

```
!pip install pandas
!pip install numpy
!pip install matplotlib
!pip install seaborn
!pip install scikit-learn
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.1.4)
Requirement already satisfied: numpy<2,>=1.22.4 in /usr/local/lib/python3.10/dist-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (1.26.4)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.2.1)
Requirement already satisfied: cyclor>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.53.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.5)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.26.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-packages (0.13.1)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in /usr/local/lib/python3.10/dist-packages (from seaborn) (1.26.4)
Requirement already satisfied: pandas>=1.2 in /usr/local/lib/python3.10/dist-packages (from seaborn) (2.1.4)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in /usr/local/lib/python3.10/dist-packages (from seaborn) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.2.1)
Requirement already satisfied: cyclor>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (4.53.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (24.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.2->seaborn) (2024.1)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.2->seaborn) (2024.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.16.0)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.3.2)
Requirement already satisfied: numpy<2.0,>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.26.4)
Requirement already satisfied: scipy>=1.5.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.13.1)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.5.0)
```

```
import pandas as pd
# Load the data
df = pd.read_csv("/content/Cleaned_HR_Employee_Attrition.csv")
```

```
print(df.describe())
```

```
count    1470.000000    1470.000000    1470.000000    1470.000000    1470.000000    \
mean      36.923810    802.485714      9.192517      2.912925      1.000000
std        9.135373    403.509100      8.106864      1.024165      0.000000
min       18.000000    102.000000      1.000000      1.000000      1.000000
25%       30.000000    465.000000      2.000000      2.000000      1.000000
50%       36.000000    802.000000      7.000000      3.000000      1.000000
75%       43.000000   1157.000000     14.000000      4.000000      1.000000
max       60.000000   1499.000000     29.000000      5.000000      1.000000

EmployeeNumber  EnvironmentSatisfaction  HourlyRate  JobInvolvement  \
count    1470.000000      1470.000000    1470.000000    1470.000000
mean     1024.865306      2.721769      65.891156      2.729932
std       602.024335      1.093082      20.329428      0.711561
min        1.000000      1.000000      30.000000      1.000000
25%       491.250000      2.000000      48.000000      2.000000
50%       1020.500000      3.000000      66.000000      3.000000
75%      1555.750000      4.000000      83.750000      3.000000
max      2068.000000      4.000000     100.000000      4.000000

JobLevel  ...  StandardHours  StockOptionLevel  TotalWorkingYears  \
count    1470.000000  ...      1470.000000      1470.000000      1470.000000
mean       2.063946  ...         80.000000         0.793878        11.279592
std        1.106940  ...          0.000000         0.852077         7.780782
min         1.000000  ...          0.000000         0.000000          0.000000
25%         1.000000  ...          0.000000         0.000000          6.000000
50%         2.000000  ...          0.000000         1.000000         10.000000
```

75%	3.000000	...	80.0	1.000000	15.000000
max	5.000000	...	80.0	3.000000	40.000000

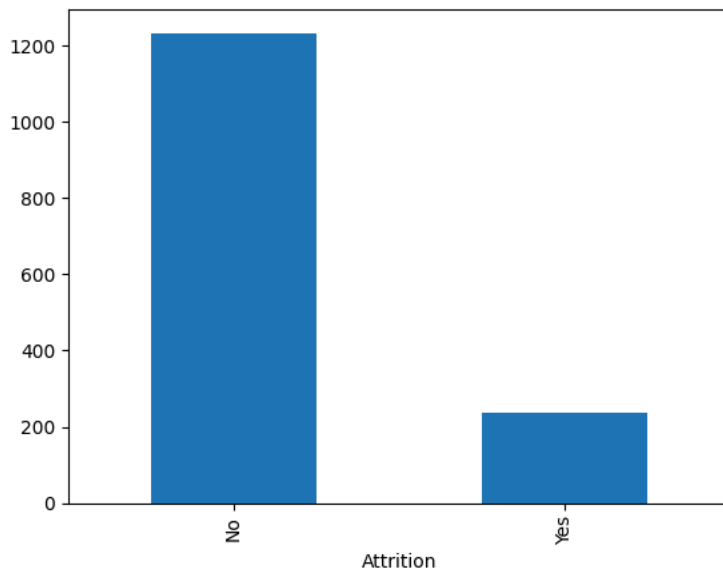
	TrainingTimesLastYear	WorkLifeBalance	YearsAtCompany	\
count	1470.000000	1470.000000	1470.000000	
mean	2.799320	2.761224	7.008163	
std	1.289271	0.706476	6.126525	
min	0.000000	1.000000	0.000000	
25%	2.000000	2.000000	3.000000	
50%	3.000000	3.000000	5.000000	
75%	3.000000	3.000000	9.000000	
max	6.000000	4.000000	40.000000	

	YearsInCurrentRole	YearsSinceLastPromotion	YearsWithCurrManager	\
count	1470.000000	1470.000000	1470.000000	
mean	4.229252	2.187755	4.123129	
std	3.623137	3.222430	3.568136	
min	0.000000	0.000000	0.000000	
25%	2.000000	0.000000	2.000000	
50%	3.000000	1.000000	3.000000	
75%	7.000000	3.000000	7.000000	
max	18.000000	15.000000	17.000000	

	Tenure
count	1470.000000
mean	7.008163
std	6.126525
min	0.000000
25%	3.000000
50%	5.000000
75%	9.000000

```
df['Attrition'].value_counts().plot(kind='bar')
```

```
<Axes: xlabel='Attrition'>
```



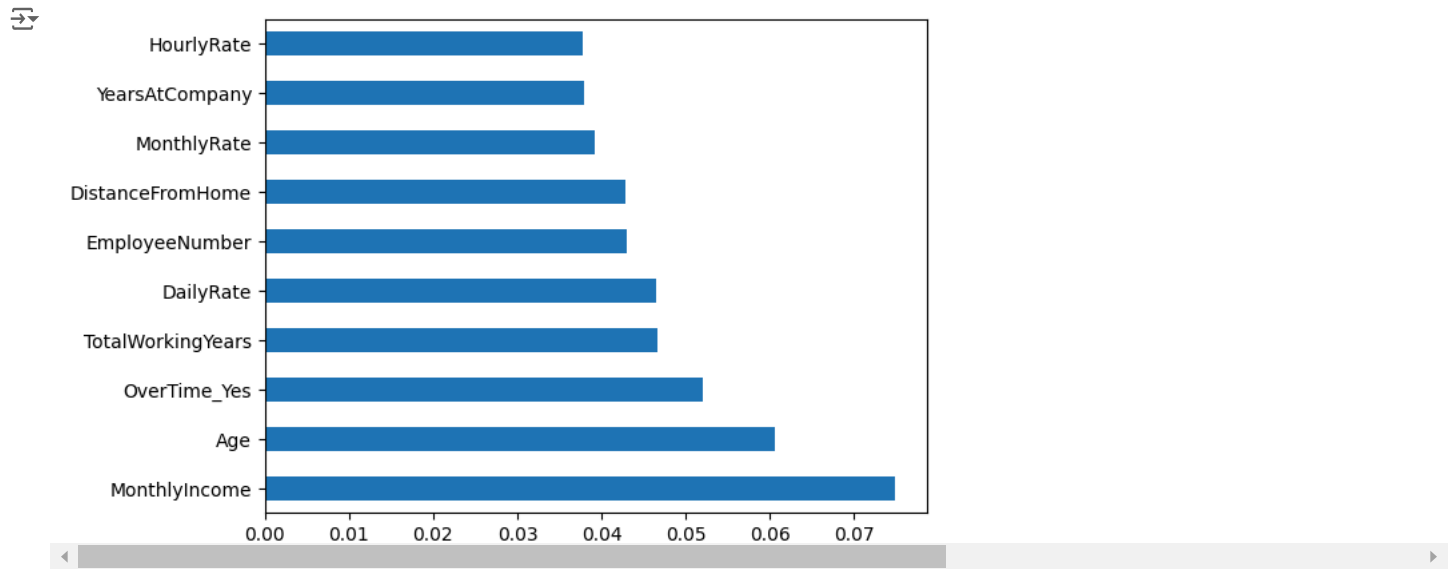
```
df = pd.get_dummies(df, drop_first=True)
#The Attrition column was renamed to Attrition_Yes by get_dummies
X = df.drop('Attrition_Yes', axis=1)
y = df['Attrition_Yes']
```

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
model = RandomForestClassifier()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy * 100:.2f}%')
```

```
Accuracy: 86.17%
```

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
feature_importances = pd.Series(model.feature_importances_, index=X.columns)
feature_importances.nlargest(10).plot(kind='barh')
plt.show()
```



```
import joblib
joblib.dump(model, 'random_forest_model.pkl')
```

```
['random_forest_model.pkl']
```