pathrise

project

What is problem??

The primary goal of your analysis is to provide insight into whether a colleague will eventually settle at a company

Does an individual with their current circumstances find a job once they join Pathrise or not?We will proceed to address the issue further pathrise

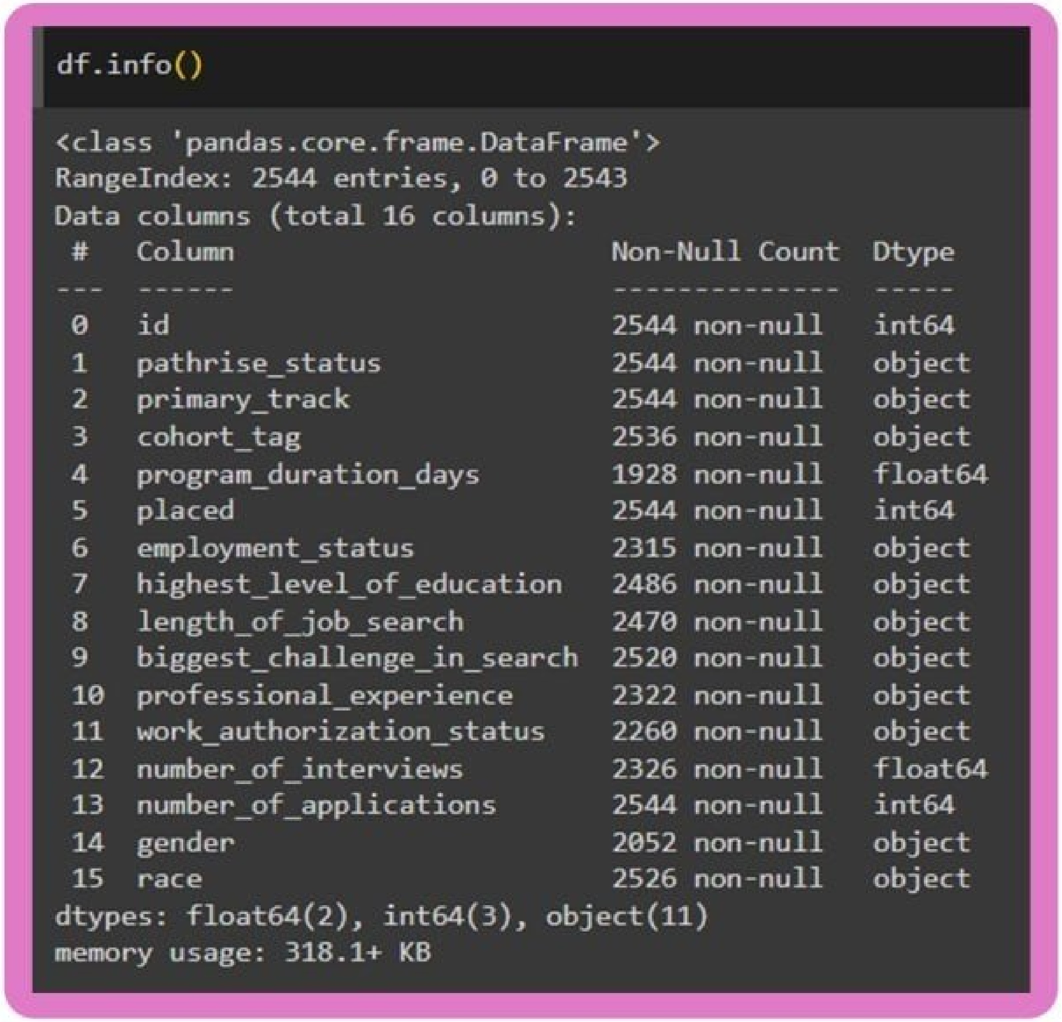
Land your Dream job

Our mentors help you get more interviews and ace them for top companies like Google, Amazon, McKinsey and more

What ‘s pathrise?

Pathrise is a career accelerator that works on your behalf to help you land your next job! We use proprietary tech, data-driven strategies, and one-on-one expert mentorship to provide you with resume, interview, networking, and negotiation support. Our fellows typically experience an increase in interview scores, more job offers, and even increased salaries

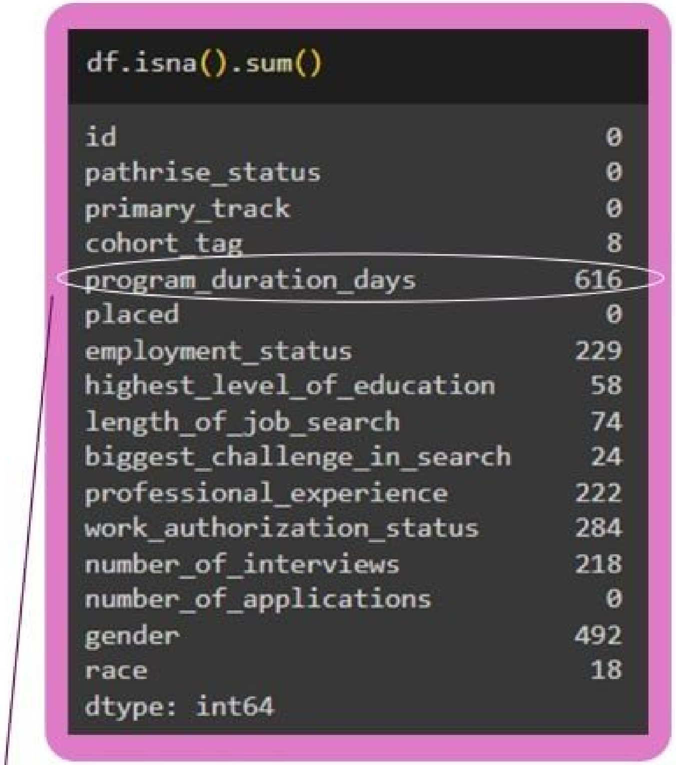
You can see information about the data here



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2544 entries, 0 to 2543
Data columns (total 16 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   id                           2544 non-null   int64
 1   pathrise_status              2544 non-null   object
 2   primary_track                2544 non-null   object
 3   cohort_tag                   2536 non-null   object
 4   program_duration_days        1928 non-null   float64
 5   placed                       2544 non-null   int64
 6   employment_status            2315 non-null   object
 7   highest_level_of_education   2486 non-null   object
 8   length_of_job_search         2470 non-null   object
 9   biggest_challenge_in_search  2520 non-null   object
 10  professional_experience      2322 non-null   object
 11  work_authorization_status    2260 non-null   object
 12  number_of_interviews         2326 non-null   float64
 13  number_of_applications       2544 non-null   int64
 14  gender                       2052 non-null   object
 15  race                         2526 non-null   object
dtypes: float64(2), int64(3), object(11)
memory usage: 318.1+ KB
```

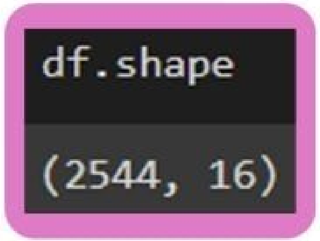
- ● Missing values
- ● object
- ● float64



```
df.isna().sum()
```

```
id                               0
pathrise_status                  0
primary_track                    0
cohort_tag                       8
program_duration_days          616
placed                           0
employment_status              229
highest_level_of_education      58
length_of_job_search            74
biggest_challenge_in_search     24
professional_experience        222
work_authorization_status      284
number_of_interviews           218
number_of_applications           0
gender                         492
race                            18
dtype: int64
```

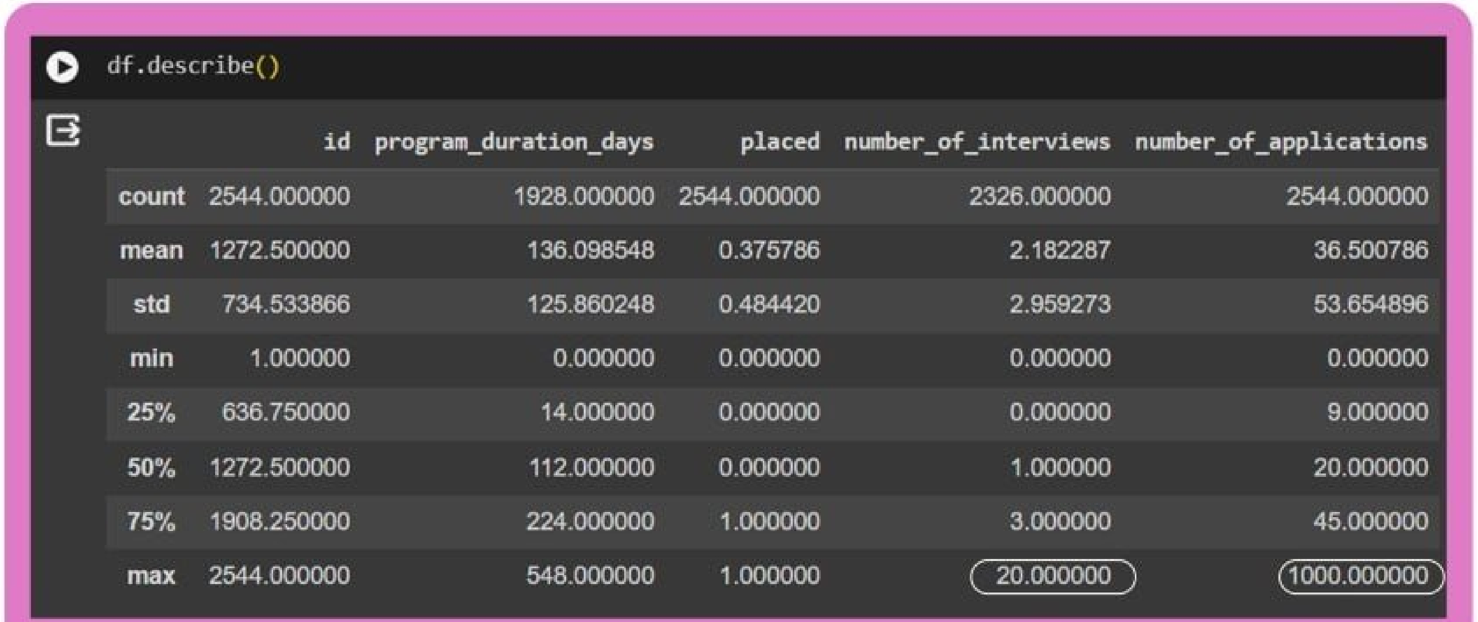
Because this column has a lot of missing values and we can't fill them with wrong information, so we delete the column



```
df.shape
```

```
(2544, 16)
```

Our data has16
lines columns and2544

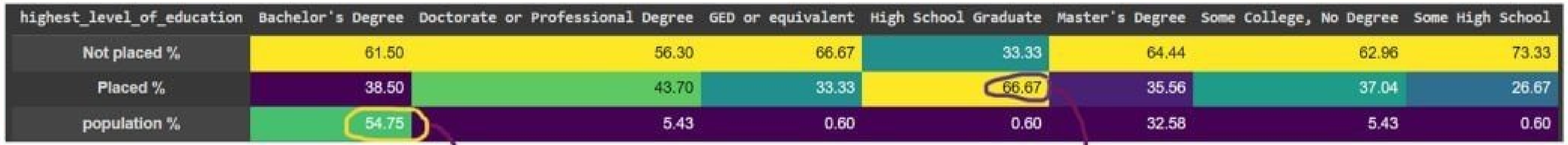


```
df.describe()
```

```
                id  program_duration_days       placed  number_of_interviews  number_of_applications
count  2544.000000            1928.000000  2544.000000           2326.000000             2544.000000
mean   1272.500000             136.098548     0.375786              2.182287               36.500786
std     734.533866             125.860248     0.484420              2.959273               53.654896
min       1.000000               0.000000     0.000000              0.000000                0.000000
25%     636.750000              14.000000     0.000000              0.000000                9.000000
50%    1272.500000             112.000000     0.000000              1.000000               20.000000
75%    1908.250000             224.000000     1.000000              3.000000               45.000000
max    2544.000000             548.000000     1.000000             20.000000             1000.000000
```

These are outliers because the number of interviews is unusual

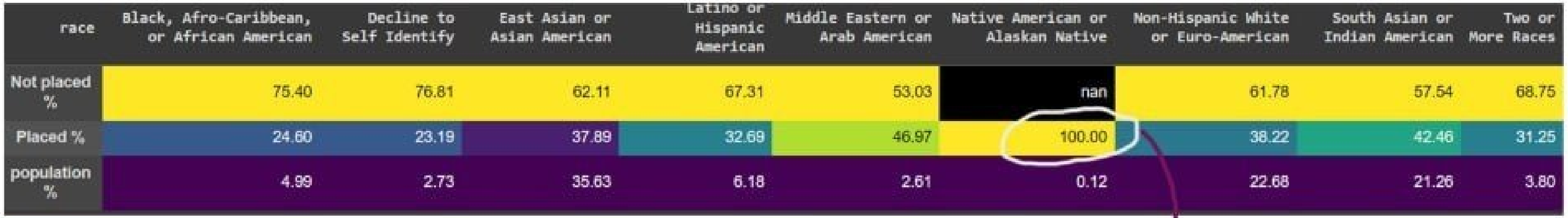
These are outliers because the number of requests is unusual



highest_level_of_education	Bachelor's Degree	Doctorate or Professional Degree	GED or equivalent	High School Graduate	Master's Degree	Some College, No Degree	Some High School
Not placed %	61.50	56.30	66.67	33.33	64.44	62.96	73.33
Placed %	38.50	43.70	33.33	66.67	35.56	37.04	26.67
population %	54.75	5.43	0.60	0.60	32.58	5.43	0.60

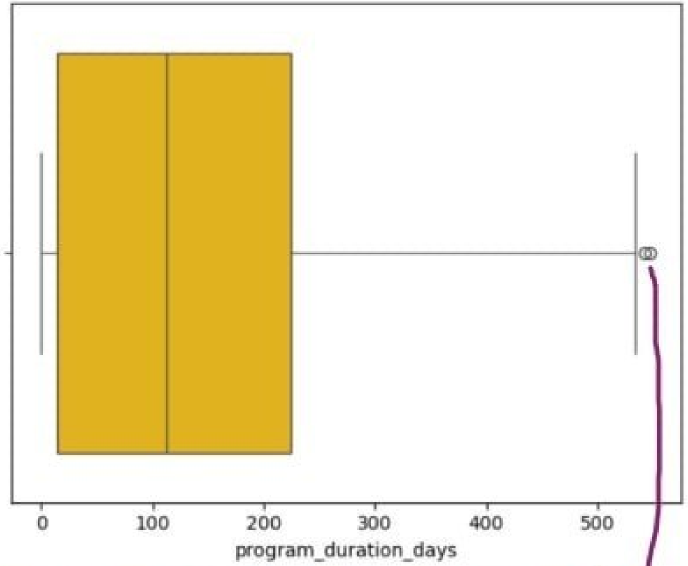
Most input

Entries were few, but most of them got jobs

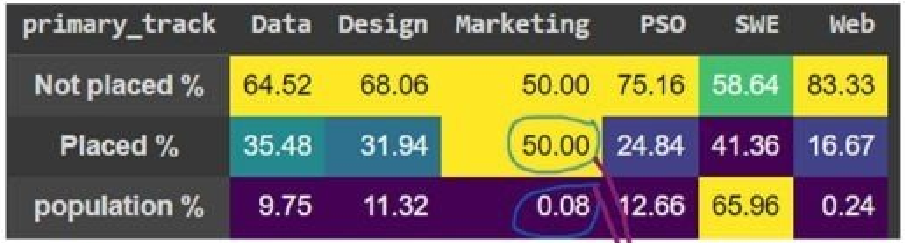


race	Black, Afro-Caribbean, or African American	Decline to Self Identify	East Asian or Asian American	Latino or Hispanic American	Middle Eastern or Arab American	Native American or Alaskan Native	Non-Hispanic White or Euro-American	South Asian or Indian American	Two or More Races
Not placed %	75.40	76.81	62.11	67.31	53.03	nan	61.78	57.54	68.75
Placed %	24.60	23.19	37.89	32.69	46.97	100.00	38.22	42.46	31.25
population %	4.99	2.73	35.63	6.18	2.61	0.12	22.68	21.26	3.80

entries were few, but they all got jobs

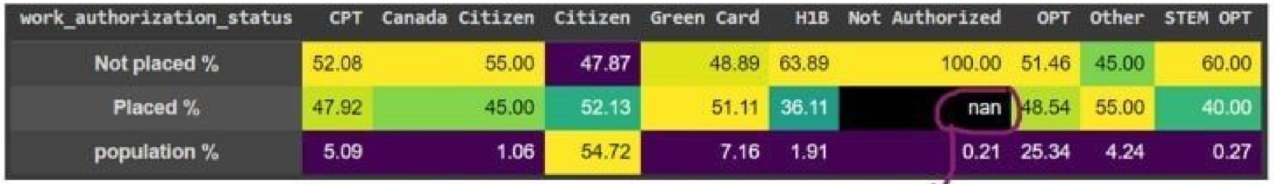


The Path Rise period is one year, but in these amounts, it has increased during the Corona period



primary_track	Data	Design	Marketing	PSO	SWE	Web
Not placed %	64.52	68.06	50.00	75.16	58.64	83.33
Placed %	35.48	31.94	50.00	24.84	41.36	16.67
population %	9.75	11.32	0.08	12.66	65.96	0.24

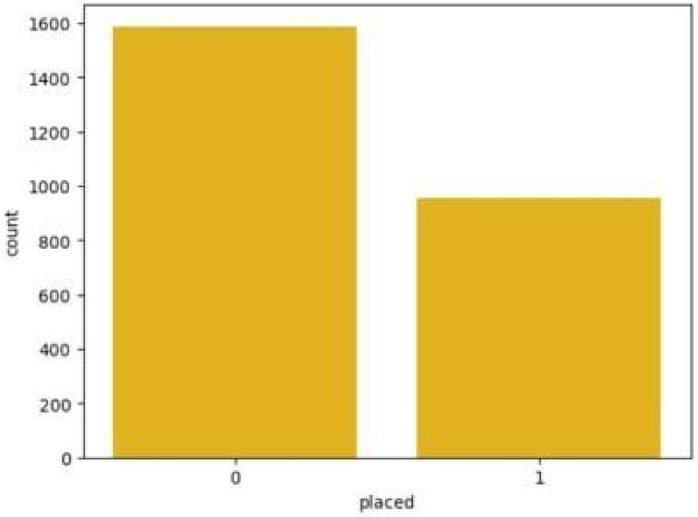
These have the lowest entry percentage, but half of them have found a job



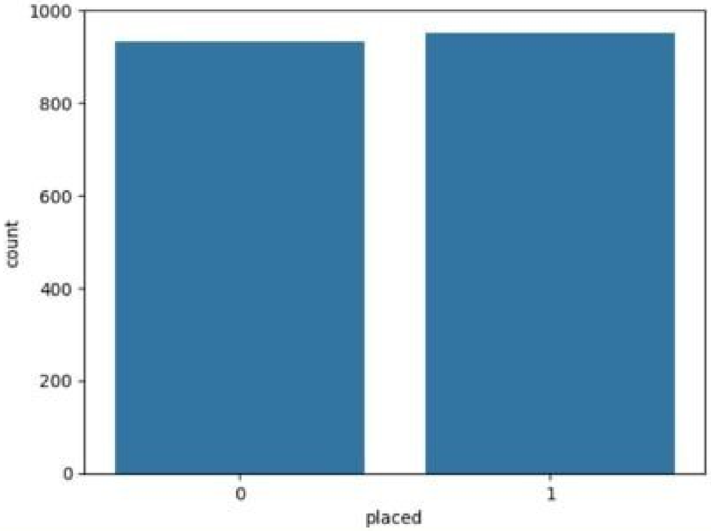
work_authorization_status	CPT	Canada Citizen	Citizen	Green Card	H1B	Not Authorized	OPT	Other	STEM OPT
Not placed %	52.08	55.00	47.87	48.89	63.89	100.00	51.46	45.00	60.00
Placed %	47.92	45.00	52.13	51.11	36.11	nan	48.54	55.00	40.00
population %	5.09	1.06	54.72	7.16	1.91	0.21	25.34	4.24	0.27

None of these entries can find a job

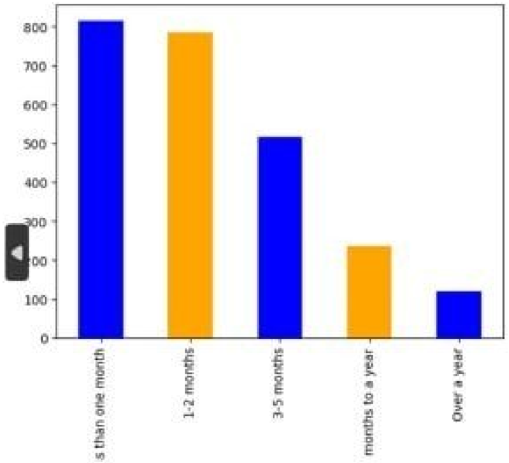
Unbalanced data



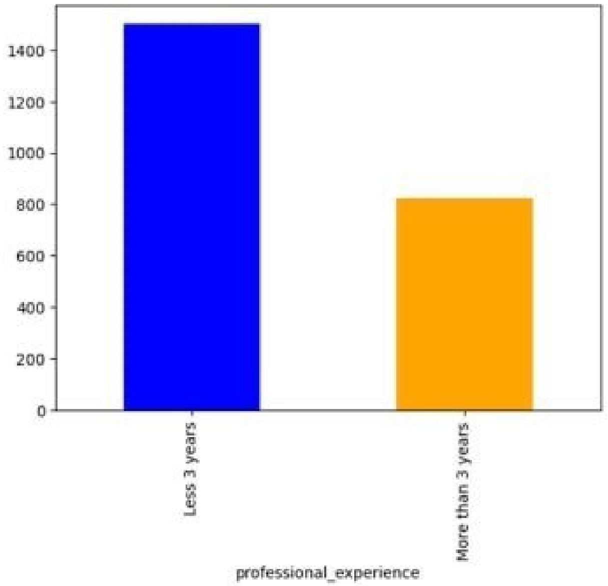
Balanced data after removing the values whose fate is not known in the Pathrise Status column



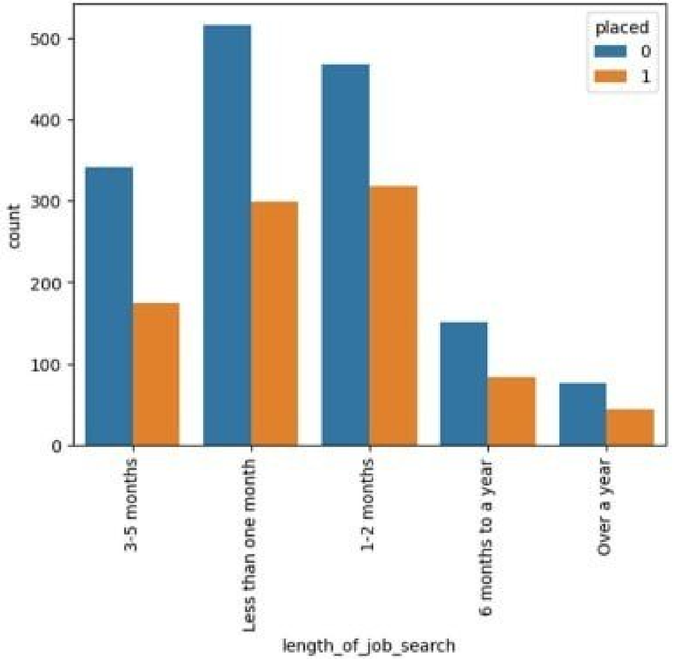
Before categorizing the Professional experience column



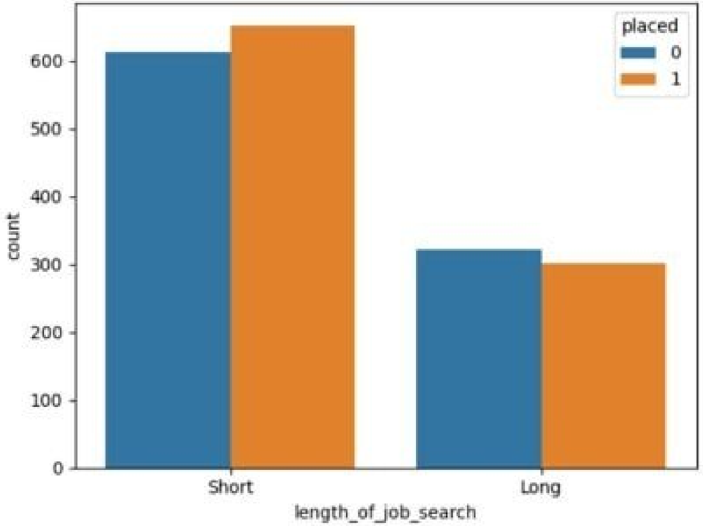
After categorizing the Professional experience column

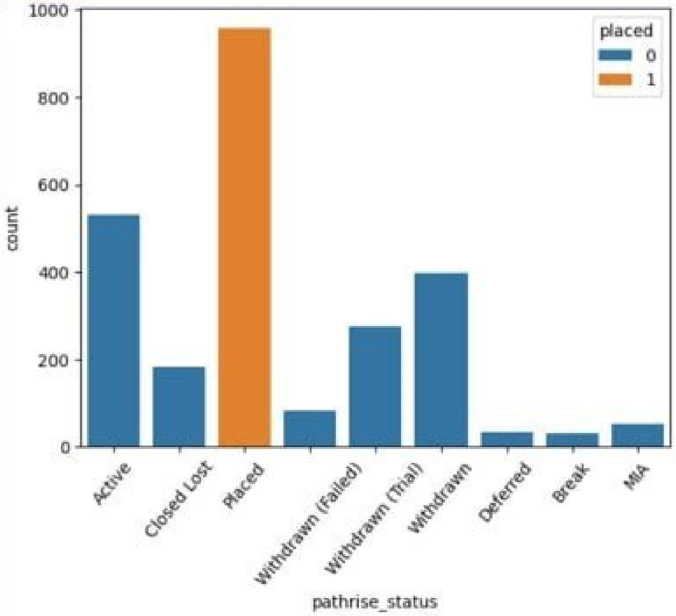


Before categorizing the Length of Job Search column



Aftercategorizing the Length of Job Search column

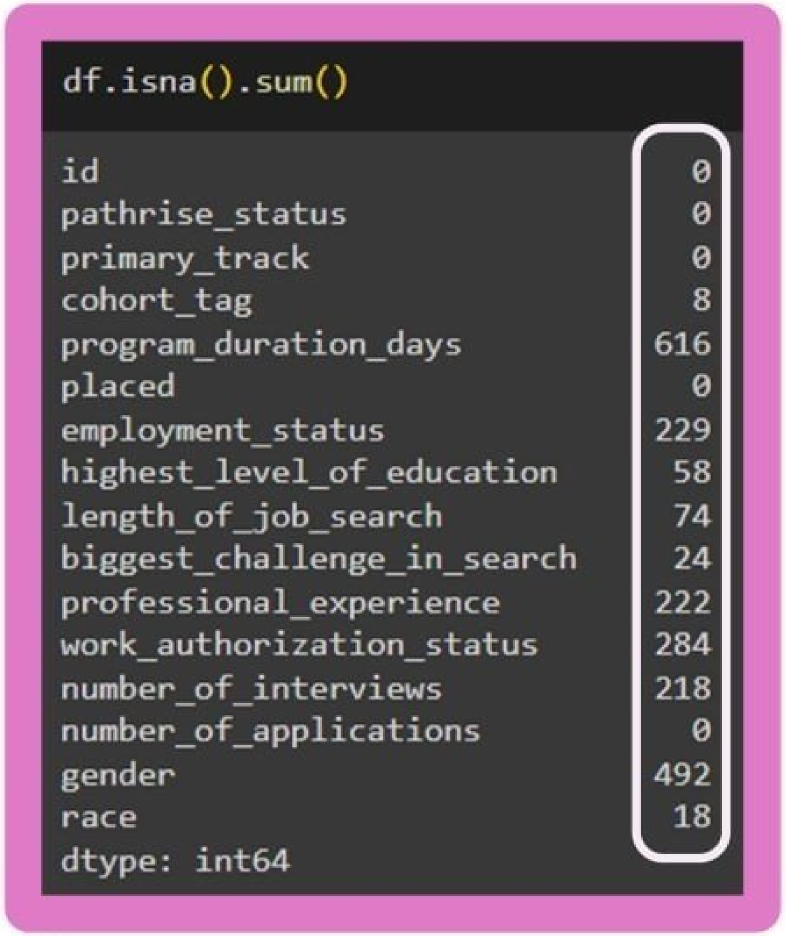




.Placed" means placing an order or transaction"
.Active" means currently ongoing or in progress"
.Withdrawn" means canceling or withdrawing a transaction or project"
.Withdrawn (Trial)" means canceling a trial or trial phase"
.Closed Lost" means ending a deal or project with a loss"
.Withdrawn (Failed)" means canceling a deal or project due to failure"
.MIA" can mean "missing" or unplanned ("unaccounted for")"
.Deferred" means postponing or rewriting a deal or project"
Break" depending on the context these data relate to, it may have"
various meanings, including temporary closure or interruption

=> **Therefore, we can remove "Active," "MIA," "Deferred," and "Break" as their .statuses are uncertain in the future, and they may find a job**

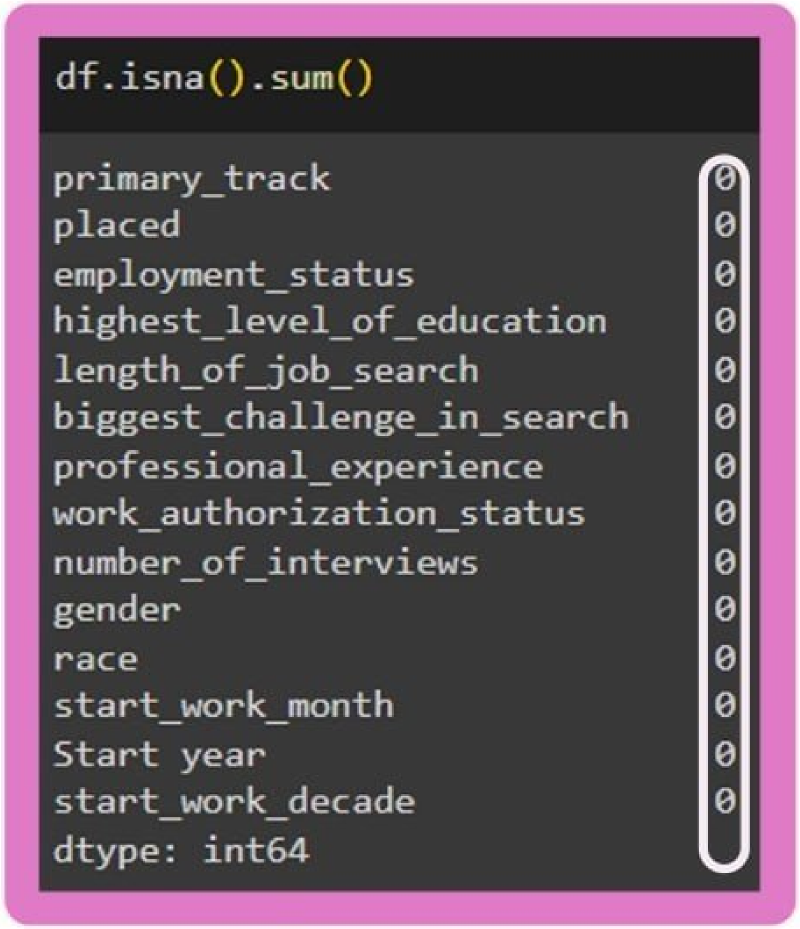
Before



```
df.isna().sum()
```

```
id                              0
pathrise_status                 0
primary_track                   0
cohort_tag                      8
program_duration_days         616
placed                          0
employment_status             229
highest_level_of_education     58
length_of_job_search           74
biggest_challenge_in_search    24
professional_experience       222
work_authorization_status     284
number_of_interviews          218
number_of_applications          0
gender                        492
race                           18
dtype: int64
```

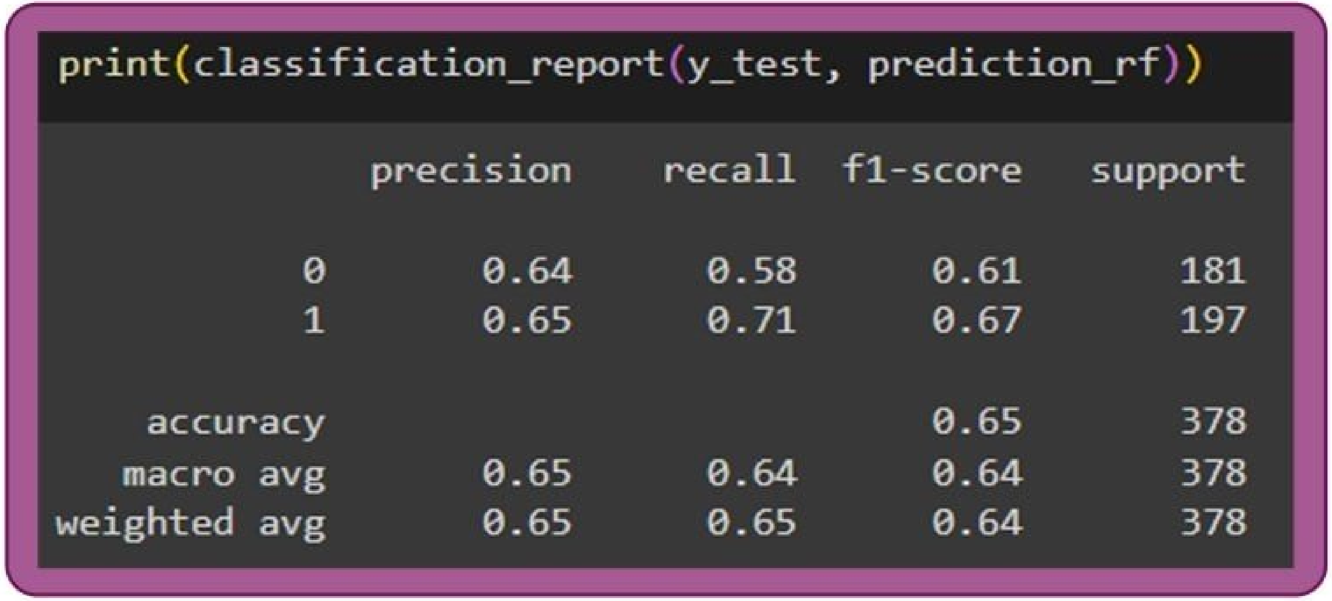
After



```
df.isna().sum()
```

```
primary_track                  0
placed                         0
employment_status              0
highest_level_of_education     0
length_of_job_search           0
biggest_challenge_in_search    0
professional_experience        0
work_authorization_status      0
number_of_interviews           0
gender                         0
race                           0
start_work_month               0
Start year                     0
start_work_decade              0
dtype: int64
```

We removed the missing values, which were few in number, and filled in the missing values, which were large, using the averaging method and frequent mode, which gave more accuracy to the model



```
print(classification_report(y_test, prediction_rf))
```

```
              precision    recall  f1-score   support

           0       0.64      0.58      0.61       181
           1       0.65      0.71      0.67       197

    accuracy                           0.65       378
   macro avg       0.65      0.64      0.64       378
weighted avg       0.65      0.65      0.64       378
```

Confusion Matrix Error I & Error II

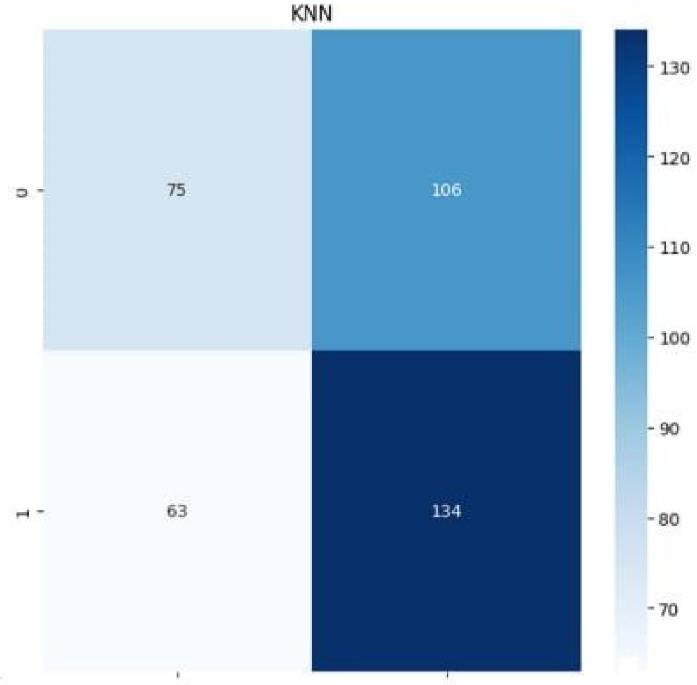
Accuracy = (TP+TN)/(TP+FP+FN+TN) صحت(نسبت پیش بینی های درست به کل نمونه)

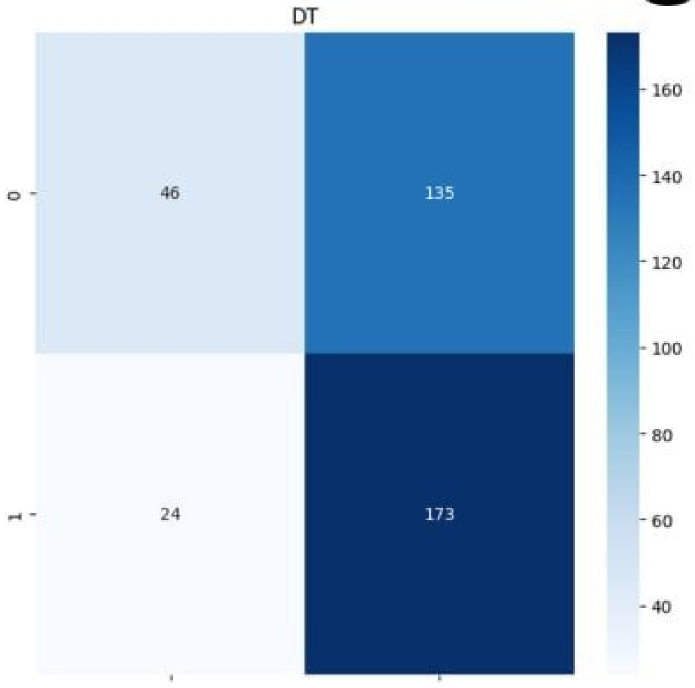
Precision = TP/(TP+FP) دقت(نسبت پیش بینی های مثبت درست به کل پیش بینی های مثبت)

Recall = TP/(TP+FN) بازخوانی(نسبت پیش بینی های مثبت درست به تعداد واقعی مثبت)

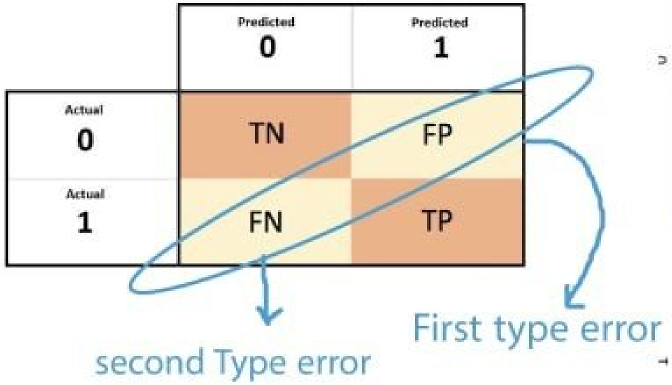
F1 Score = 2*(Recall*Precision)/(Recall + Precision)

Confusion matrix diagrams





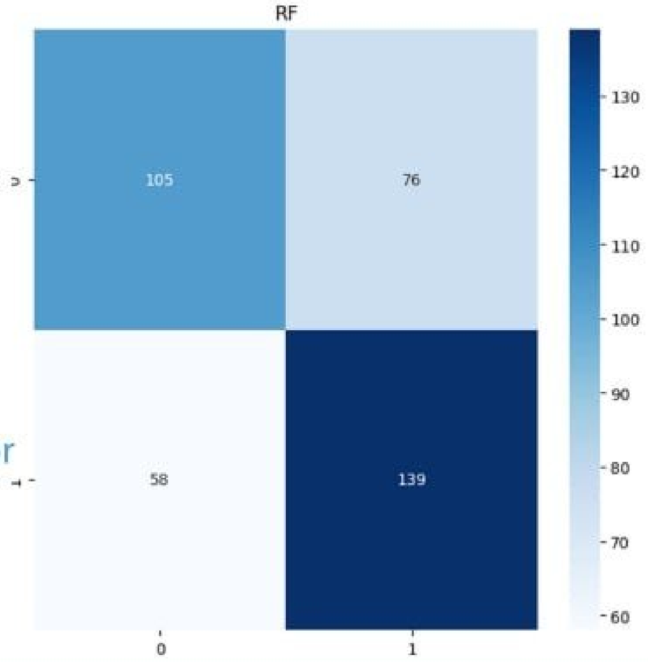
We prefer our First Type error to be lower in this situation because Pathrise invests in people and Pathrise loses if people don't get jobs

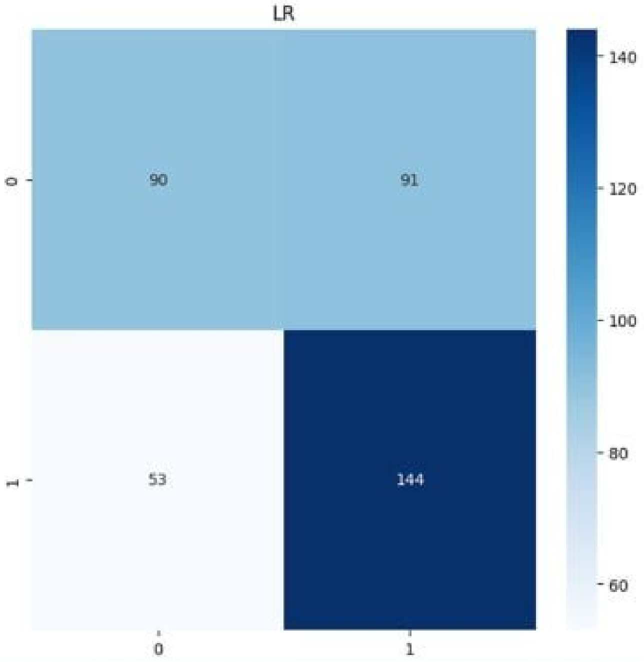


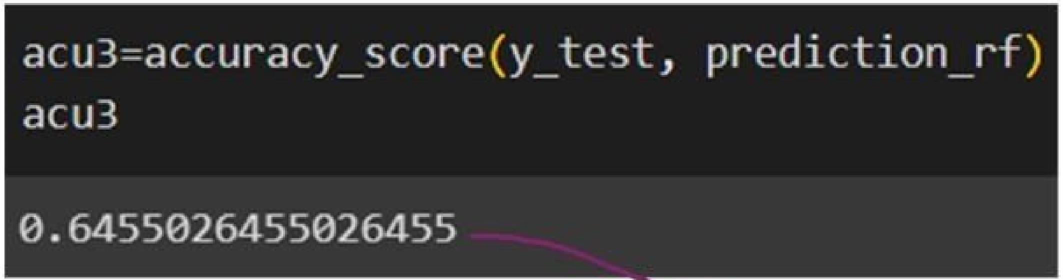
	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

First type error

second Type error



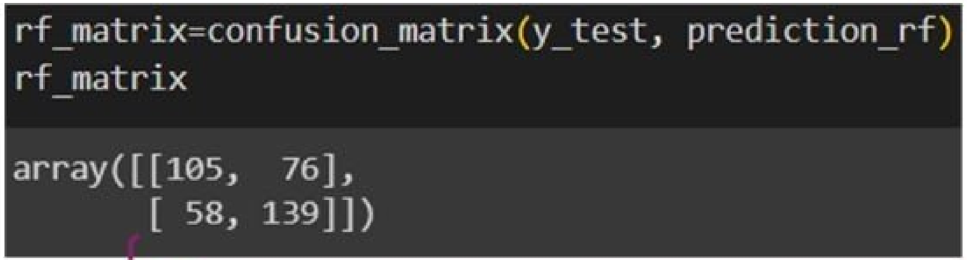




```
acu3=accuracy_score(y_test, prediction_rf)
acu3
```

```
0.6455026455026455
```

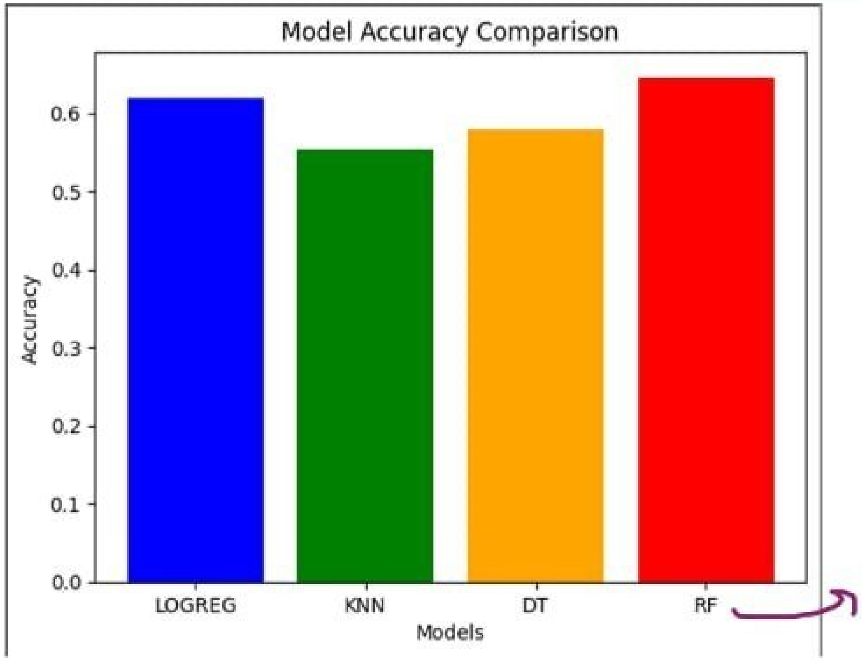
Accuracy of random forest model



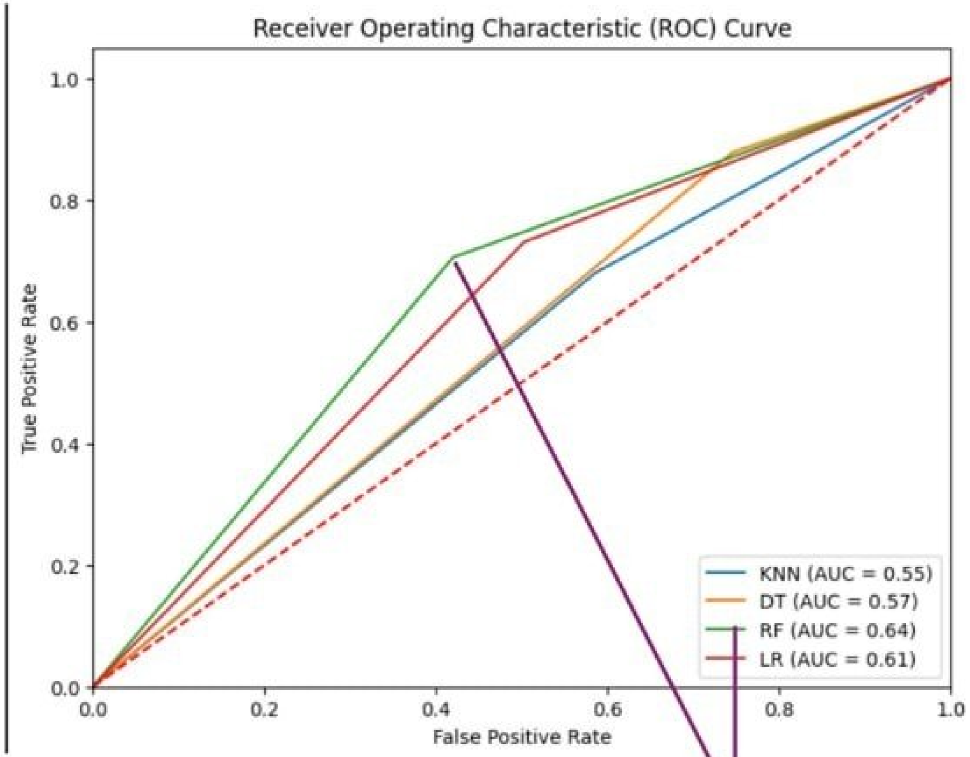
```
rf_matrix=confusion_matrix(y_test, prediction_rf)
rf_matrix
```

```
array([[105,  76],
       [ 58, 139]])
```

The confusion matrix of the random forest that has the lowest First type error



The highest accuracy is related to random forests



The line that covers the largest area corresponds to random forests