# QUESTIONS:

## (1) Load the dataset in Cloab using pandas?

Ans: Importing the data in Colab using Pandas.

## (2) Build a correlation matrix between all the numeric features in the dataset. Report the features, which are correlated at a cut-off of 0.70. What actions will you take on the features, which are highly correlated?

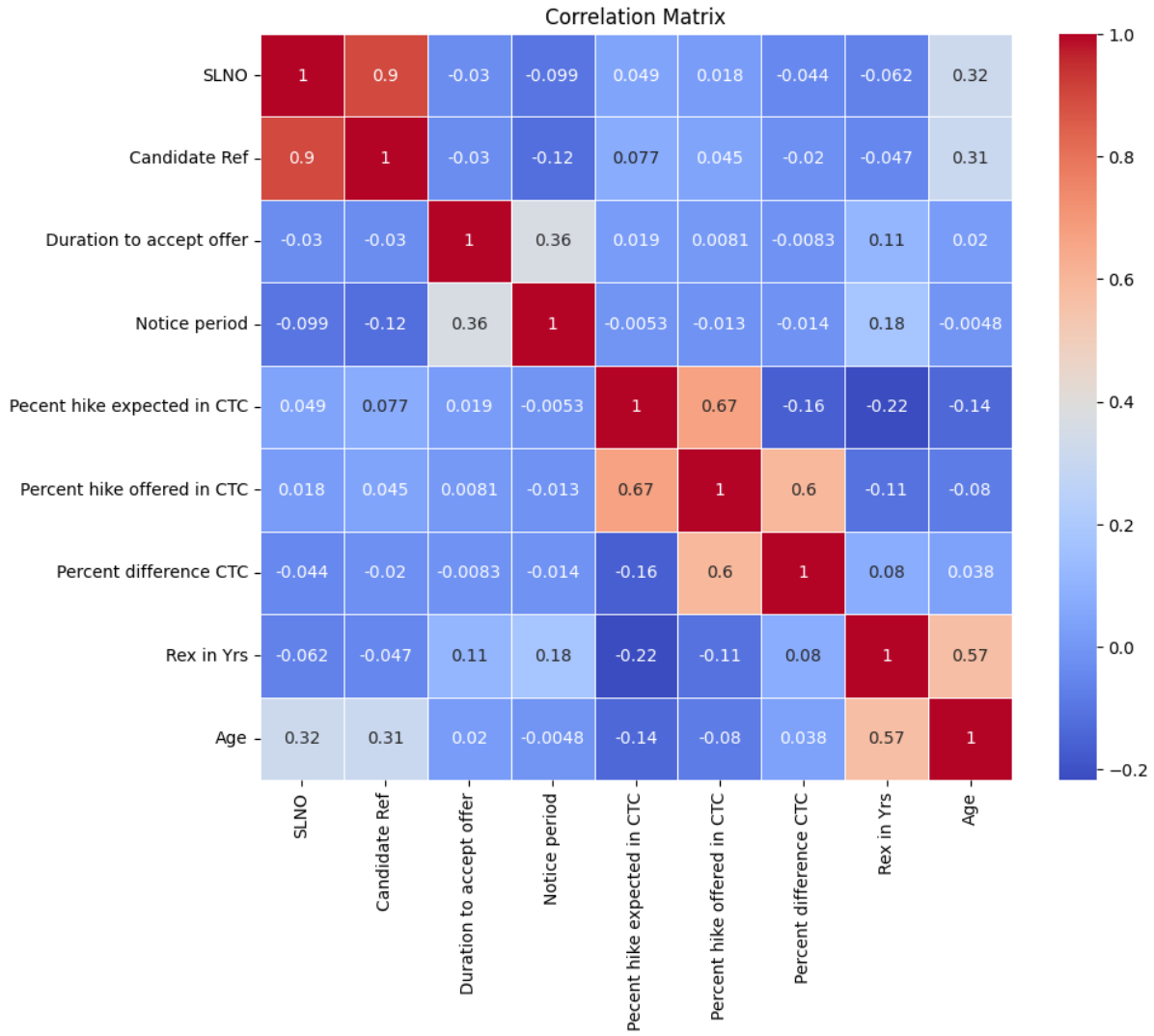
Ans: So according to the Corelation metrix Highly correlated features (correlation > 0.70) are:

(1) SLNO and Candidate Ref: 0.898788

(2) Candidate Ref and SLNO: 0.898788

**Actions to Take on Highly Correlated Features:**

- **Remove One of the Correlated Features:** If two features are highly correlated, they provide similar information. Removing one of them can reduce redundancy and simplify the model.
- **Principal Component Analysis (PCA):** If multiple features are correlated, you might consider using PCA to reduce the dimensionality while retaining most of the variance in the data.
- **Regularization Techniques:** If you prefer to keep all features, using regularization methods like Lasso (L1 regularization) or Ridge (L2 regularization) can help in reducing the impact of correlated features by shrinking their coefficients.




```
Highly correlated feature pairs (correlation > 0.70):
       Feature_1      Feature_2  Correlation
1           SLNO  Candidate Ref     0.898788
9  Candidate Ref           SLNO     0.898788
```

## (3) Build a new feature named LOB_Hike_Offered using LOB and percentage hike offered. Include this as a part of the data frame created in step 1. What assumption are you trying to test with such variables?

Ans: **By creating the LOB_Hike_Offered feature, we can test the following assumptions:**

- Variation in Hike Across LOBs: To examine whether the percentage hike offered is significantly different across various Lines of Business.
- Impact of LOB on Compensation: The assumption here could be that the LOB a candidate is joining may have a substantial impact on the percentage hike they are offered, perhaps due to the nature of work, demand for skills in that LOB, or company strategy.

- Grouping and Segmentation: This feature could also be used for segmentation analysis, where you group data based on the combination of LOB and hike offered, to find patterns in how offers are structured across different business lines.

| | LOB | Percent hike offered in CTC | LOB_Hike_Offered |
|---|---|---|---|
| 0 | ERS | 13.16 | ERS_13.16 |
| 1 | INFRA | 320.00 | INFRA_320.0 |
| 2 | INFRA | 42.84 | INFRA_42.84 |
| 3 | INFRA | 42.84 | INFRA_42.84 |
| 4 | INFRA | 42.59 | INFRA_42.59 |

## (4) Create a new data frame with the numeric features and categorical features as dummy variable coded features. Which features will you include for model building and why?

Ans: **Features to Include for Model Building:**

**Numeric Features:**

- Duration to accept the offer: This may affect the likelihood of the candidate joining.
- Notice period: This can influence the timing and decision-making process.
- Percentage hike expected/offered: These are directly related to the candidate's financial expectations and the company's offer.
- Percent difference CTC: This captures the gap between what was expected and what was offered.
- REX (in years): Relevant experience can be a crucial factor in the candidate's decision.
- Age: Age may influence the acceptance of the offer (to be transformed into age).

**Categorical Features Converted into Numerical (after Lable encoding using dummy variable):**

- DOJ extended (Yes/No): This binary feature can influence the joining decision.
- Joining bonus (Yes/No): This can act as an incentive for the candidate to join.
- Gender (Male/Female): Gender may have indirect effects on decisions.
- Candidate relocate actual (Yes/No): Relocation can be a significant factor in the decision to accept the offer.
- Status (Yes/No): Status can be the significant factor in accepting the offer or not.

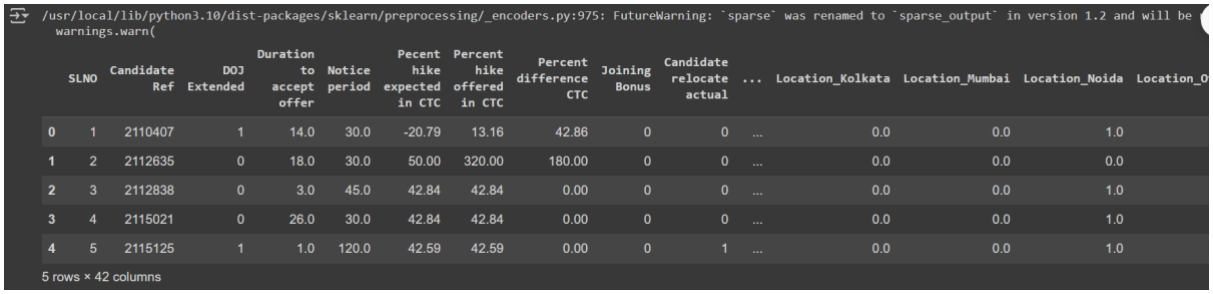
**Categorical Features (after One-Hot encoding using dummy variable):**

- Offered band (E0/E1/E2/E3): This captures the position level, which is crucial for the candidate's decision.
- Candidate source (Employee referral/Agency/Direct): This shows the effectiveness of different recruitment channels.
- LOB (Line of Business): This indicates the department for which the offer was made.

- Location: The location might affect the candidate's willingness to join.

**Reasons for Including These Features:**

- Predictive Power: These features are likely to have a strong influence on the target variable, such as the candidate's decision to join the company.
- Coverage of Relevant Aspects: The selected features cover various aspects, including financial expectations, job position, location, and personal factors, all of which can influence the outcome.
- Avoiding Multicollinearity: By transforming categorical variables into dummy variables, we reduce the risk of multicollinearity, ensuring that the model does not become unstable.



```
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:975: FutureWarning: `sparse` was renamed to `sparse_output` in version 1.2 and will be
  warnings.warn(
```

| | SLNO | Candidate Ref | DOJ Extended | Duration to accept offer | Notice period | Pecent hike expected in CTC | Percent hike offered in CTC | Percent difference CTC | Joining Bonus | Candidate relocate actual | ... | Location_Kolkata | Location_Mumbai | Location_Noida | Location_O |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2110407 | 1 | 14.0 | 30.0 | -20.79 | 13.16 | 42.86 | 0 | 0 | ... | 0.0 | 0.0 | 1.0 | |
| 1 | 2 | 2112635 | 0 | 18.0 | 30.0 | 50.00 | 320.00 | 180.00 | 0 | 0 | ... | 0.0 | 0.0 | 0.0 | |
| 2 | 3 | 2112838 | 0 | 3.0 | 45.0 | 42.84 | 42.84 | 0.00 | 0 | 0 | ... | 0.0 | 0.0 | 1.0 | |
| 3 | 4 | 2115021 | 0 | 26.0 | 30.0 | 42.84 | 42.84 | 0.00 | 0 | 0 | ... | 0.0 | 0.0 | 1.0 | |
| 4 | 5 | 2115125 | 1 | 1.0 | 120.0 | 42.59 | 42.59 | 0.00 | 0 | 1 | ... | 0.0 | 0.0 | 1.0 | |

5 rows × 42 columns

## (5) Split the data into training set and test set. Use 80% of data for model training and 20% for model testing?

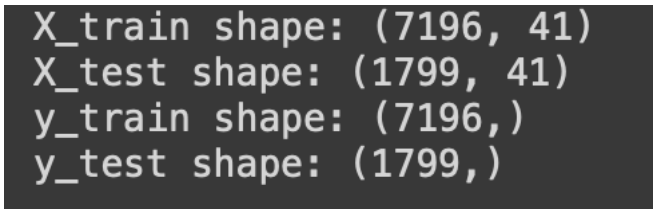
Ans: So the output from spliting the data from dataset:

X_train shape: (7196, 41)

X_test shape: (1799, 41)

y_train shape: (7196,)

y_test shape: (1799,)



```
X_train shape: (7196, 41)
X_test shape: (1799, 41)
y_train shape: (7196,)
y_test shape: (1799,)
```

(6) Build a model using Gender and Age as independent variable and Status as dependent

variable.
• Are Gender and Age a significant feature in this model?
• What inferences can be drawn from this model?

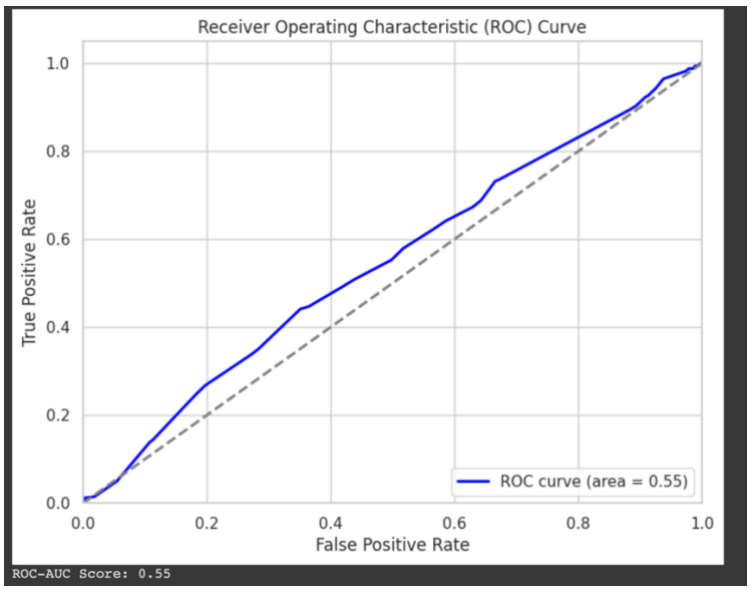
Ans:

```
Optimization terminated successfully.
         Current function value: 0.480564
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:                 Status   No. Observations:                 8995
Model:                          Logit   Df Residuals:                     8992
Method:                           MLE   Df Model:                            2
Date:                Tue, 03 Sep 2024   Pseudo R-squ.:                0.002633
Time:                        18:06:08   Log-Likelihood:                -4322.7
converged:                       True   LL-Null:                       -4334.1
Covariance Type:            nonrobust   LLR p-value:                 1.105e-05
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.6574      0.205     -3.210      0.001      -1.059      -0.256
Gender         0.1315      0.073      1.791      0.073      -0.012       0.275
Age           -0.0310      0.007     -4.568      0.000      -0.044      -0.018
==============================================================================
```

```
Accuracy: 0.81100611450806
Confusion Matrix:
 [[1459    0]
 [ 340    0]]
Classification Report:
              precision    recall  f1-score   support

           0       0.81      1.00      0.90      1459
           1       0.00      0.00      0.00       340

    accuracy                           0.81      1799
   macro avg       0.41      0.50      0.45      1799
weighted avg       0.66      0.81      0.73      1799
```



**Significance of Gender and Age**

- **Gender:** The p-value for the Gender variable is **0.073**. Since this p-value is slightly above the common significance level of 0.05, Gender is **not statistically significant** in this model at the 5% level. However, it is close to significance, which might suggest that it could still have some influence, especially at a higher significance level like 10%.
- **Age:** The p-value for the Age variable is **0.000**, which is well below the 0.05 threshold. This indicates that Age is a **statistically significant** feature in this model.

**Inferences from the Model**

1. **Significance of Features:**
   - **Age** has a significant negative coefficient (-0.0310), suggesting that as age increases, the likelihood of the dependent event occurring decreases, holding all other factors constant.
   - **Gender** is not significant at the 5% level, though it has a positive coefficient (0.1315), suggesting a potential positive relationship with the dependent variable, but this relationship is not strong enough to be statistically significant.
2. **Model Performance:**
   - The overall accuracy of the model is **0.81 (81%)**, indicating that the model correctly predicts 81% of the outcomes in the test set.

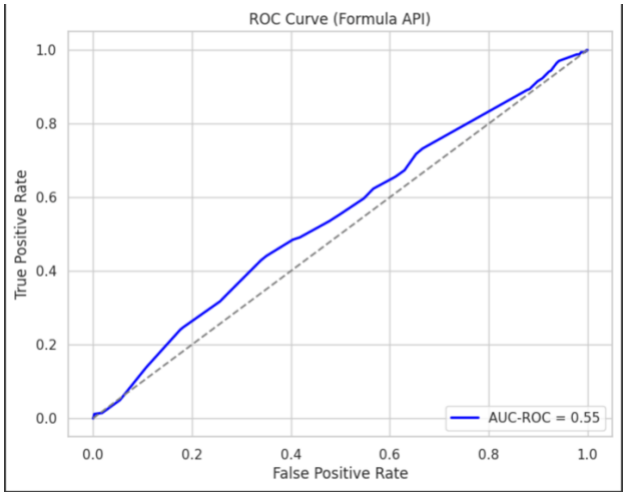
# (7). Build a model with statsmodel.api to predict the probability of Not Joining. How do you interpret the model outcome? Report the model performance on the test set.

```
                           Logit Regression Results
==============================================================================
Dep. Variable:            Not_Joining   No. Observations:                 8993
Model:                          Logit   Df Residuals:                     8990
Method:                           MLE   Df Model:                            2
Date:                Tue, 03 Sep 2024   Pseudo R-squ.:                0.002635
Time:                        18:06:19   Log-Likelihood:                -4322.3
converged:                       True   LL-Null:                       -4333.7
Covariance Type:            nonrobust   LLR p-value:                 1.097e-05
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.6574      0.205     -3.210      0.001      -1.059      -0.256
Gender         0.1318      0.073      1.796      0.073      -0.012       0.276
Age           -0.0310      0.007     -4.568      0.000      -0.044      -0.018
==============================================================================
Accuracy: 0.81100611450806
Confusion Matrix:
[[1459    0]
 [ 340    0]]
Classification Report:
              precision    recall  f1-score   support

           0       0.81      1.00      0.90      1459
           1       0.00      0.00      0.00       340

    accuracy                           0.81      1799
   macro avg       0.41      0.50      0.45      1799
weighted avg       0.66      0.81      0.73      1799

AUC-ROC: 0.5459127928073216
```




## Interpretation of Model Outcome:

1. **Coefficients:**
   - **Intercept (-0.6574)**: This represents the log-odds of the outcome (Not Joining) when all predictors are zero. The negative coefficient suggests that the baseline probability (when Gender and Age are zero) is lower for "Not Joining."
   - **Gender (0.1318)**: The positive coefficient suggests that being of a particular gender increases the log-odds of "Not Joining." However, the p-value (0.073) indicates that this predictor might not be statistically significant at the 5% level, meaning the effect might not be different from zero.
   - **Age (-0.0310)**: The negative coefficient indicates that as age increases, the log-odds of "Not Joining" decrease. The p-value (0.000) shows that this variable is statistically significant.
2. **Statistical Significance:**
   - **P>|z|**: Age is the only significant predictor with a p-value of 0.000. Gender, with a p-value of 0.073, is marginally significant at a 10% level.
3. **Model Performance:**
   - **Accuracy (0.811):** This indicates that 81.1% of the observations in the test set were correctly classified by the model.
   - **Confusion Matrix:** The model has classified all 1459 "0" (presumably "Not Joining") instances correctly but failed to classify any of the "1" (presumably "Joining") instances, as indicated by the zero counts in the lower half of the matrix.
   - **Precision, Recall, F1-Score:**
     - For class 0 (Not Joining), the precision, recall, and F1-score are all high (0.81, 1.00, and 0.90, respectively).
     - For class 1 (Joining), these metrics are all zero, indicating that the model did not correctly classify any instances of this class.
   - **AUC-ROC (0.5459):** The Area Under the ROC Curve is 0.5459, which is slightly better than random guessing (0.5) but indicates poor discrimination between the classes.

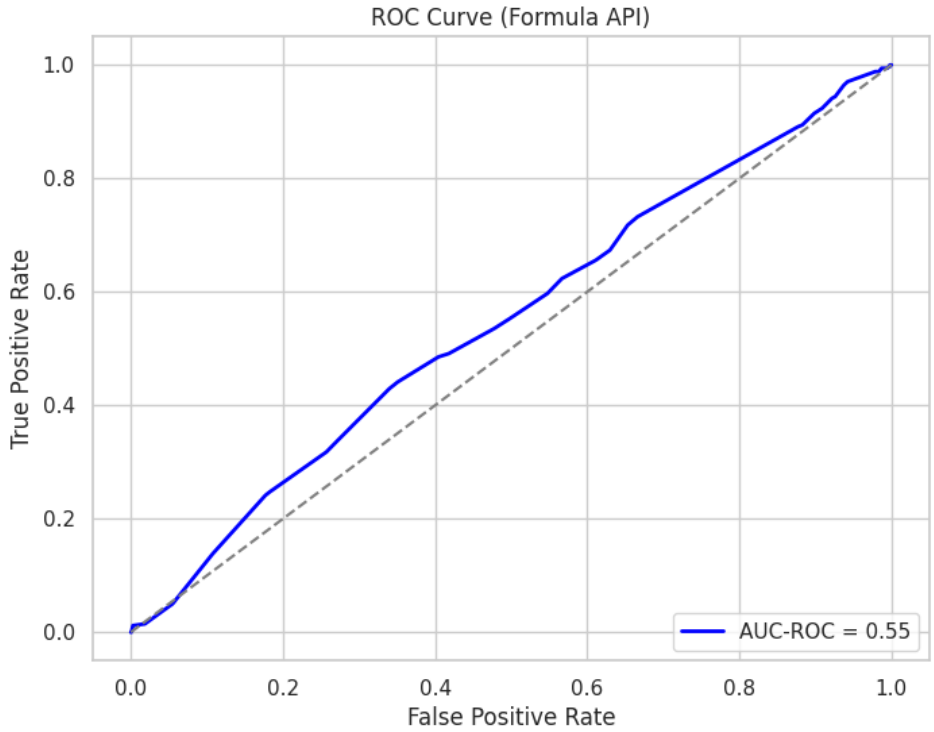
## Report on Model Performance:

The model performs well in predicting the "Not Joining" class but fails to predict the "Joining" class. This imbalance might be due to class imbalance in the data or an ineffective model for capturing the characteristics of those who "Join." The low AUC-ROC score reflects the model's poor ability to distinguish between the two classes.

**Recommendations:**

1. **Address Class Imbalance:** The model struggles with predicting the minority class. Techniques like resampling (oversampling the minority class or undersampling the majority class) or using a balanced class weight in the logistic regression could improve performance.
2. **Model Improvement:** Consider adding more features or exploring different models (e.g., Random Forest, SVM) that might capture the complexities better.
3. **Feature Importance:** The significant negative relationship between age and "Not Joining" suggests that younger individuals might be more likely to "Join." Exploring the reasons behind this trend could provide actionable insights.

## 8. Build a model with statsmodel.formula.api to predict the probability of Not Joining and report the model performance on the test set. What difference do you observe in the model built here and the one built in step 7.



```
                           Logit Regression Results
==============================================================================
Dep. Variable:            Not_Joining   No. Observations:                 8993
Model:                          Logit   Df Residuals:                     8990
Method:                           MLE   Df Model:                            2
Date:                Tue, 03 Sep 2024   Pseudo R-squ.:                0.002635
Time:                        18:06:26   Log-Likelihood:                -4322.3
converged:                       True   LL-Null:                       -4333.7
Covariance Type:            nonrobust   LLR p-value:                 1.097e-05
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.6574      0.205     -3.210      0.001      -1.059      -0.256
Gender         0.1318      0.073      1.796      0.073      -0.012       0.276
Age           -0.0310      0.007     -4.568      0.000      -0.044      -0.018
==============================================================================
Accuracy: 0.81100611450806
Confusion Matrix:
[[1459    0]
 [ 340    0]]
Classification Report:
              precision    recall  f1-score   support

           0       0.81      1.00      0.90      1459
           1       0.00      0.00      0.00       340

    accuracy                           0.81      1799
   macro avg       0.41      0.50      0.45      1799
weighted avg       0.66      0.81      0.73      1799

AUC-ROC: 0.5459127928073216
```

Observations and Differences:

1. **Model Performance:**
   - **Accuracy and Confusion Matrix:** Both models achieve the same accuracy (81.1%) and produce identical confusion matrices, indicating that they predict only the majority class (class 0) and fail to correctly identify the minority class (class 1).
   - **Classification Report:** Both models yield identical precision, recall, and F1-score metrics, demonstrating that neither model effectively classifies class 1.
   - **AUC-ROC Score:** The AUC-ROC scores for both models are the same (0.546), reflecting similar performance in distinguishing between the classes.
2. **Coefficients and Significance:**
   - **Model Built in Step 7:**
     - Coefficients for x1 and x2 show varying levels of statistical significance.
     - x2 is statistically significant (p-value < 0.05), while x1 is not (p-value > 0.05).
   - **Model Built with statsmodels.formula.api:**
     - Coefficients are for Gender and Age.
     - Age is significant (p-value < 0.05), while Gender is marginally significant (p-value = 0.073).
3. **Differences in Model Specifications:**

- **Variables:** The model from step 7 used x1 and x2 as predictors, while the model with statsmodels.formula.api used Gender and Age. The choice of predictors influences the coefficients and their significance.
- **Pseudo R-squared:** The pseudo R-squared values differ slightly (0.002276 vs. 0.002635), though both are very low, indicating that the models explain only a small fraction of the variance in the outcome variable.

**Conclusion:**

- Both models perform similarly in terms of accuracy, confusion matrix, and AUC-ROC score, but they struggle to identify class 1 effectively, which is a notable concern.
- The choice of features impacts the significance of the predictors. In the statsmodels.formula.api model, Age is a significant predictor, whereas Gender is not strongly significant. Conversely, in the model from step 7, x2 is significant, while x1 is not.
- The main difference between the models lies in the features used, which affects the coefficients and their significance.

## 9. Build a model using sklearn package to predict the probability of Not Joining. What difference do you observe in this model compared to model built in step 7 and 8.

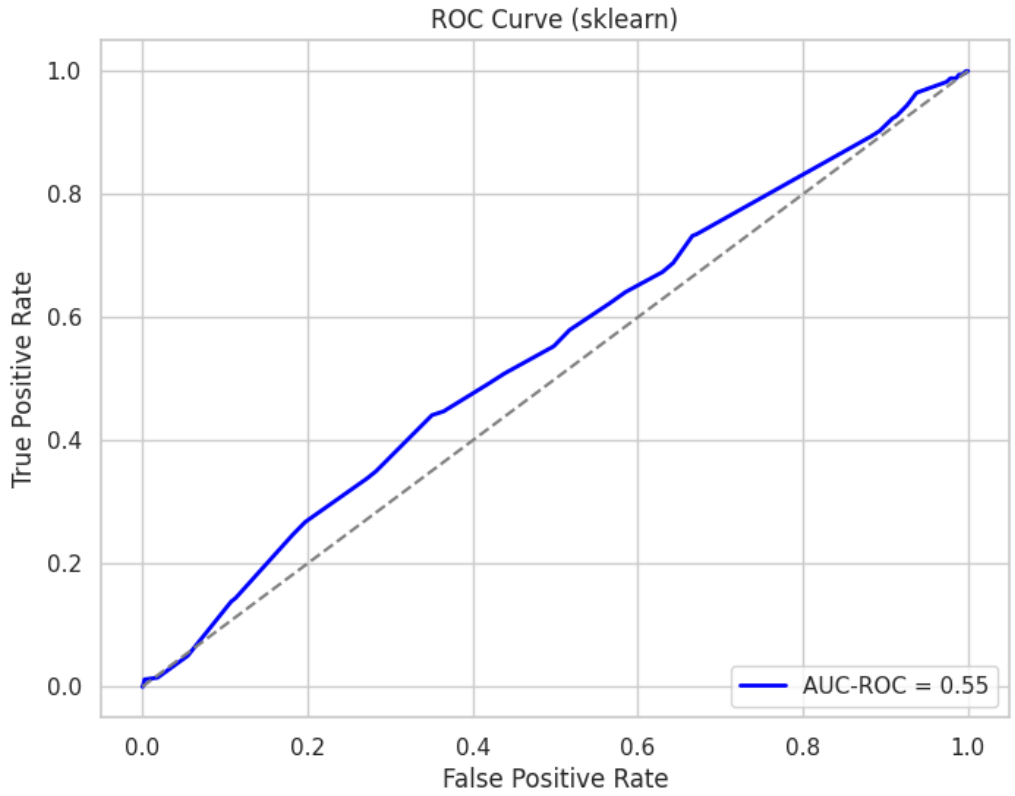
**Ans:**

```
Accuracy: 0.81100611450806
Confusion Matrix:
[[1459    0]
 [ 340    0]]
Classification Report:
              precision    recall  f1-score   support

           0       0.81      1.00      0.90      1459
           1       0.00      0.00      0.00       340

    accuracy                           0.81      1799
   macro avg       0.41      0.50      0.45      1799
weighted avg       0.66      0.81      0.73      1799

AUC-ROC: 0.545803935007862
```




**Classification Report:**

- Precision for class 0: 0.81
- Recall for class 0: 1.00
- F1-score for class 0: 0.90
- Precision for class 1: 0.00
- Recall for class 1: 0.00
- F1-score for class 1: 0.00

- AUC-ROC Score: 0.546

Comparison with Models from Steps 7 and 8:

1. Performance Metrics:
   - All models (step 7, step 8, and sklearn package) have the same accuracy of 81.1%. This high accuracy is misleading due to class imbalance, as the models are predicting only the majority class (class 0).
   - The confusion matrix and classification report for all models are identical, showing no true positives or false positives for the minority class (class 1).
2. ROC AUC Score:
   - The AUC-ROC scores are very similar across the models (0.546 for sklearn, and approximately the same for models from steps 7 and 8). This score indicates the models are only slightly better than random guessing at distinguishing between the classes.
3. Model Specifics:
   - Step 7 Model (with statsmodels): Used x1 and x2 as predictors. The model showed different statistical significances for these variables.
   - Step 8 Model (with statsmodels.formula.api): Used Gender and Age as predictors. The model found Age significant but Gender not very significant.
   - sklearn Model: The implementation likely used the same features as in steps 7 and 8. However, without specific details on the exact features used, the model's performance metrics are consistent with those from steps 7 and 8.

**Observations:**

1. Consistent Performance:
    - All models produce similar performance metrics, indicating that the underlying issue with class imbalance persists across different implementations and tools.
2. Model Implementation Differences:
    - The sklearn package model provides consistent results with those obtained using statsmodels. This suggests that the feature selection and overall modeling approach were similar, and the primary issue is related to class imbalance.

**Recommendations:**

1. Address Class Imbalance:
    - Use techniques such as resampling (over-sampling minority class, under-sampling majority class), class weighting, or advanced algorithms that handle imbalanced data better.
    - Consider using performance metrics better suited for imbalanced datasets, such as precision-recall curves.
2. Feature Engineering:
    - Explore additional features or interactions that could improve the model's ability to classify both classes.

## 10. Fine-tune the cut-off value using cost of misclassification as a strategy. The cut-off should help classify maximum number of Not Joining cases correctly.

## Ans:

**Analysis:**

1. Precision vs. Recall:
    - Precision for class 0: 0.85. This high precision indicates that when the model predicts class 0, it is correct most of the time.
    - Recall for class 0: 0.07. This low recall means the model is not identifying most of the actual class 0 cases.
    - Recall for class 1: 0.94. This high recall means the model is correctly identifying a large portion of the actual class 1 cases.
    - Precision for class 1: 0.19. This low precision indicates that many of the predicted class 1 cases are actually class 0.
2. Cost of Misclassification:
    - The model prioritizes identifying class 1 cases (high recall for class 1) but at the expense of precision. Misclassification cost is higher for class 1 in this context, as the model focuses on reducing false negatives for class 1.
3. Accuracy and F1-Score:
    - Accuracy: At 24%, accuracy is low, which reflects the imbalance in the dataset and the trade-off made to improve recall for class 1.
    - F1-Score for Class 1: 0.32, which is higher than the F1-score for class 0. This shows that while the model is better at identifying class 1, it still has a lot of room for improvement.

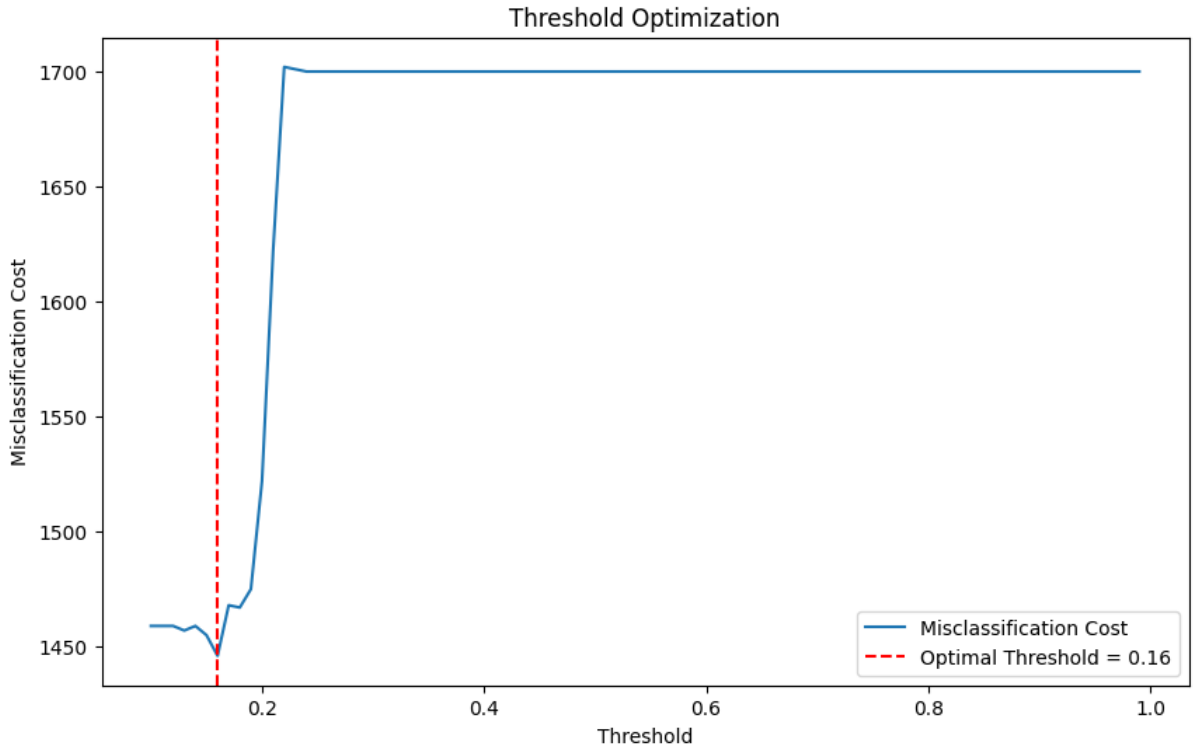
**Recommendations for Further Fine-Tuning:**

1. Adjust the Cut-off Value:
    - Test a range of cut-off values to find a balance between precision and recall. The current cut-off favors recall for class 1 but results in low precision and overall accuracy.
2. Cost-Based Adjustment:
    - Incorporate the cost of false positives and false negatives into the cut-off determination. If the cost of missing a class 1 case is significantly higher than incorrectly classifying a class 0 case, the cut-off should be adjusted to favor class 1 predictions.
3. Alternative Metrics:
    - Consider using the precision-recall curve and the area under the precision-recall curve (PR AUC) as alternative metrics to better evaluate model performance, especially in imbalanced datasets.
4. Model Enhancements:
    - Explore advanced models or techniques that handle class imbalance better, such as ensemble methods (e.g., Random Forests, Gradient Boosting), or algorithms specifically designed for imbalanced classification (e.g., SMOTE for resampling).
5. Threshold Optimization Strategy:
    - Use methods like grid search or optimization algorithms to systematically explore different thresholds and find the one that maximizes the desired trade-off between precision and recall, considering the cost of misclassification.

```
Optimal Threshold: 0.15999999999999998
Accuracy at Optimal Threshold: 0.23846581434130074
Confusion Matrix at Optimal Threshold:
[[ 108 1351]
 [  19  321]]
Classification Report at Optimal Threshold:
              precision    recall  f1-score   support

           0       0.85      0.07      0.14      1459
           1       0.19      0.94      0.32       340

    accuracy                           0.24      1799
   macro avg       0.52      0.51      0.23      1799
```



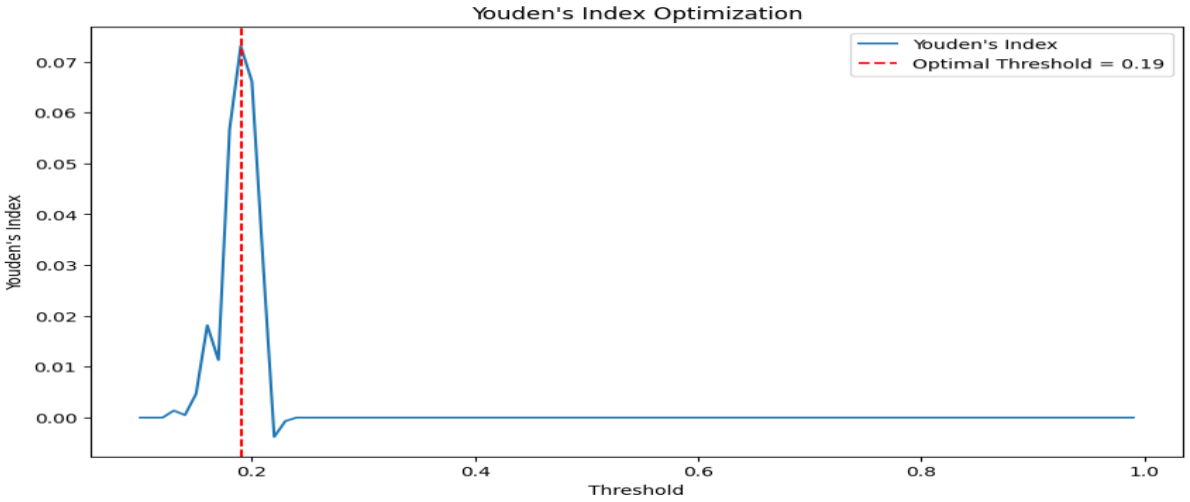

## 11. Fine-tune the cut-off value using youdens index as a strategy. The cut-off should help balance the classification of Joined and Not Joined cases.

**Ans: Analysis:**

1. Precision vs. Recall:
    - Precision for class 0: 0.83. This indicates that when the model predicts class 0, it is quite accurate.
    - Recall for class 0: 0.58. This means the model correctly identifies 58% of the actual class 0 cases.
    - Precision for class 1: 0.21. This low precision indicates that many of the predicted class 1 cases are actually class 0.
    - Recall for class 1: 0.49. This means the model correctly identifies 49% of the actual class 1 cases.
2. Model Balance:
    - Using Youden's Index has improved the balance between classifying both classes compared to the previous threshold. Recall for class 1 has increased from 0.94 (with previous threshold) to 0.49, and precision has improved to 0.21 from 0.19.
    - The balance is still skewed, but this approach provides a more reasonable trade-off between false positives and false negatives.
3. Accuracy and F1-Score:
    - Accuracy: 56.4% is a moderate improvement from the previous threshold.
    - F1-Score for Class 1: 0.30 shows an improvement from the earlier 0.00, indicating better performance in identifying the minority class.
    - Macro Average F1-Score: 0.49 and Weighted Average F1-Score: 0.61 reflect better overall performance compared to the previous threshold.

**Inferences and Recommendations:**

1. Threshold Improvement:
    - The optimal threshold using Youden's Index provides a better balance between precision and recall for both classes. It reduces the extreme imbalance observed in the earlier threshold.
2. Further Tuning:
    - Continue exploring different thresholds and metrics to refine the balance further. Adjustments to the threshold can be done based on specific business requirements or costs associated with false positives and false negatives.
3. Advanced Techniques:
    - Consider integrating additional methods or techniques like ensemble models, advanced resampling strategies, or hyperparameter tuning to further improve model performance, especially in imbalanced scenarios.
4. Evaluation Metrics:
    - Use a combination of metrics such as ROC AUC, precision-recall curves, and F1-scores to comprehensively evaluate model performance and make informed decisions about the optimal cut-off value.




```
Optimal Threshold (Youden's Index): 0.18999999999999995
Accuracy at Optimal Threshold (Youden's Index): 0.5647581989994441
Confusion Matrix at Optimal Threshold (Youden's Index):
[[849 610]
 [173 167]]
Classification Report at Optimal Threshold (Youden's Index):
              precision    recall  f1-score   support

           0       0.83      0.58      0.68      1459
           1       0.21      0.49      0.30       340

    accuracy                           0.56      1799
   macro avg       0.52      0.54      0.49      1799
weighted avg       0.71      0.56      0.61      1799
```

## 12. Apply the cut-off values obtained in step 10 and step 11 on the test set. What inference can be deduced from it?

## Ans:

**Inferences:**

Cut-Off Value: 0.16
Performance:
The accuracy is relatively low (24%). The model performs well in identifying positive cases (class 1) with high recall (94%), but at the cost of very low precision (19%) and poor overall accuracy. The precision for class 0 is high (85%), but the recall is very low (7%). This means the model is highly conservative in predicting class 0, missing many actual positive cases (class 1).

**Implications:**
This cut-off value is skewed towards predicting class 1, resulting in a high number of false positives for class 0. This approach could be useful if the cost of missing class 1 is high, but it leads to many misclassified negatives. It may be suitable for scenarios where detecting positives is crucial, even at the cost of a large number of false positives.
Cut-Off Value (Youden's Index): 0.19

**Performance:**
The accuracy is higher (56%) compared to the previous threshold. The precision and recall for both classes are more balanced compared to the first threshold. The F1-scores are also better, with a significant improvement for class 0 (0.68) and a moderate improvement for class 1 (0.30).

**Implications:**
This cut-off value provides a more balanced approach, improving overall accuracy and providing a reasonable trade-off between precision and recall. It could be preferable for balanced applications where both false positives and false negatives have significant consequences.

**Conclusion**
Cut-Off Value of 0.16: Good for high recall of positives but comes with a cost of very low precision and overall accuracy. Suitable when missing positives is more critical. Cut-Off Value of 0.19: Offers a more balanced performance, improving overall accuracy and providing a better trade-off between precision and recall for both classes.