employee_attrition_prediction

August 17, 2025

```
[5]: import pandas as pd
     df = pd.read_csv("HR attrition.csv")
     df.head()
```

```
[5]:    Age Attrition     BusinessTravel  DailyRate              Department  \
     0   41       Yes      Travel_Rarely       1102                   Sales
     1   49        No  Travel_Frequently        279  Research & Development
     2   37       Yes      Travel_Rarely       1373  Research & Development
     3   33        No  Travel_Frequently       1392  Research & Development
     4   27        No      Travel_Rarely        591  Research & Development

        DistanceFromHome  Education EducationField  EmployeeCount  EmployeeNumber  \
     0                 1          2  Life Sciences              1               1
     1                 8          1  Life Sciences              1               2
     2                 2          2          Other              1               4
     3                 3          4  Life Sciences              1               5
     4                 2          1        Medical              1               7

        …  RelationshipSatisfaction StandardHours  StockOptionLevel  \
     0  …                         1            80                 0
     1  …                         4            80                 1
     2  …                         2            80                 0
     3  …                         3            80                 0
     4  …                         4            80                 1

        TotalWorkingYears  TrainingTimesLastYear WorkLifeBalance  YearsAtCompany  \
     0                  8                      0               1               6
     1                 10                      3               3              10
     2                  7                      3               3               0
     3                  8                      3               3               8
     4                  6                      3               3               2

       YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
     0                  4                        0                     5
     1                  7                        1                     7
     2                  0                        0                     0
     3                  7                        3                     0
     4                  2                        2                     2
```

```
[5 rows x 35 columns]
```

[7]:

```
print("shape of dataset:", df.shape)
```

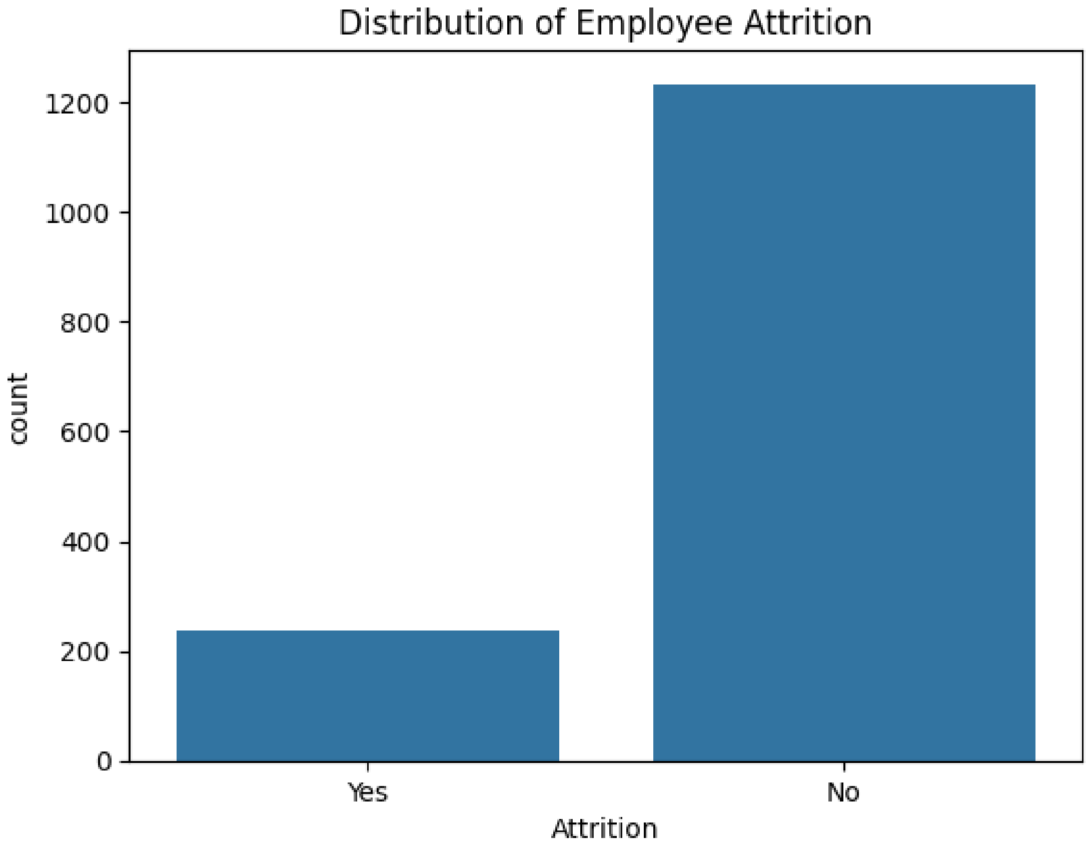
```
shape of dataset: (1470, 35)
```

[8]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix,␣
 ↪roc_auc_score, accuracy_score
```

[9]:

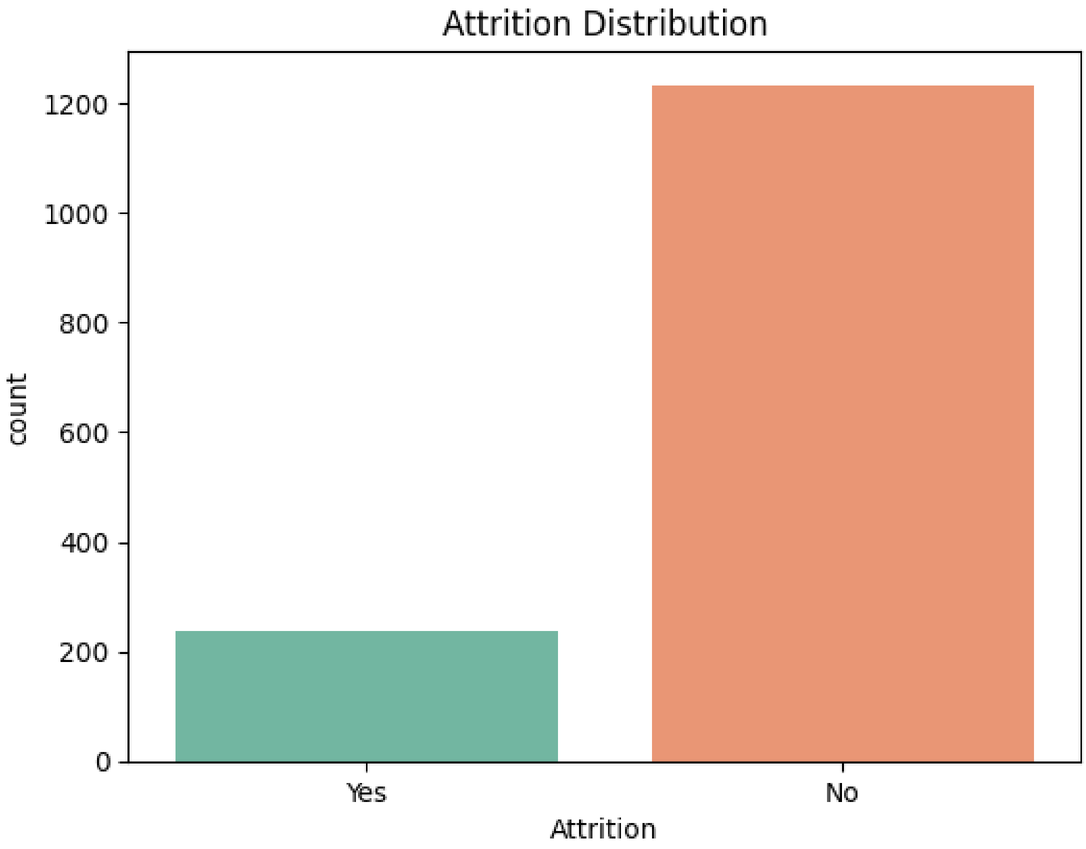
```
import pandas as pd
df = pd.read_csv("HR attrition.csv")
df['Attrition'].value_counts()
import seaborn as sns
import matplotlib.pyplot as plt
sns.countplot(data = df, x= 'Attrition')
plt.title('Distribution of Employee Attrition')
plt.show()
df.describe
df.isnull().sum()
categorical_cols = df.select_dtypes(include=['object']).columns
numerical_cols = df.select_dtypes(include=['int64','float64']).columns
print("categorical columns", categorical_cols)
print("numerical columns", numerical_cols)
sns.countplot(data = df, x='Attrition', palette ='Set2')
plt.title("Attrition Distribution")
plt.show()
attrition_rate = df['Attrition'].value_counts(normalize= True)*100
print(attrition_rate)
```



```
categorical columns Index(['Attrition', 'BusinessTravel', 'Department',
'EducationField', 'Gender',
       'JobRole', 'MaritalStatus', 'Over18', 'OverTime'],
      dtype='object')
numerical columns Index(['Age', 'DailyRate', 'DistanceFromHome', 'Education',
'EmployeeCount',
       'EmployeeNumber', 'EnvironmentSatisfaction', 'HourlyRate',
       'JobInvolvement', 'JobLevel', 'JobSatisfaction', 'MonthlyIncome',
       'MonthlyRate', 'NumCompaniesWorked', 'PercentSalaryHike',
       'PerformanceRating', 'RelationshipSatisfaction', 'StandardHours',
       'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',
       'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole',
       'YearsSinceLastPromotion', 'YearsWithCurrManager'],
      dtype='object')
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_2160\3593943961.py:15: 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.
```

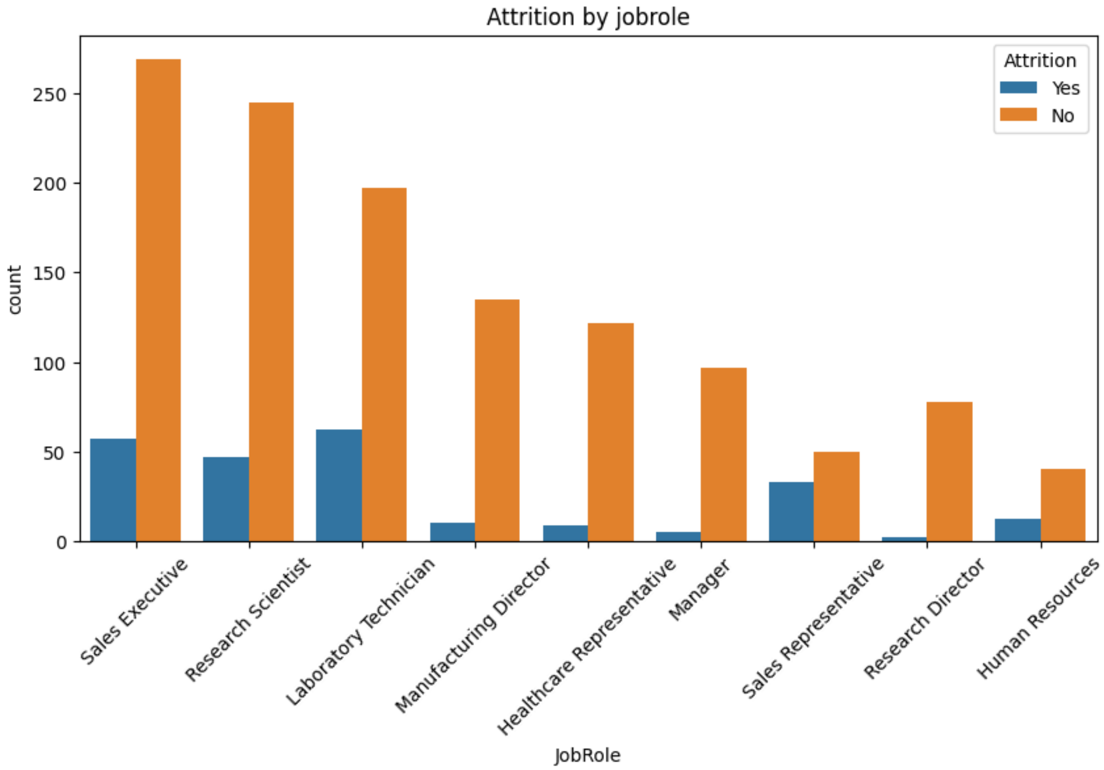
```
  sns.countplot(data = df, x='Attrition', palette ='Set2')
```



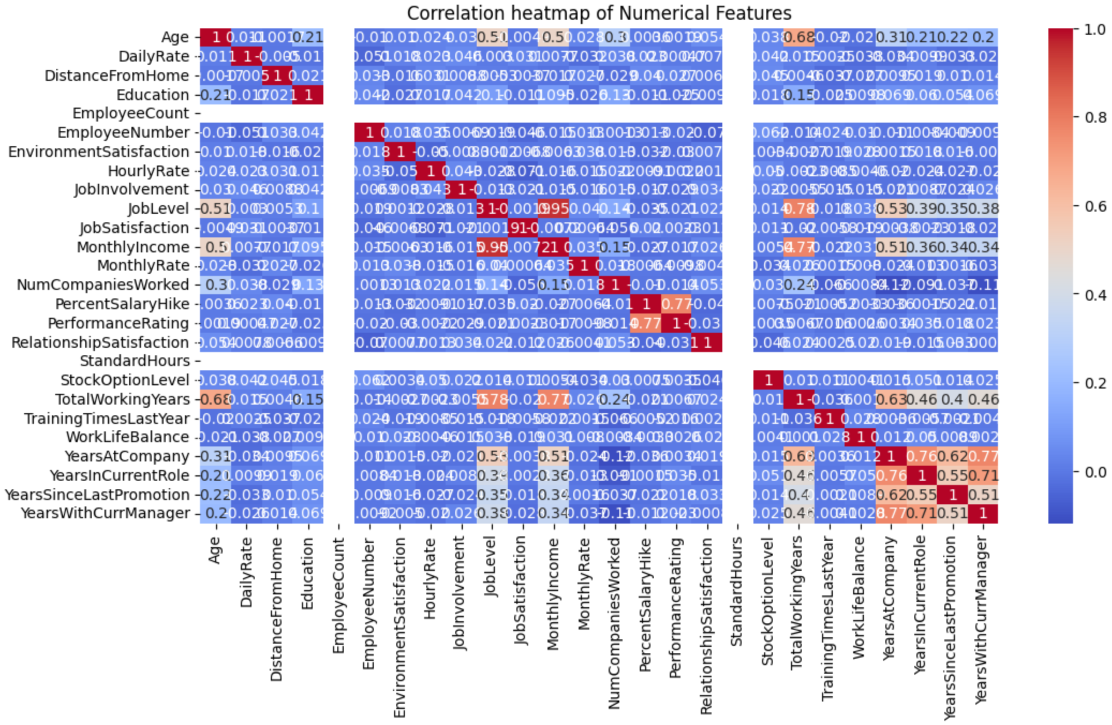
```
Attrition
No     83.877551
Yes    16.122449
Name: proportion, dtype: float64
```

[10]:

```
plt.figure(figsize=(10,5))
sns.countplot(data=df, x='JobRole', hue='Attrition')
plt.xticks(rotation=45)
plt.title("Attrition by jobrole")
plt.show()
```



```
[11]: plt.figure(figsize=(12,6))
      sns.heatmap(df[numerical_cols].corr(), annot=True, cmap='coolwarm')
      plt.title("Correlation heatmap of Numerical Features")
      plt.show()
```



```
[12]: from sklearn.preprocessing import LabelEncoder
      df_encoded = df.copy()
      label_enc = LabelEncoder()
      for col in categorical_cols:
          df_encoded[col] = label_enc.fit_transform(df_encoded[col])
```

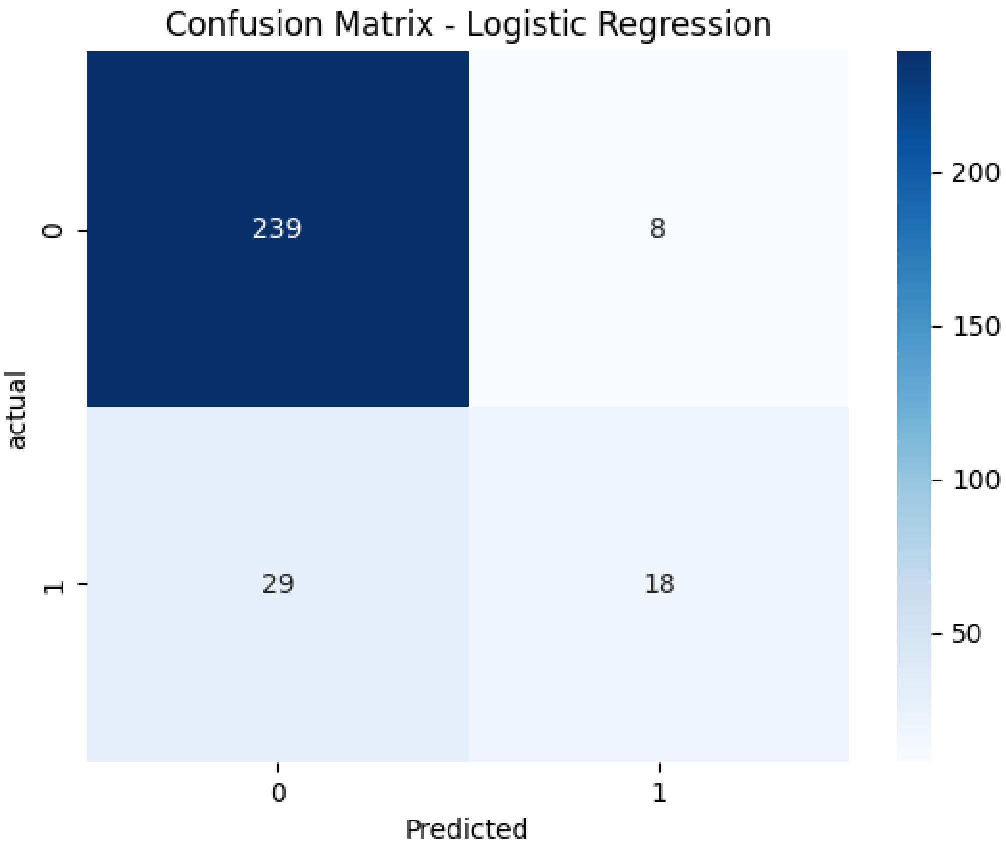
```
[13]: from sklearn.model_selection import train_test_split
      x = df_encoded.drop('Attrition', axis = 1)
      y = df_encoded['Attrition']
      x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.2,␣
       ↪random_state = 42, stratify=y)
```

```
[14]: from sklearn.preprocessing import StandardScaler
      scaler = StandardScaler()
      x_train = scaler.fit_transform(x_train)
      x_test = scaler.transform(x_test)
```

```
[15]: from sklearn.linear_model import LogisticRegression
      from sklearn.metrics import classification_report, accuracy_score,␣
       ↪confusion_matrix
      model = LogisticRegression(max_iter = 1000)
      model.fit(x_train,y_train)
      y_pred = model.predict(x_test)
```

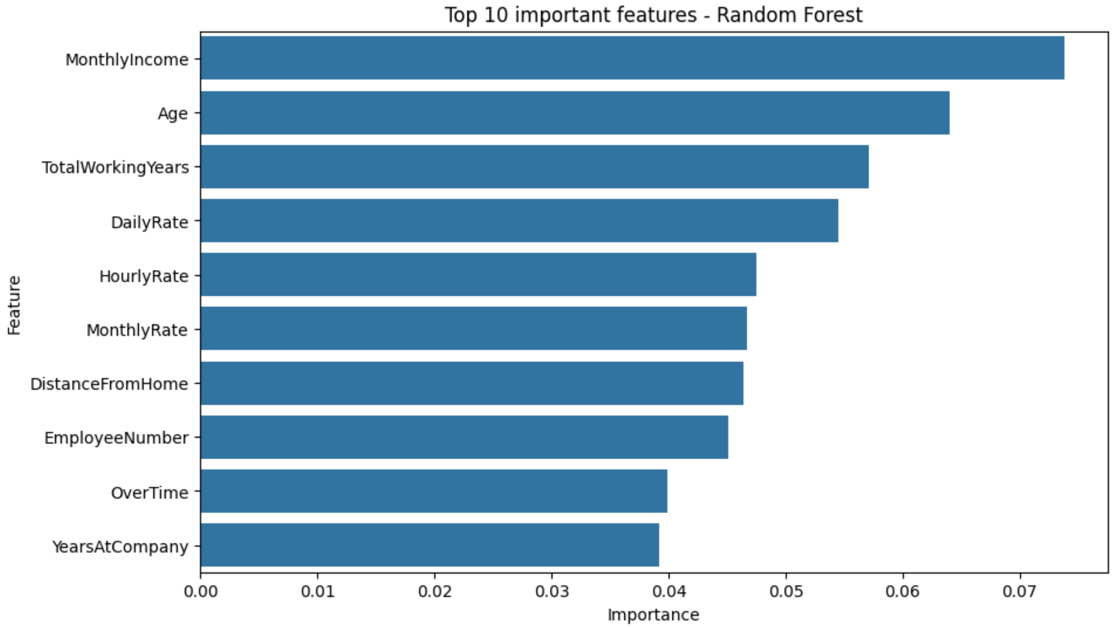
```
print("Logistic regression accuracy", accuracy_score(y_test, y_pred))
print("Classification Report", classification_report(y_test, y_pred))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues')
plt.title("Confusion Matrix - Logistic Regression")
plt.xlabel("Predicted")
plt.ylabel("actual")
plt.show()
```

```
Logistic regression accuracy 0.8741496598639455
Classification Report               precision    recall  f1-score   support

           0       0.89      0.97      0.93       247
           1       0.69      0.38      0.49        47

    accuracy                           0.87       294
   macro avg       0.79      0.68      0.71       294
weighted avg       0.86      0.87      0.86       294
```



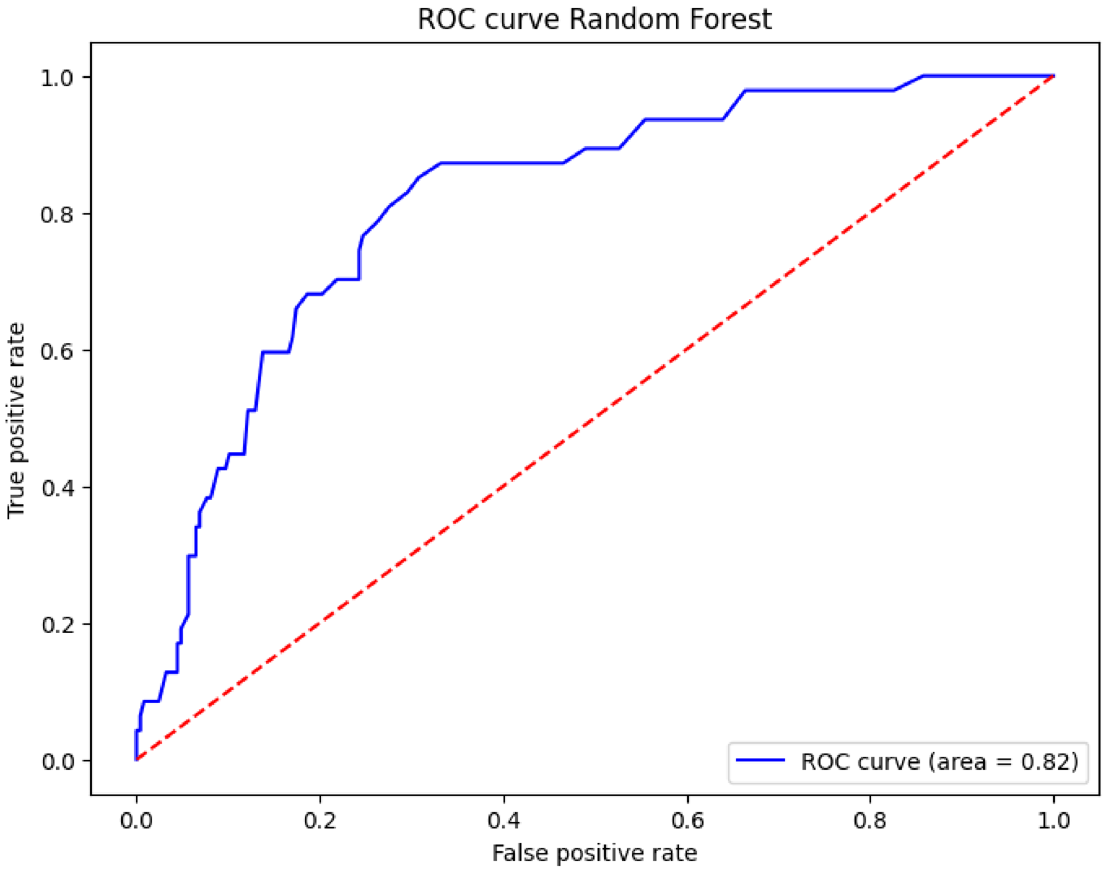
```
[17]: from sklearn.ensemble import RandomForestClassifier
      rf_model = RandomForestClassifier(n_estimators =200, random_state = 42)
      rf_model.fit(x_train,y_train)
      y_pred_rf = rf_model.predict(x_test)
      print("Random Forest Accuracy", accuracy_score(y_test,y_pred_rf))
      print("Classification Report", classification_report(y_test,y_pred_rf))
      importances = rf_model.feature_importances_
      feature_names = x.columns
      feat_imp_df = pd.DataFrame({"Feature": feature_names, "Importance":␣
       ↪importances})
      feat_imp_df = feat_imp_df.sort_values(by = "Importance", ascending=False)
      plt.figure(figsize=(10,6))
      sns.barplot(x="Importance", y="Feature", data = feat_imp_df.head(10))
      plt.title("Top 10 important features - Random Forest")
      plt.show()
```

```
Random Forest Accuracy 0.8333333333333334
Classification Report               precision    recall  f1-score   support

           0       0.85      0.97      0.91       247
           1       0.43      0.13      0.20        47

    accuracy                           0.83       294
   macro avg       0.64      0.55      0.55       294
weighted avg       0.79      0.83      0.79       294
```



[18]:

```
from sklearn.metrics import roc_curve, auc
y_proba_rf = rf_model.predict_proba(x_test)[:,1]
fpr,tpr, thresholds = roc_curve(y_test, y_proba_rf)
roc_auc = auc(fpr,tpr)
plt.figure(figsize=(8,6))
plt.plot(fpr,tpr, color='blue', label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0,1], [0,1], color= 'red', linestyle='--')
plt.xlabel("False positive rate")
plt.ylabel("True positive rate")
plt.title("ROC curve Random Forest")
plt.legend(loc='lower right')
plt.show()
```



[1]:

```
pip install joblib
```

```
Requirement already satisfied: joblib in
c:\users\hp\appdata\local\programs\python\python312\lib\site-packages (1.5.1)
Note: you may need to restart the kernel to use updated packages.
```

```
[notice] A new release of pip is available: 25.1.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

[ ]:

```

```