test F1 score: 0.21425509600188763
Standard Deviation of Test F1 score: 0.023383435646381127

/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36825/3632486508.py:1
33: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has been
renamed 'tick_labels' since Matplotlib 3.9; support for the old name will be
```

```
dropped in 3.11.
  plt.boxplot(scores[top_indices].T, vert=False,
labels=np.array(ftr_names)[top_indices])
```




```
All predicted and true classes for 10 random states saved to:
/Users/fruit/Desktop/DATA1030-Fall2024/Key-Resume-Attributes-Impacting-Job-
Callbacks/results/all_predicted_vs_true_SVC.csv
```




```
All models saved successfully to /Users/fruit/Desktop/DATA1030-Fall2024/Key-
Resume-Attributes-Impacting-Job-Callbacks/results/SVC_models.pkl!
([0.21641791044776118,
  0.2265625,
  0.18518518518518517,
  0.22888283378746593,
  0.1596958174904943,
  0.24113475177304963,
  0.2358974358974359,
  0.2109090909090909,
  0.21505376344086022,
  0.22281167108753316],
 [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'])])),
                ('standardscaler', StandardScaler()),
                ('svc',
                 SVC(C=1, class_weight='balanced', probability=True,
                     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'])])),
                ('standardscaler', StandardScaler()),
                ('svc',
                 SVC(C=1, class_weight='balanced', probability=True,
                     random_state=42))]),
  Pipeline(steps=[('columntransformer',
                   ColumnTransformer(transformers=[('impute_ord',
                                                    Pipeline(steps=[('imputer',
SimplePreprocessor(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'])])),
                ('standardscaler', StandardScaler()),
                ('svc',
                 SVC(C=0.1, class_weight='balanced', probability=True,
                     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'])])),
                ('standardscaler', StandardScaler()),
                ('svc',
                 SVC(C=0.1, class_weight='balanced', probability=True,
                     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'])])),
                ('standardscaler', StandardScaler()),
                ('svc',
                 SVC(C=1, class_weight='balanced', probability=True,
                     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'])])),
                ('standardscaler', StandardScaler()),
                ('svc',
                 SVC(C=1, class_weight='balanced', probability=True,
                     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'])])),
                ('standardscaler', StandardScaler()),
                ('svc',
                 SVC(C=0.1, class_weight='balanced', probability=True,
                     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'])])),
                ('standardscaler', StandardScaler()),
                ('svc',
                 SVC(C=1, class_weight='balanced', probability=True,
                     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'])])),
                ('standardscaler', StandardScaler()),
                ('svc',
                 SVC(C=0.1, class_weight='balanced', probability=True,
                     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',
```

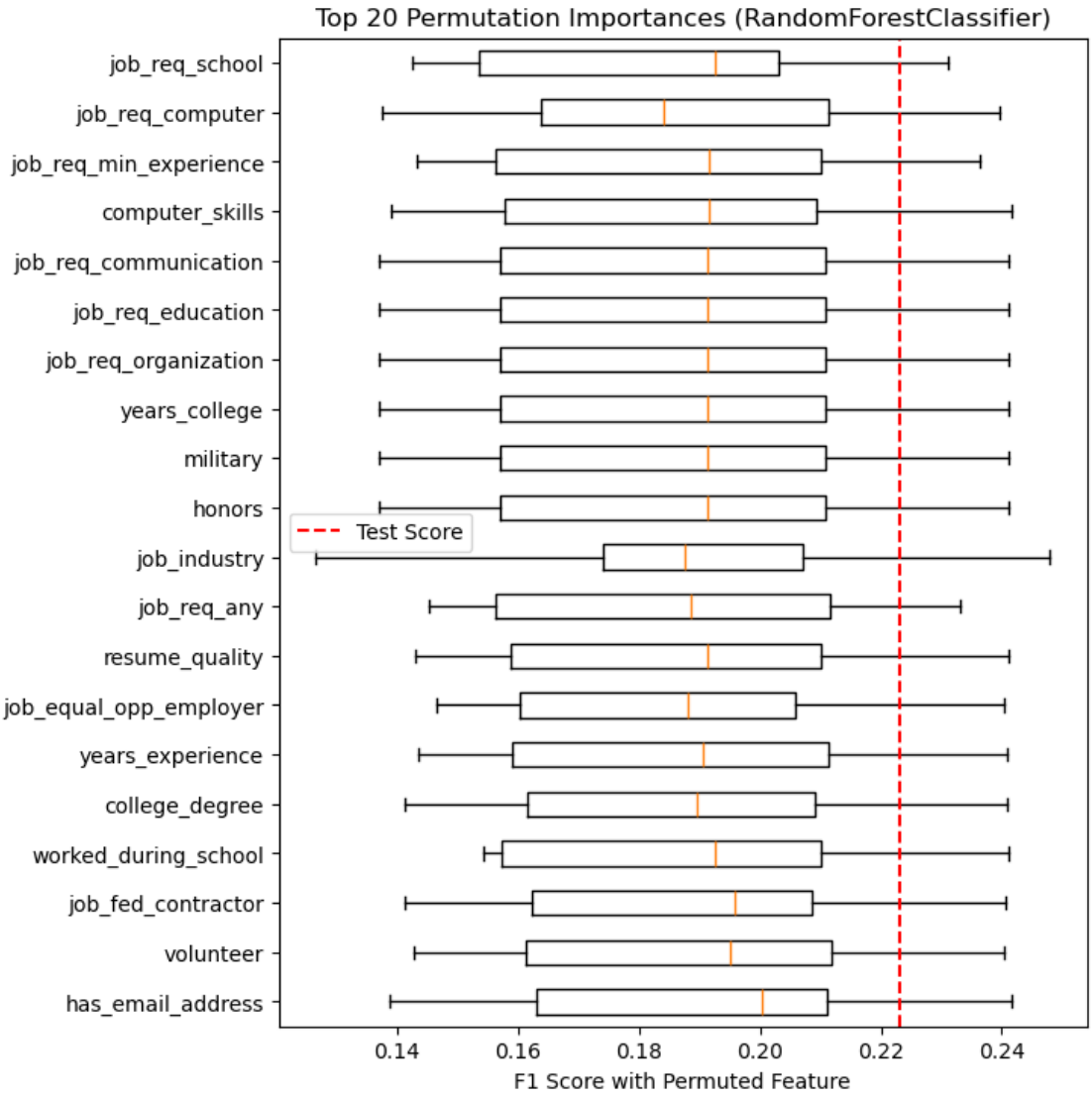
```
                 '8',
                 '10'],
                ['none_listed',
                 'high_school_grad',
                 'some_college',
                 'college'],
                [0,
                 1,
                 2,
                 3,
                 4],
                [1,
                 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'])])),
                  ('standardscaler', StandardScaler()),
                  ('svc',
                   SVC(C=0.1, class_weight='balanced', probability=True,
                       random_state=42))])])
```

[20]:
```
# random forest classifier
print("-------------Random Forest--------------")
rf_clf = RandomForestClassifier(random_state=42, n_estimators=1000, n_jobs=-1)
param_grid = {
    'randomforestclassifier__max_depth': [1, 5, 10, 30, 50],
    'randomforestclassifier__max_features': ['sqrt', 'log2', 0.5],
    'randomforestclassifier__max_samples': [0.5, 0.75, 1.0],
    'randomforestclassifier__class_weight': [None, 'balanced']
    }
rfc_model = MLpipe_KFold_Accuracy(X, y, ML_algo=rf_clf, param_grid=param_grid)
pprint(rfc_model)
```

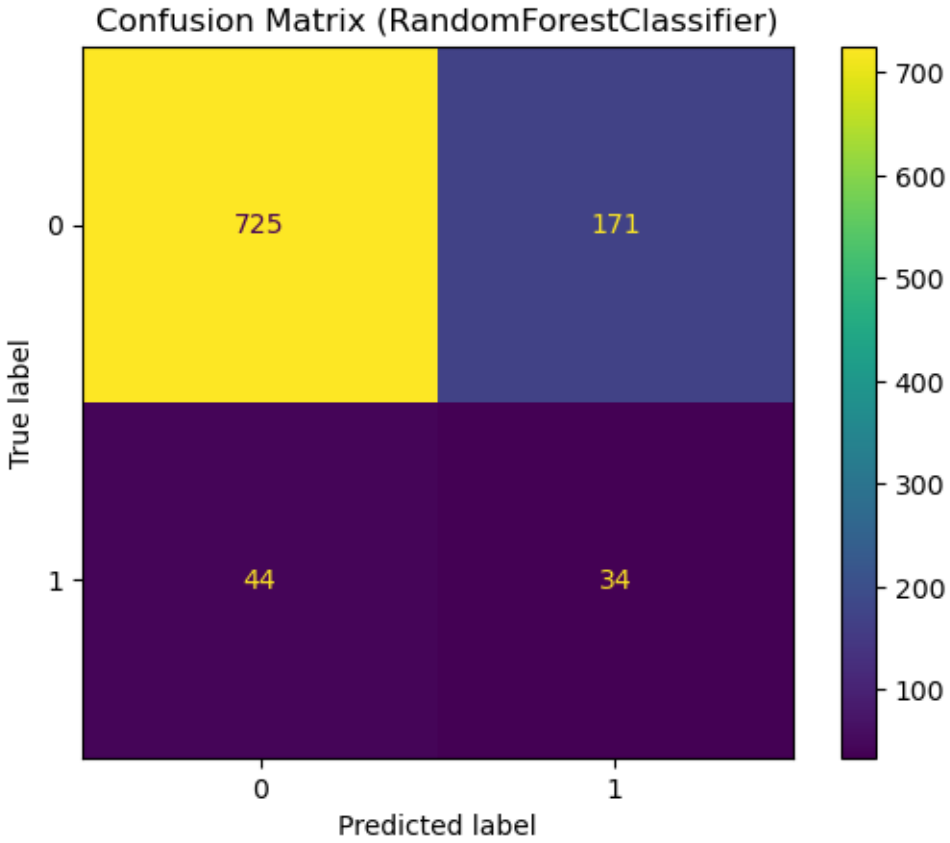
```
-------------Random Forest--------------
Fitting 4 folds for each of 90 candidates, totalling 360 fits
Fitting 4 folds for each of 90 candidates, totalling 360 fits
Fitting 4 folds for each of 90 candidates, totalling 360 fits
Fitting 4 folds for each of 90 candidates, totalling 360 fits
```

```
Fitting 4 folds for each of 90 candidates, totalling 360 fits
Fitting 4 folds for each of 90 candidates, totalling 360 fits
Fitting 4 folds for each of 90 candidates, totalling 360 fits
Fitting 4 folds for each of 90 candidates, totalling 360 fits
Fitting 4 folds for each of 90 candidates, totalling 360 fits
Fitting 4 folds for each of 90 candidates, totalling 360 fits
Mean Test F1 score: 0.22302347896841646
Standard Deviation of Test F1 score: 0.02587566925552086
```

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36825/3632486508.py:1
33: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has been
renamed 'tick_labels' since Matplotlib 3.9; support for the old name will be
dropped in 3.11.
  plt.boxplot(scores[top_indices].T, vert=False,
labels=np.array(ftr_names)[top_indices])
```



```
All predicted and true classes for 10 random states saved to:
/Users/fruit/Desktop/DATA1030-Fall2024/Key-Resume-Attributes-Impacting-Job-
Callbacks/results/all_predicted_vs_true_RandomForestClassifier.csv
```




```
All models saved successfully to /Users/fruit/Desktop/DATA1030-Fall2024/Key-
Resume-Attributes-Impacting-Job-
Callbacks/results/RandomForestClassifier_models.pkl!
([0.24561403508771928,
  0.21224489795918366,
  0.1981981981981982,
  0.247557003257329,
  0.17624521072796934,
  0.20477815699658702,
  0.26143790849673204,
  0.23857868020304568,
  0.2052980132450331,
  0.24028268551236748],
 [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,…
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                ('standardscaler', StandardScaler()),
                ('randomforestclassifier',
                 RandomForestClassifier(class_weight='balanced', max_depth=5,
                                        max_features='log2', max_samples=0.75,
                                        n_estimators=1000, n_jobs=-1,
                                        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,…
                                                    'college_degree', 'honors',
                                                    'worked_during_school',
                                                    'computer_skills',
                                                    'special_skills',
                                                    'volunteer', 'military',
                                                    'employment_holes',
                                                    'has_email_address'])])),
                 ('standardscaler', StandardScaler()),
                 ('randomforestclassifier',
                  RandomForestClassifier(class_weight='balanced', max_depth=5,
                                         max_features='log2', max_samples=0.5,
                                         n_estimators=1000, n_jobs=-1,
                                         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,…
                                                 'race', 'gender',
                                                 'college_degree', 'honors',
                                                 'worked_during_school',
                                                 'computer_skills',
                                                 'special_skills',
                                                 'volunteer', 'military',
                                                 'employment_holes',
                                                 'has_email_address'])])),
                ('standardscaler', StandardScaler()),
                ('randomforestclassifier',
                 RandomForestClassifier(class_weight='balanced', max_depth=5,
                                        max_samples=1.0, n_estimators=1000,
                                        n_jobs=-1, 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,…
                                                  'race', 'gender',
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                ('standardscaler', StandardScaler()),
                ('randomforestclassifier',
                 RandomForestClassifier(class_weight='balanced', max_depth=5,
                                        max_samples=0.75, n_estimators=1000,
                                        n_jobs=-1, 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,…
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                ('standardscaler', StandardScaler()),
                ('randomforestclassifier',
                 RandomForestClassifier(class_weight='balanced', max_depth=5,
                                        max_features=0.5, max_samples=0.5,
                                        n_estimators=1000, n_jobs=-1,
                                        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,…
                                                'college_degree', 'honors',
                                                'worked_during_school',
                                                'computer_skills',
                                                'special_skills',
                                                'volunteer', 'military',
                                                'employment_holes',
                                                'has_email_address'])])),
                ('standardscaler', StandardScaler()),
                ('randomforestclassifier',
                 RandomForestClassifier(class_weight='balanced', max_depth=5,
                                        max_features=0.5, max_samples=1.0,
                                        n_estimators=1000, n_jobs=-1,
                                        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,…
                                                'college_degree', 'honors',
                                                'worked_during_school',
                                                'computer_skills',
                                                'special_skills',
                                                'volunteer', 'military',
                                                'employment_holes',
                                                'has_email_address'])])),
                ('standardscaler', StandardScaler()),
                ('randomforestclassifier',
                 RandomForestClassifier(class_weight='balanced', max_depth=5,
                                        max_features=0.5, max_samples=1.0,
                                        n_estimators=1000, n_jobs=-1,
                                        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,…
                                                 'college_degree', 'honors',
                                                 'worked_during_school',
                                                 'computer_skills',
                                                 'special_skills',
                                                 'volunteer', 'military',
                                                 'employment_holes',
                                                 'has_email_address'])])),
                 ('standardscaler', StandardScaler()),
                 ('randomforestclassifier',
                  RandomForestClassifier(class_weight='balanced', max_depth=1,
                                         max_features='log2', max_samples=0.75,
                                         n_estimators=1000, n_jobs=-1,
                                         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,…
                                                 'race', 'gender',
                                                 'college_degree', 'honors',
                                                 'worked_during_school',
```

```
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                 ('standardscaler', StandardScaler()),
                 ('randomforestclassifier',
                  RandomForestClassifier(class_weight='balanced', max_depth=5,
                                         max_samples=0.75, n_estimators=1000,
                                         n_jobs=-1, 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,…
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
```

```
                ('standardscaler', StandardScaler()),
                ('randomforestclassifier',
                 RandomForestClassifier(class_weight='balanced', max_depth=5,
                                        max_features=0.5, max_samples=0.75,
                                        n_estimators=1000, n_jobs=-1,
                                        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,…
                                                   'race', 'gender',
                                                   'college_degree', 'honors',
                                                   'worked_during_school',
                                                   'computer_skills',
                                                   'special_skills',
                                                   'volunteer', 'military',
                                                   'employment_holes',
                                                   'has_email_address'])])),
                ('standardscaler', StandardScaler()),
                ('randomforestclassifier',
                 RandomForestClassifier(class_weight='balanced', max_depth=5,
```

```
                                       max_samples=0.75, n_estimators=1000,
                                       n_jobs=-1, 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,…
                                                     'college_degree', 'honors',
                                                     'worked_during_school',
                                                     'computer_skills',
                                                     'special_skills',
                                                     'volunteer', 'military',
                                                     'employment_holes',
                                                     'has_email_address'])])),
                    ('standardscaler', StandardScaler()),
                    ('randomforestclassifier',
                     RandomForestClassifier(class_weight='balanced', max_depth=5,
                                            max_features=0.5, max_samples=0.5,
                                            n_estimators=1000, n_jobs=-1,
                                            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,…
                                                  'college_degree', 'honors',
                                                  'worked_during_school',
                                                  'computer_skills',
                                                  'special_skills',
                                                  'volunteer', 'military',
                                                  'employment_holes',
                                                  'has_email_address'])])),
                ('standardscaler', StandardScaler()),
                ('randomforestclassifier',
                 RandomForestClassifier(class_weight='balanced', max_depth=5,
                                        max_features=0.5, max_samples=1.0,
                                        n_estimators=1000, n_jobs=-1,
                                        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,…
                                                    'college_degree', 'honors',
                                                    'worked_during_school',
                                                    'computer_skills',
                                                    'special_skills',
                                                    'volunteer', 'military',
                                                    'employment_holes',
                                                    'has_email_address'])])),
                 ('standardscaler', StandardScaler()),
                 ('randomforestclassifier',
                  RandomForestClassifier(class_weight='balanced', max_depth=5,
                                         max_features=0.5, max_samples=1.0,
                                         n_estimators=1000, n_jobs=-1,
                                         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,…
                                                 'college_degree', 'honors',
                                                 'worked_during_school',
                                                 'computer_skills',
                                                 'special_skills',
                                                 'volunteer', 'military',
                                                 'employment_holes',
                                                 'has_email_address'])])),
                 ('standardscaler', StandardScaler()),
                 ('randomforestclassifier',
                  RandomForestClassifier(class_weight='balanced', max_depth=1,
                                         max_features='log2', max_samples=0.75,
                                         n_estimators=1000, n_jobs=-1,
                                         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,…
                                                 'race', 'gender',
                                                 'college_degree', 'honors',
                                                 'worked_during_school',
                                                 'computer_skills',
                                                 'special_skills',
                                                 'volunteer', 'military',
                                                 'employment_holes',
                                                 'has_email_address'])])),
                ('standardscaler', StandardScaler()),
                ('randomforestclassifier',
                 RandomForestClassifier(class_weight='balanced', max_depth=5,
                                        max_samples=0.75, n_estimators=1000,
                                        n_jobs=-1, 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',
```

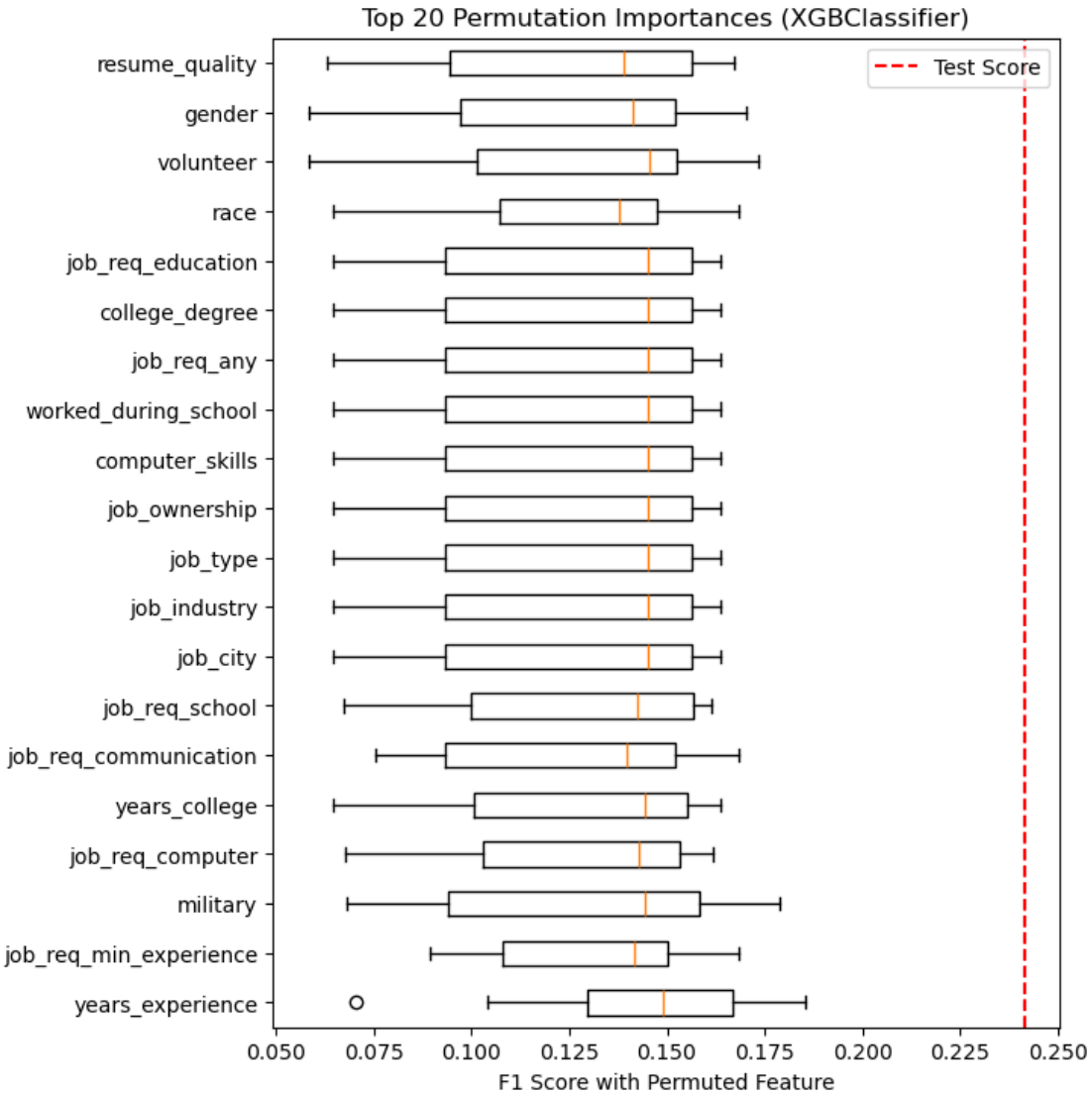
```
 '10'],
['none_listed',
 'high_school_grad',
 'some_college',
 'college'],
[0,
 1,
 2,
 3,
 4],
[1,
 2,
 3,…
                                        'college_degree', 'honors',
                                        'worked_during_school',
                                        'computer_skills',
                                        'special_skills',
                                        'volunteer', 'military',
                                        'employment_holes',
                                        'has_email_address'])])),
  ('standardscaler', StandardScaler()),
  ('randomforestclassifier',
   RandomForestClassifier(class_weight='balanced', max_depth=5,
                          max_features=0.5, max_samples=0.75,
                          n_estimators=1000, n_jobs=-1,
                          random_state=42))])])
```

[14]:
```
# KNN
#### No class_weight ####
#### Not Using ####
#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))
```

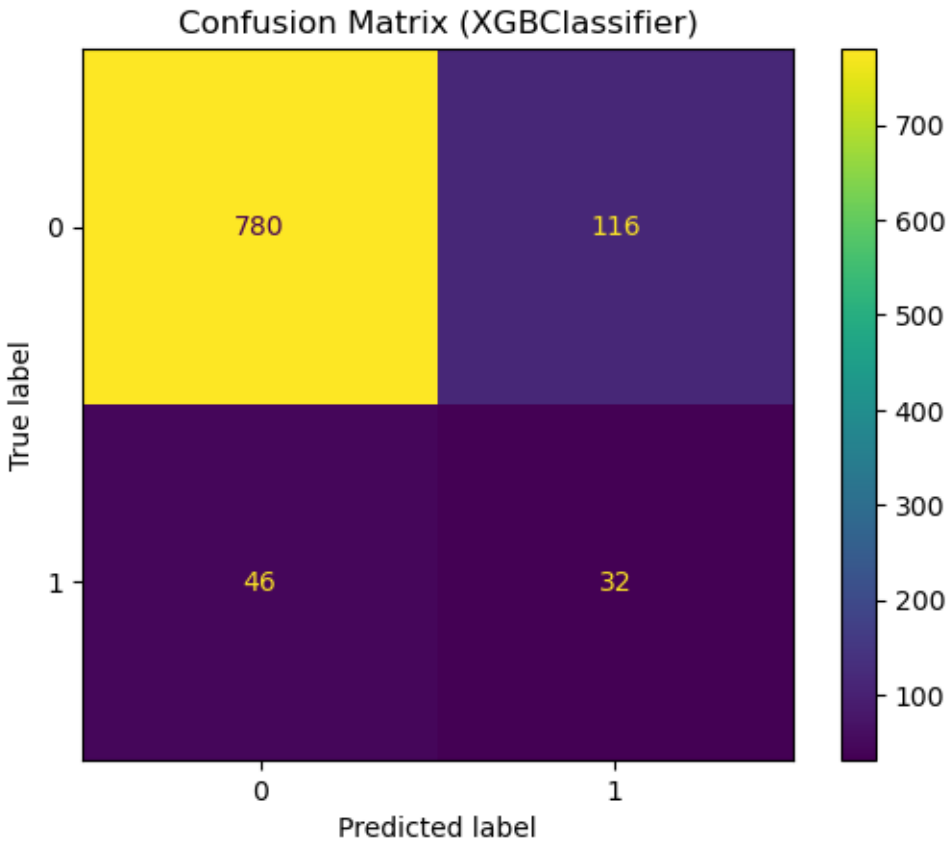
[27]:
```
from xgboost import XGBClassifier
from collections import Counter

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__learning_rate': [0.01, 0.05, 0.1],
    'xgbclassifier__max_depth': [3, 5, 10],
    'xgbclassifier__subsample': [0.75, 0.9, 1],
    '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='aucpr', n_estimators = 1000,␣
 ↪random_state=42, n_jobs=-1)

# Run the pipeline with XGBoost
print("----------------XGBoost-----------------")
xgb_model = MLpipe_KFold_Accuracy(X, y, ML_algo=xgb_clf,␣
 ↪param_grid=param_grid_xgb)
pprint(xgb_model)
```

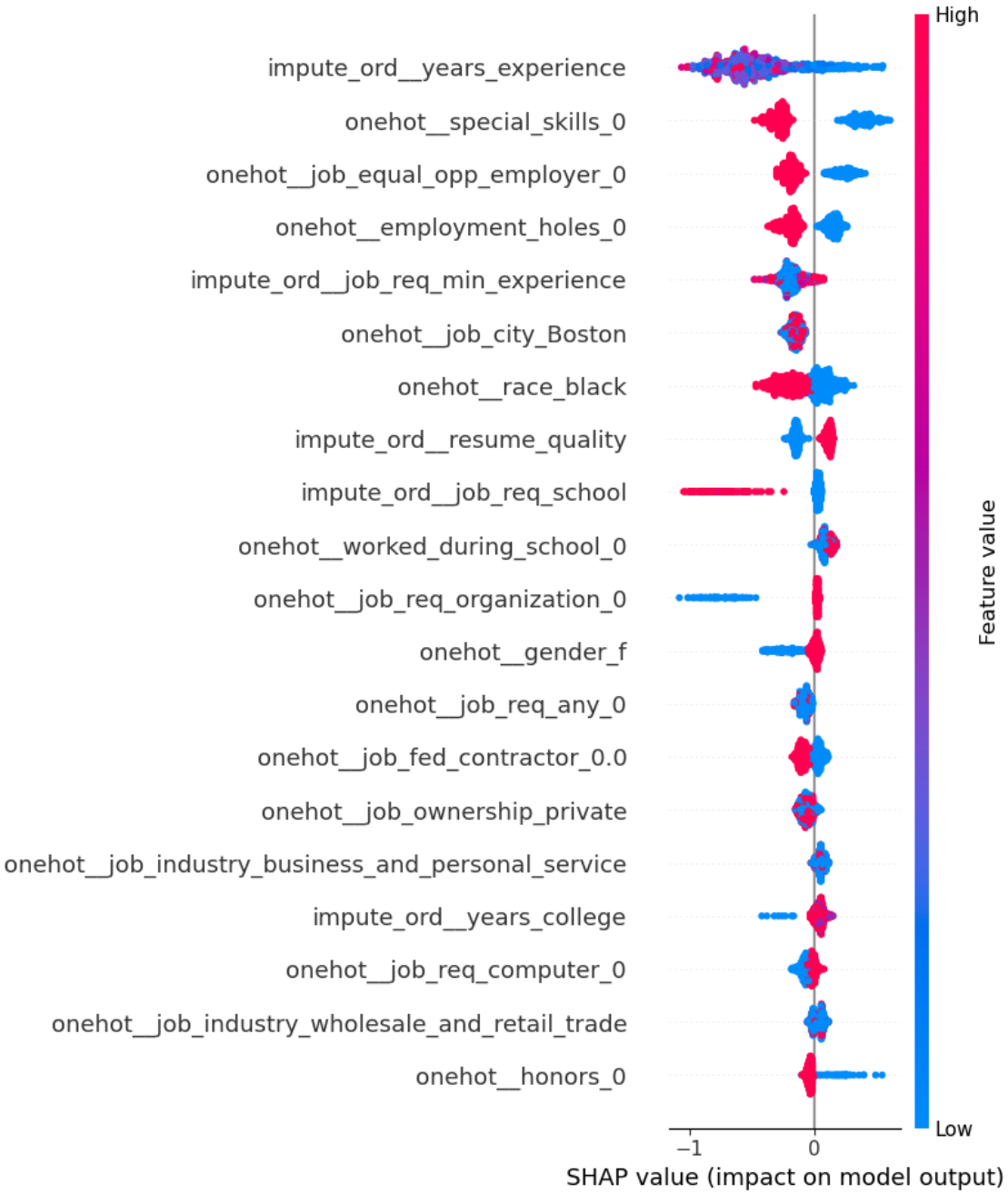
```
Suggested scale_pos_weight: 11.423469387755102
----------------XGBoost-----------------
Fitting 4 folds for each of 81 candidates, totalling 324 fits
Fitting 4 folds for each of 81 candidates, totalling 324 fits
Fitting 4 folds for each of 81 candidates, totalling 324 fits
Fitting 4 folds for each of 81 candidates, totalling 324 fits
Fitting 4 folds for each of 81 candidates, totalling 324 fits
Fitting 4 folds for each of 81 candidates, totalling 324 fits
Fitting 4 folds for each of 81 candidates, totalling 324 fits
Fitting 4 folds for each of 81 candidates, totalling 324 fits
Fitting 4 folds for each of 81 candidates, totalling 324 fits
Fitting 4 folds for each of 81 candidates, totalling 324 fits
Mean Test F1 score: 0.24140004393826806
Standard Deviation of Test F1 score: 0.028728909675321074

/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36825/3537192852.py:1
33: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has been
renamed 'tick_labels' since Matplotlib 3.9; support for the old name will be
dropped in 3.11.
  plt.boxplot(scores[top_indices].T, vert=False,
labels=np.array(ftr_names)[top_indices])
```

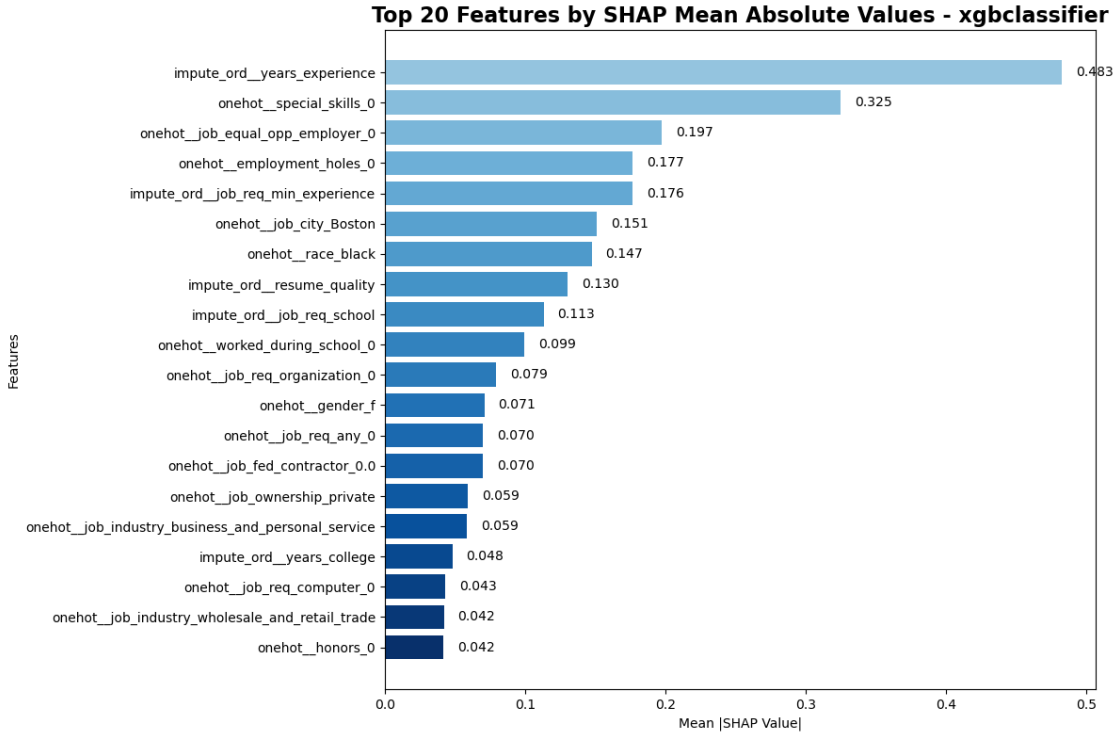


All predicted and true classes for 10 random states saved to:
/Users/fruit/Desktop/DATA1030-Fall2024/Key-Resume-Attributes-Impacting-Job-Callbacks/results/all_predicted_vs_true_XGBClassifier.csv

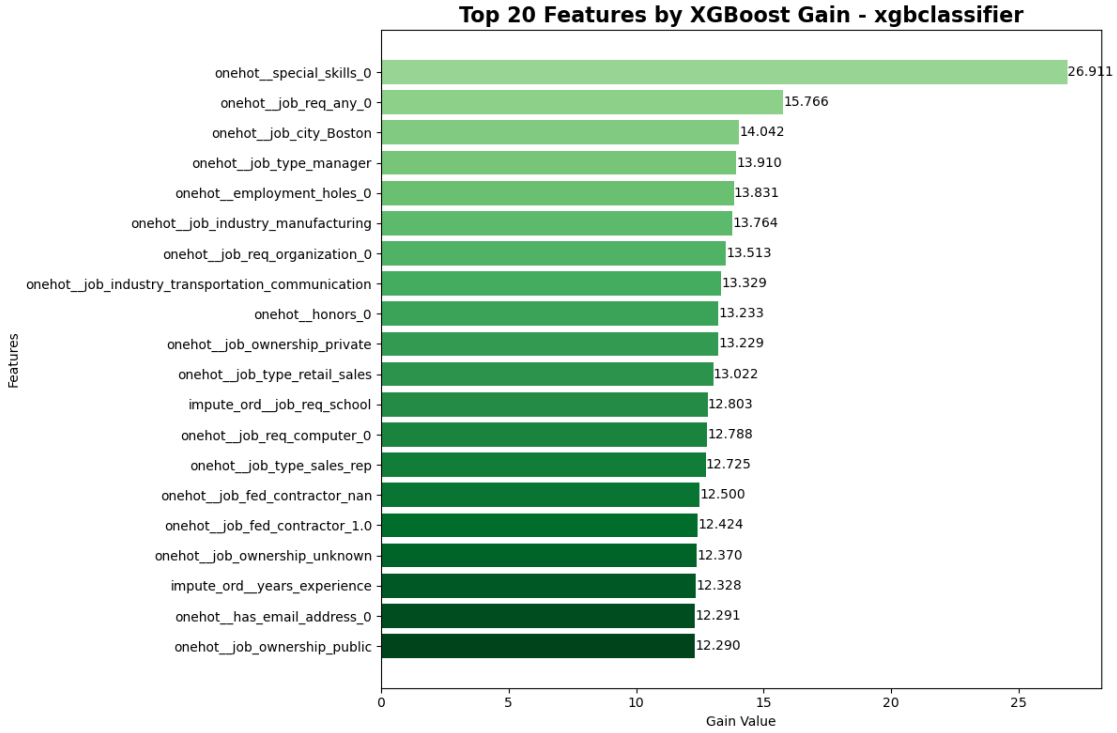



```
<IPython.core.display.HTML object>
```


impute_ord__years_experience
onehot__special_skills_0
onehot__job_equal_opp_employer_0
onehot__employment_holes_0
impute_ord__job_req_min_experience
onehot__job_city_Boston
onehot__race_black
impute_ord__resume_quality
impute_ord__job_req_school
onehot__worked_during_school_0
onehot__job_req_organization_0
onehot__gender_f
onehot__job_req_any_0
onehot__job_fed_contractor_0.0
onehot__job_ownership_private
onehot__job_industry_business_and_personal_service
impute_ord__years_college
onehot__job_req_computer_0
onehot__job_industry_wholesale_and_retail_trade
onehot__honors_0
High
Feature value
Low
−1
0
SHAP value (impact on model output)




```
Generating force plot for index: 1
Index 1 - Predicted Class: 0, True Class: 0

<shap.plots._force.AdditiveForceVisualizer at 0x30cda9970>

Force plot saved to: /Users/fruit/Desktop/DATA1030-Fall2024/Key-Resume-
Attributes-Impacting-Job-Callbacks/figures/force_plot_1_xgb.html
Generating force plot for index: 525
Index 525 - Predicted Class: 0, True Class: 0

<shap.plots._force.AdditiveForceVisualizer at 0x16363bd70>

Force plot saved to: /Users/fruit/Desktop/DATA1030-Fall2024/Key-Resume-
Attributes-Impacting-Job-Callbacks/figures/force_plot_525_xgb.html
Generating force plot for index: 777
Index 777 - Predicted Class: 1, True Class: 0

<shap.plots._force.AdditiveForceVisualizer at 0x30cdaae10>

Force plot saved to: /Users/fruit/Desktop/DATA1030-Fall2024/Key-Resume-
Attributes-Impacting-Job-Callbacks/figures/force_plot_777_xgb.html
```




```
All models saved successfully to /Users/fruit/Desktop/DATA1030-Fall2024/Key-
Resume-Attributes-Impacting-Job-Callbacks/results/XGBClassifier_models.pkl!
([0.2191780821917808,
  0.21649484536082475,
  0.24444444444444444,
  0.2937062937062937,
  0.200836820083682,
  0.25,
  0.21551724137931033,
  0.256,
  0.2346368715083799,
  0.2831858407079646],
 [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=0.01,
                               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=1000, n_jobs=-1,
                               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=1000, n_jobs=-1,
                                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=3, max_leaves=None,
                               min_child_weight=None, missing=nan,
                               monotone_constraints=None, multi_strategy=None,
                               n_estimators=1000, n_jobs=-1,
                               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=1000, n_jobs=-1,
                               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=0.01,
                               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=1000, n_jobs=-1,
```

```
                              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=1000, n_jobs=-1,
                              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=0.01,
                               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=1000, n_jobs=-1,
                               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=0.01,
                               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=1000, n_jobs=-1,
                               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=1000, n_jobs=-1,
                               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=0.01,
                              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=1000, n_jobs=-1,
                              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=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,
```

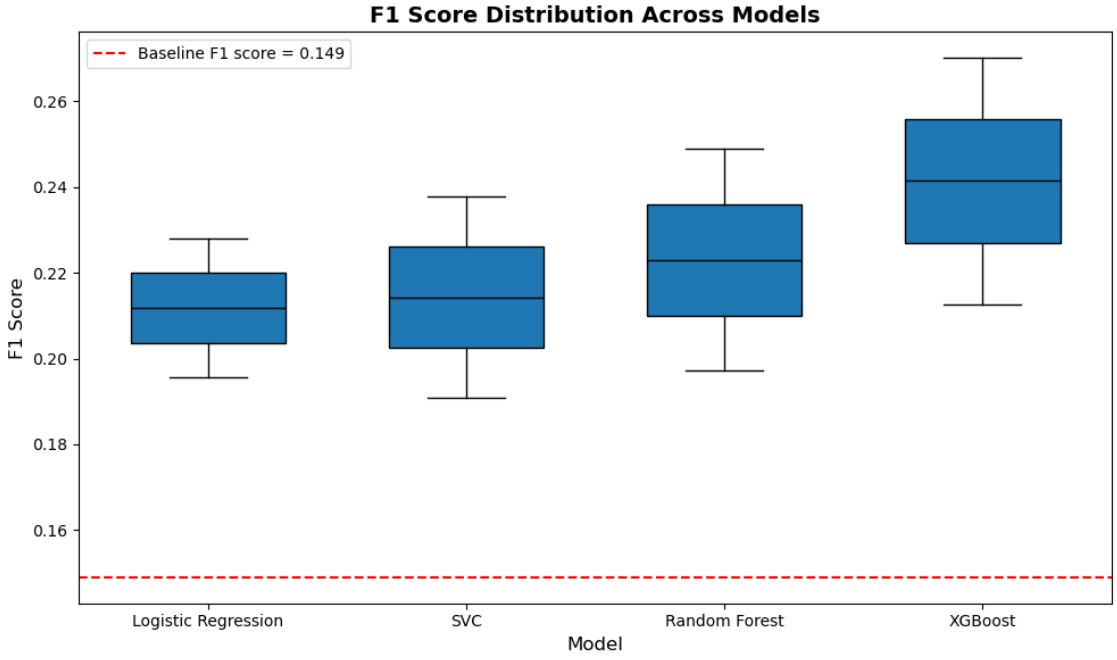
```
                                monotone_constraints=None, multi_strategy=None,
                                n_estimators=50, n_jobs=None,
                                num_parallel_tree=None, random_state=42, …))]),
    Pipeline(steps=[('columntransformer',
                     ColumnTransformer(transformers=[('impute_ord',
                                                      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, …))])]
```

[ ]:
```
import matplotlib.pyplot as plt
```

```
# Data for the models
models = ["Logistic Regression", "SVC", "Random Forest", "XGBoost"]

# Mean F1 scores and standard deviations
means = [0.2118, 0.2143, 0.2230, 0.2414]
std_devs = [0.0162, 0.0234, 0.0259, 0.0287]

# Generate data for box plot as [max, min, mean]
f1_scores = [
    [means[0] + std_devs[0], means[0] - std_devs[0], means[0]],  # XGBoost
    [means[1] + std_devs[1], means[1] - std_devs[1], means[1]],  # Random Forest
    [means[2] + std_devs[2], means[2] - std_devs[2], means[2]],  # SVC
    [means[3] + std_devs[3], means[3] - std_devs[3], means[3]],  # Logistic␣
 ↪Regression
]

# Baseline F1 score
baseline_f1 = 0.1490

# Create the box plot
plt.figure(figsize=(10, 6))
plt.boxplot(f1_scores, labels=models, patch_artist=True, widths=0.6,␣
 ↪medianprops=dict(color="black"))

# Add a horizontal line for the baseline
plt.axhline(y=baseline_f1, color='red', linestyle='--', linewidth=1.5,␣
 ↪label=f'Baseline F1 score = {baseline_f1}')

# Add title and axis labels
plt.title("F1 Score Distribution Across Models", fontsize=14, weight='bold')
plt.xlabel("Model", fontsize=12)
plt.ylabel("F1 Score", fontsize=12)

# Add a legend
plt.legend(loc='upper right')

# Show the plot
plt.tight_layout()

plt.savefig(f"/Users/fruit/Desktop/DATA1030-Fall2024/
 ↪Key-Resume-Attributes-Impacting-Job-Callbacks/figures/
 ↪F1_score_distribution_across_models", dpi=800)
plt.show()
```

```
/var/folders/z3/bw7f0jc54372c3jpx59322cw0000gn/T/ipykernel_36825/1319165980.py:2
3: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has been
renamed 'tick_labels' since Matplotlib 3.9; support for the old name will be
```

```
dropped in 3.11.
  plt.boxplot(f1_scores, labels=models, patch_artist=True, widths=0.6,
medianprops=dict(color="black"))
```



[47]:
```
# Specify the path to the pickle file
pickle_path = '/Users/fruit/Desktop/DATA1030-Fall2024/
 ↪Key-Resume-Attributes-Impacting-Job-Callbacks/results/
 ↪RandomForestClassifier_models.pkl'

# Load the saved models
with open(pickle_path, 'rb') as file:
    models = pickle.load(file)

# Assuming you saved a list of models (e.g., from GridSearchCV with multiple␣
 ↪runs):
# Inspect the best XGBoost model
for i in range(10):
    best_xgb_model = models[i]  # If multiple models, you can iterate through␣
 ↪the list
    xgb_model = best_xgb_model.named_steps['randomforestclassifier']

    # Print the full parameters of the XGBoost model
    print("Best XGBoost Parameters:")
    pprint(xgb_model.get_params())
    print()
```

```
Best XGBoost Parameters:
{'bootstrap': True,
 'ccp_alpha': 0.0,
 'class_weight': 'balanced',
 'criterion': 'gini',
 'max_depth': 5,
 'max_features': 'log2',
 'max_leaf_nodes': None,
 'max_samples': 0.75,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'monotonic_cst': None,
 'n_estimators': 1000,
 'n_jobs': -1,
 'oob_score': False,
 'random_state': 42,
 'verbose': 0,
 'warm_start': False}

Best XGBoost Parameters:
{'bootstrap': True,
 'ccp_alpha': 0.0,
 'class_weight': 'balanced',
 'criterion': 'gini',
 'max_depth': 5,
 'max_features': 'log2',
 'max_leaf_nodes': None,
 'max_samples': 0.5,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'monotonic_cst': None,
 'n_estimators': 1000,
 'n_jobs': -1,
 'oob_score': False,
 'random_state': 42,
 'verbose': 0,
 'warm_start': False}

Best XGBoost Parameters:
{'bootstrap': True,
 'ccp_alpha': 0.0,
 'class_weight': 'balanced',
 'criterion': 'gini',
 'max_depth': 5,
```

```
 'max_features': 'sqrt',
 'max_leaf_nodes': None,
 'max_samples': 1.0,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'monotonic_cst': None,
 'n_estimators': 1000,
 'n_jobs': -1,
 'oob_score': False,
 'random_state': 42,
 'verbose': 0,
 'warm_start': False}

Best XGBoost Parameters:
{'bootstrap': True,
 'ccp_alpha': 0.0,
 'class_weight': 'balanced',
 'criterion': 'gini',
 'max_depth': 5,
 'max_features': 'sqrt',
 'max_leaf_nodes': None,
 'max_samples': 0.75,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'monotonic_cst': None,
 'n_estimators': 1000,
 'n_jobs': -1,
 'oob_score': False,
 'random_state': 42,
 'verbose': 0,
 'warm_start': False}

Best XGBoost Parameters:
{'bootstrap': True,
 'ccp_alpha': 0.0,
 'class_weight': 'balanced',
 'criterion': 'gini',
 'max_depth': 5,
 'max_features': 0.5,
 'max_leaf_nodes': None,
 'max_samples': 0.5,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
```

```
 'min_weight_fraction_leaf': 0.0,
 'monotonic_cst': None,
 'n_estimators': 1000,
 'n_jobs': -1,
 'oob_score': False,
 'random_state': 42,
 'verbose': 0,
 'warm_start': False}

Best XGBoost Parameters:
{'bootstrap': True,
 'ccp_alpha': 0.0,
 'class_weight': 'balanced',
 'criterion': 'gini',
 'max_depth': 5,
 'max_features': 0.5,
 'max_leaf_nodes': None,
 'max_samples': 1.0,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'monotonic_cst': None,
 'n_estimators': 1000,
 'n_jobs': -1,
 'oob_score': False,
 'random_state': 42,
 'verbose': 0,
 'warm_start': False}

Best XGBoost Parameters:
{'bootstrap': True,
 'ccp_alpha': 0.0,
 'class_weight': 'balanced',
 'criterion': 'gini',
 'max_depth': 5,
 'max_features': 0.5,
 'max_leaf_nodes': None,
 'max_samples': 1.0,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'monotonic_cst': None,
 'n_estimators': 1000,
 'n_jobs': -1,
 'oob_score': False,
 'random_state': 42,
```

```
 'verbose': 0,
 'warm_start': False}

Best XGBoost Parameters:
{'bootstrap': True,
 'ccp_alpha': 0.0,
 'class_weight': 'balanced',
 'criterion': 'gini',
 'max_depth': 1,
 'max_features': 'log2',
 'max_leaf_nodes': None,
 'max_samples': 0.75,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'monotonic_cst': None,
 'n_estimators': 1000,
 'n_jobs': -1,
 'oob_score': False,
 'random_state': 42,
 'verbose': 0,
 'warm_start': False}

Best XGBoost Parameters:
{'bootstrap': True,
 'ccp_alpha': 0.0,
 'class_weight': 'balanced',
 'criterion': 'gini',
 'max_depth': 5,
 'max_features': 'sqrt',
 'max_leaf_nodes': None,
 'max_samples': 0.75,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'monotonic_cst': None,
 'n_estimators': 1000,
 'n_jobs': -1,
 'oob_score': False,
 'random_state': 42,
 'verbose': 0,
 'warm_start': False}

Best XGBoost Parameters:
{'bootstrap': True,
 'ccp_alpha': 0.0,
```

```
 'class_weight': 'balanced',
 'criterion': 'gini',
 'max_depth': 5,
 'max_features': 0.5,
 'max_leaf_nodes': None,
 'max_samples': 0.75,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'monotonic_cst': None,
 'n_estimators': 1000,
 'n_jobs': -1,
 'oob_score': False,
 'random_state': 42,
 'verbose': 0,
 'warm_start': False}
```

[31]:
```
import os
cwd = os.getcwd()
print("Current Working Directory:", cwd)
```

```
Current Working Directory: /Users/fruit/Desktop/DATA1030-Fall2024/Key-Resume-
Attributes-Impacting-Job-Callbacks/src
```