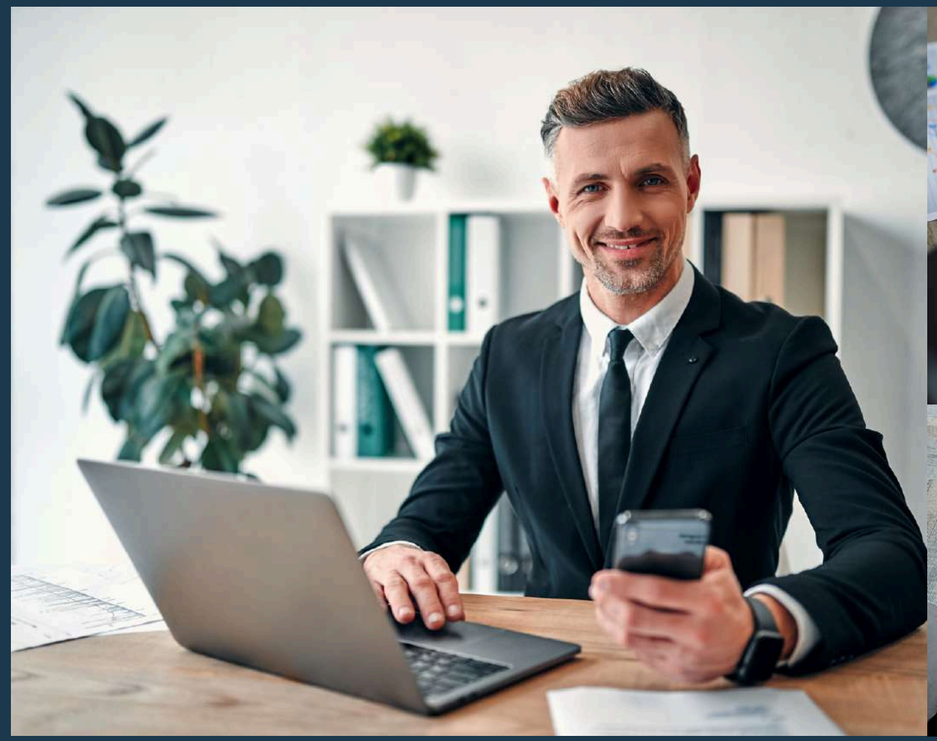
TEAM 5

RATING EMPLOYEE

Isky Dwi Aprilianto (064002200006)
Putri Syabillah (064002200015)
Vania Rahma Dewi (064002200030)
Adriansyah Maulana Putra (064002200046)
Naufal Fawwaz (064002300037)

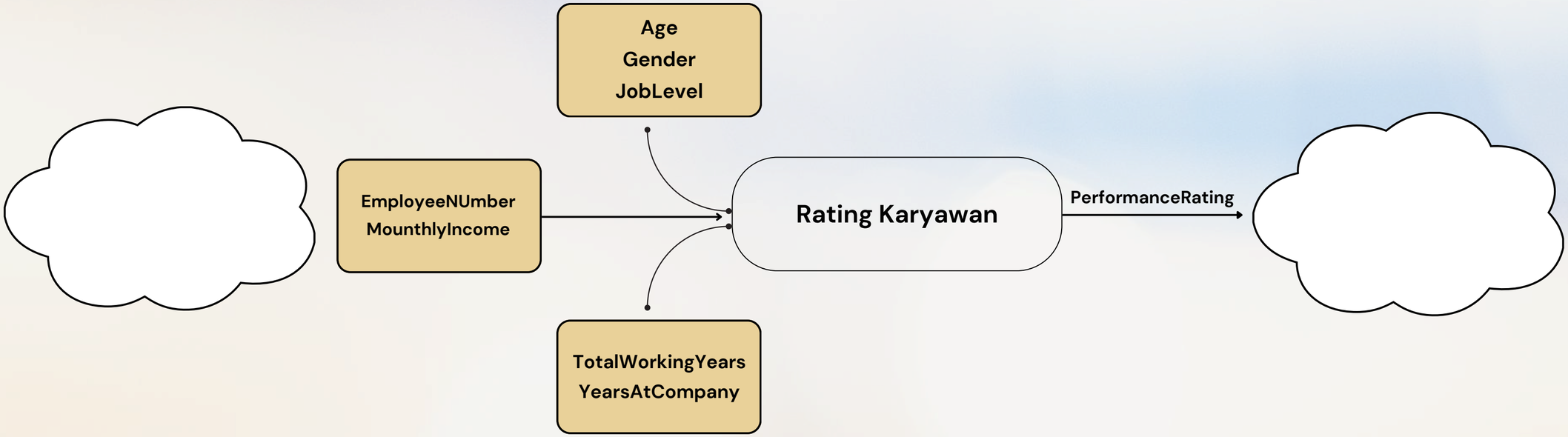
UNIVERSITAS TRISAKTI

RANCANGAN SISTEM CERDAS



UNIVERSITAS TRISAKTI

Diagram Sistem Thinking



UNIVERSITAS TRISAKTI

MEANINGFUL OBJECTIVE

Organization Objective

- Meningkatkan Produktivitas Karyawan
- Menilai kinerja karyawan dalam mencapai target bisnis

Leading Indicator

Menggunakan indikator tingkat kepuasan karyawan sebagai prediktor potensial untuk kinerja di masa depan.

Customer Outcomes

- meningkatkan produktivitas dan kinerja karyawan dalam sehari - hari
- mencapai target bisnis melalui evaluasi kinerja karyawan.
- Menciptakan pengalaman pengguna yang informatif

Model Properties

Memastikan bahwa model penilaian karyawan memiliki akurasi yang tinggi dalam mengidentifikasi kinerja yang sebenarnya.

UNIVERSITAS TRISAKTI

INTELLIGENCE EXPERIENCE

Presenting Intelligence

- Menampilkan penilaian kinerja menggunakan grafik dan metrik kinerja karyawan
- Menggunakan visualisasi data yang jelas & informatif agar customer memahami data kinerja

Creat Data to Grow System

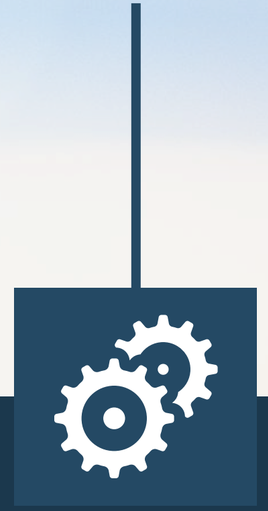
Memanfaatkan data yang dikumpulkan untuk melatih model-machine learning agar sistem dapat memberikan penilaian yang lebih akurat dan relevan seiring berjalannya waktu.

Minimize Intelligence Flaws

Memastikan bahwa sistem memiliki tingkat akurasi yang tinggi dalam mengevaluasi kinerja karyawan dan bahwa hasilnya sesuai dengan realitas kinerja mereka.

Achieve System Objective

Memberikan hasil rating yang sesuai dengan kinerja karyawan tersebut.



UNIVERSITAS TRISAKTI

INTELLIGENCE CREATION

Menggunakan model Machine Learning Random Forest, Decision Tree, dan MLP Classifier agar dapat menghasilkan prediksi rating karyawan yang sesuai serta merefining kinerja model Machine Learning tersebut dengan metode oversampling dan Learning Curve.

UNIVERSITAS TRISAKTI

INTELLIGENCE IMPLEMENTATION

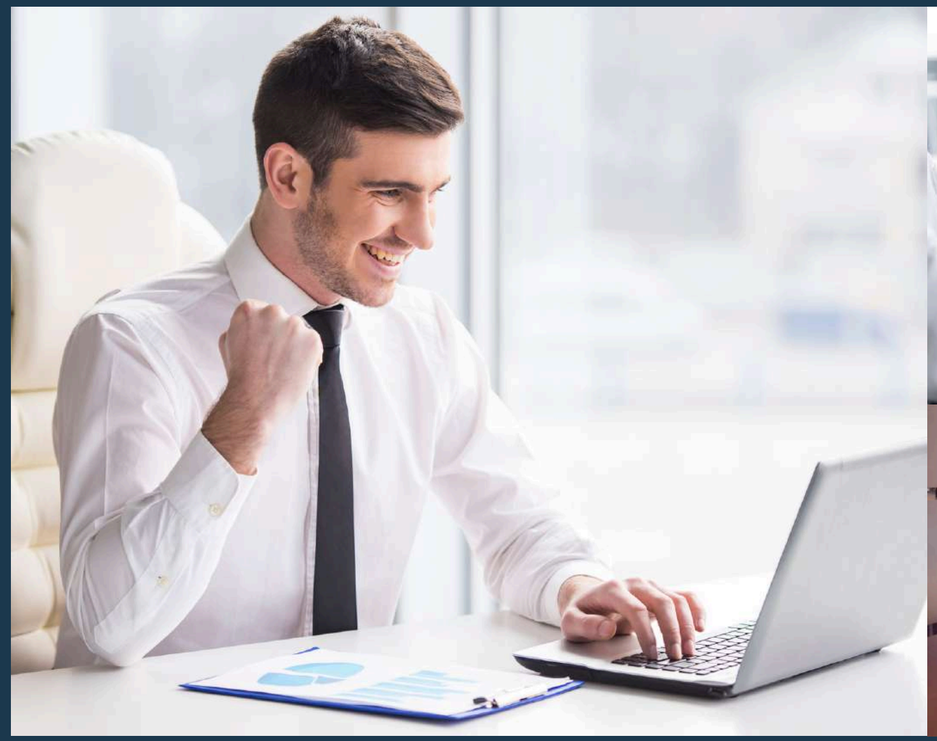
Melakukan analisis data dan menggunakan model machine learning untuk mengolah data kinerja karyawan agar mendapatkan hasil yang akurat.

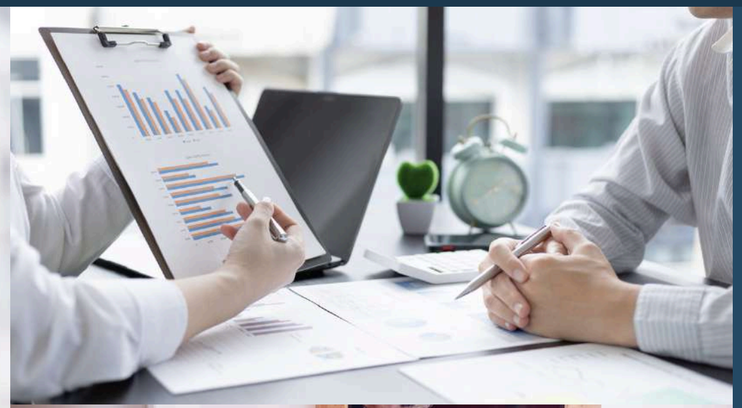
UNIVERSITAS TRISAKTI

INTELLIGENCE ORCHESTRATION

- Menggunakan model prediksi yang terbaik
- Melakukan pelatihan dan penyesuaian (test & training data) secara berkala

PEMROSESAN DATA





UNIVERSITAS TRISAKTI

IMPORT LIBRARY

UNIVERSITAS TRISAKTI

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# Models from Scikit-learn
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier

# Model Evaluations
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import confusion_matrix, classification_report

import warnings
warnings.filterwarnings('ignore')
```

UNIVERSITAS TRISAKTI

DATASET

HR_Analytics.csv (258.6 kB)

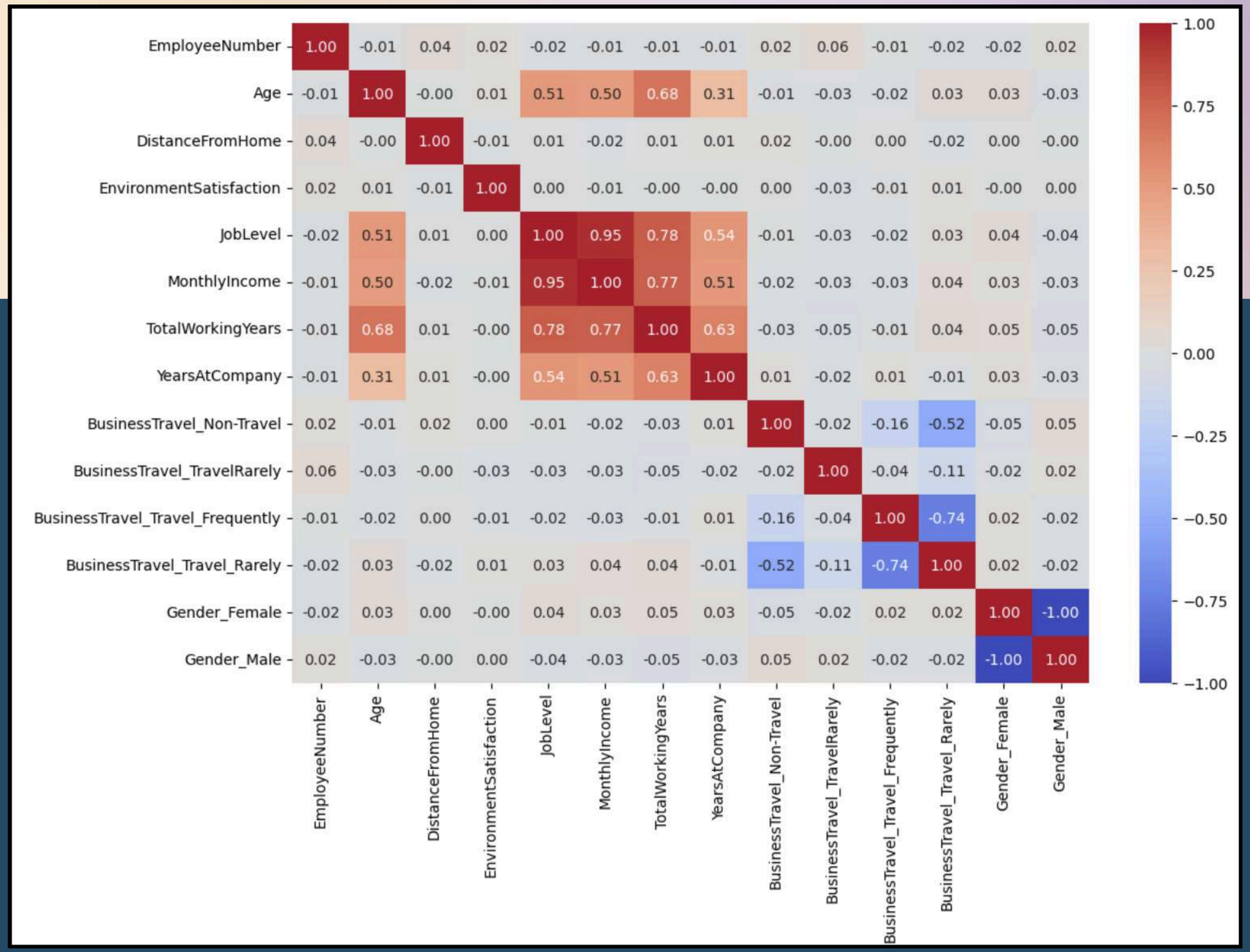
Detail | Compact | Column

38 of 38 columns

EmpID	Age	AgeGroup	Attrition	BusinessT...	DailyRate	Department	DistanceFr...	Education
RM297	18	18-25	Yes	Travel_Rarely	230	Research & Development	3	3
RM302	18	18-25	No	Travel_Rarely	812	Sales	10	3
RM458	18	18-25	Yes	Travel_Frequently	1306	Sales	5	3
RM728	18	18-25	No	Non-Travel	287	Research & Development	5	2
RM829	18	18-25	Yes	Non-Travel	247	Research & Development	8	1
RM973	18	18-25	No	Non-Travel	1124	Research & Development	1	3
RM1154	18	18-25	Yes	Travel_Frequently	544	Sales	3	2

UNIVERSITAS TRISAKTI

DIAGRAM



HEATMAP

MENGHAPUS FITUR

UNIVERSITAS TRISAKTI

```
# Menghapus beberapa kolom (ganti dengan nama-nama kolom yang ingin dihapus)
kolom_yang_dihapus = ['EmpID',
 'AgeGroup',
 'Attrition',
 'BusinessTravel',
 'DailyRate',
 'Department',
 'DistanceFromHome',
 'Education',
 'EducationField',
 'EmployeeCount',
 'EnvironmentSatisfaction',
 'Gender',
 'HourlyRate',
 'JobInvolvement',
 'JobRole',
 'JobSatisfaction',
 'MaritalStatus',
 'SalarySlab',
 'MonthlyRate',
 'NumCompaniesWorked',
 'Over18',
 'OverTime',
 'PercentSalaryHike',
 'RelationshipSatisfaction',
 'StandardHours',
 'StockOptionLevel',
 'TrainingTimesLastYear',
 'WorkLifeBalance',
 'YearsInCurrentRole',
 'YearsSinceLastPromotion',
 'YearsWithCurrManager']
df = df_raw.drop(columns=kolom_yang_dihapus)
```

UNIVERSITAS TRISAKTI

MEMISAH FITUR DAN TARGET

```
# Memilih kolom-kolom yang diinginkan untuk x
x = df_raw[['EmployeeNumber', 'Age', 'DistanceFromHome', 'BusinessTravel', 'EnvironmentSatisfaction', 'Gender', 'JobLevel', 'MonthlyIncome', 'TotalWorkingYears', 'YearsAtCompany']]

# Memilih kolom 'PerformanceRating' sebagai y
y = df_raw['PerformanceRating']
```

```
x.head()
```

[76]:

	EmployeeNumber	Age	JobLevel	MonthlyIncome	TotalWorkingYears	YearsAtCompany
0	405	18	1	1420	0	0
1	411	18	1	1200	0	0
2	614	18	1	1878	0	0
3	1012	18	1	1051	0	0
4	1156	18	1	1904	0	0

UNIVERSITAS TRISAKTI

DIAGRAM COUNTPLOT

```
]: Text(0.5, 0, 'Age(years)')
```

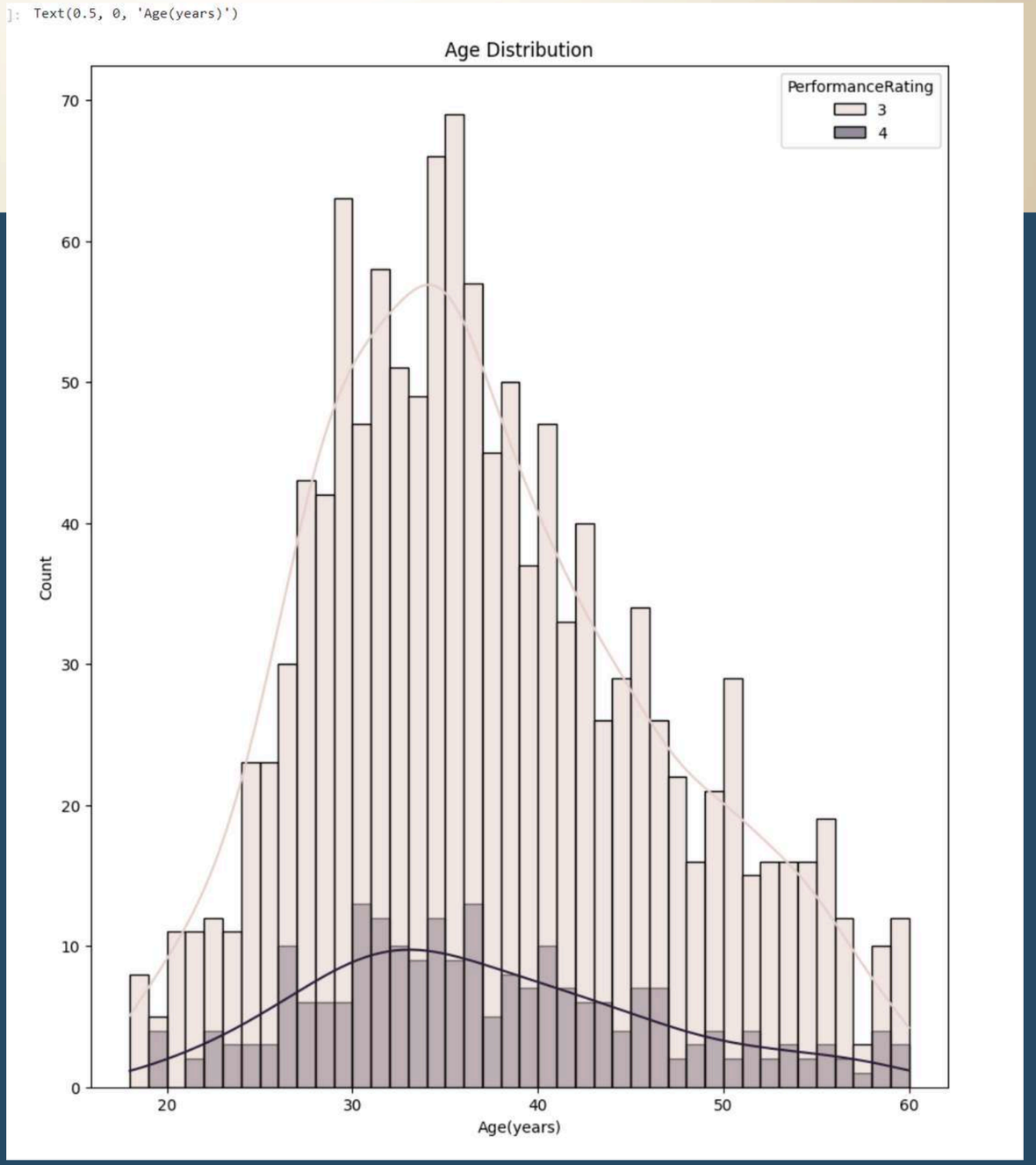
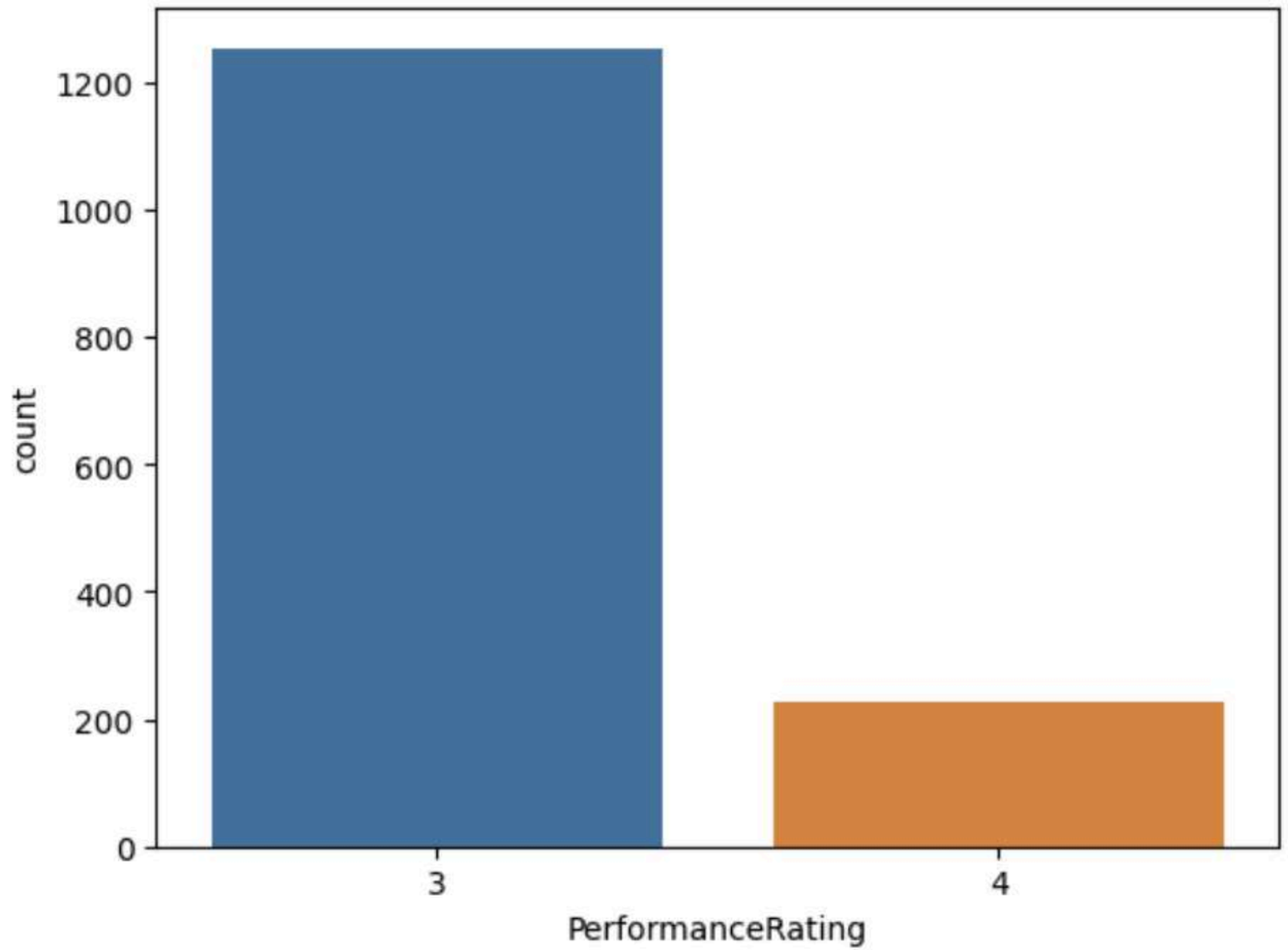


DIAGRAM COUNTPLOT

UNIVERSITAS TRISAKTI

```
sns.countplot(x=df["PerformanceRating"])
```

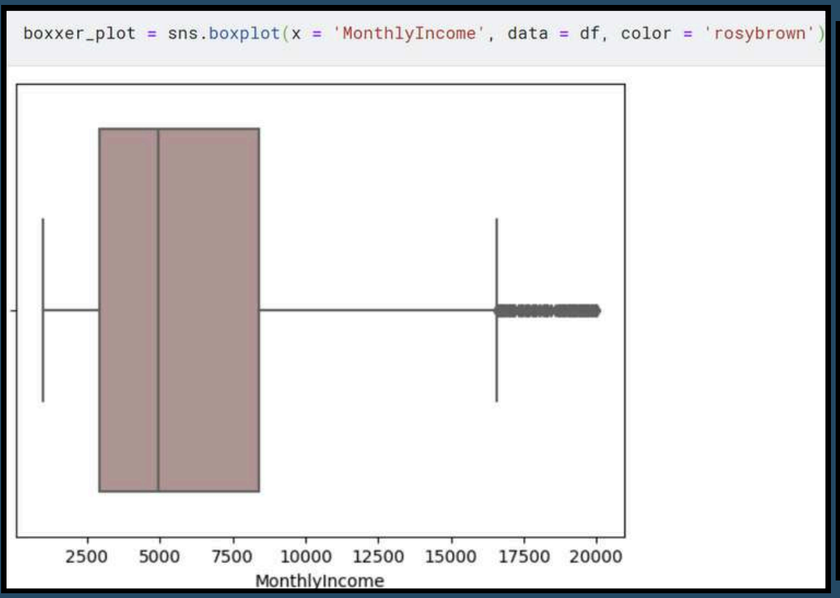
```
<Axes: xlabel='PerformanceRating', ylabel='count'>
```



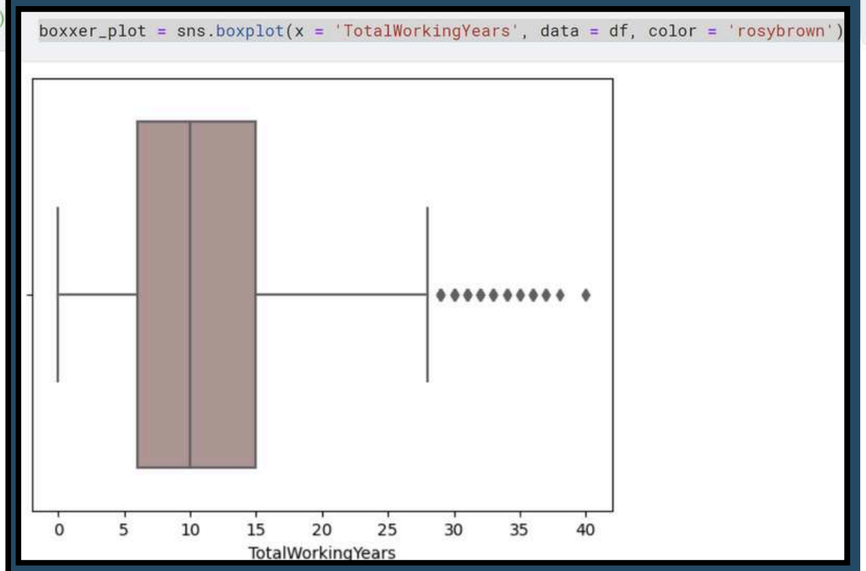
UNIVERSITAS TRISAKTI

DIAGRAM BOXPLOT

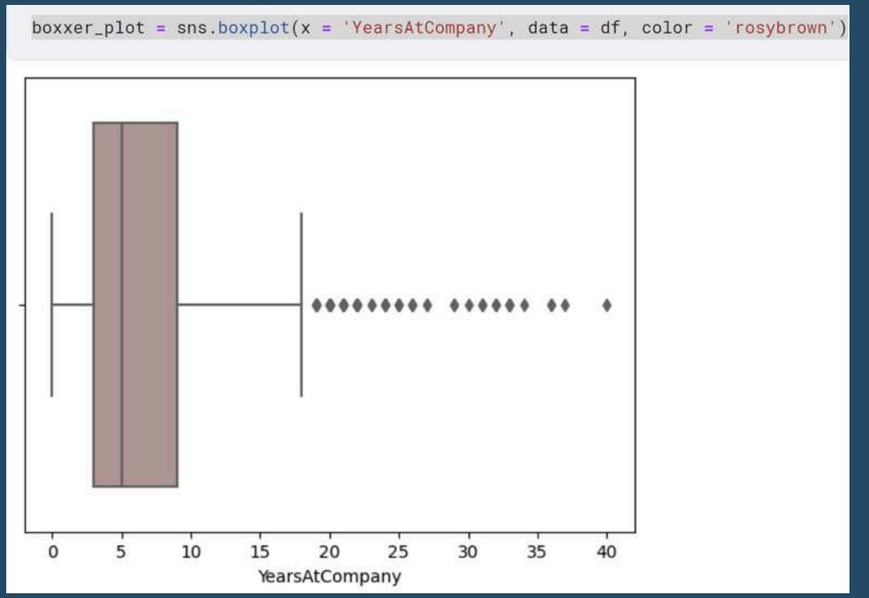
```
boxxer_plot = sns.boxplot(x = 'MonthlyIncome', data = df, color = 'rosybrown')
```




```
boxxer_plot = sns.boxplot(x = 'TotalWorkingYears', data = df, color = 'rosybrown')
```




```
boxxer_plot = sns.boxplot(x = 'YearsAtCompany', data = df, color = 'rosybrown')
```



UNIVERSITAS TRISAKTI

MENGHAPUS OUTLIERS

```
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
df_1 = df[~((df < (Q1 - 1.5 * IQR )) | (df > (Q3 + 1.5* IQR))).any(axis=1)]
print(df.shape)
print(df_1.shape)
```

```
(1480, 7)
(1366, 7)
```

MENGECEK MISSING VALUES

UNIVERSITAS TRISAKTI

```
print("Missing Values:")
print(df.isna().sum())
print("-" * 50)
print("Duplicated Data Count:", df.duplicated().sum())
```

```
Missing Values:
Age                    0
EmployeeNumber         0
JobLevel               0
MonthlyIncome          0
PerformanceRating      0
TotalWorkingYears      0
YearsAtCompany         0
dtype: int64
--------------------------------------------------
Duplicated Data Count: 10
```

PEMODELAN MACHINE LEARNING

UNIVERSITAS TRISAKTI

PREDIKSI MENGGUNAKAN RANDOM FOREST

UNIVERSITAS TRISAKTI

```
#SPLIT DATASET
x_train, x_test, y_train, y_test, = train_test_split(x,y, test_size=0.10)
```

```
from sklearn.ensemble import RandomForestClassifier
random_forest_model2 = RandomForestClassifier()
random_forest_model2.fit(x_train, y_train)
```

```
RandomForestClassifier
RandomForestClassifier()
```

```
# Melakukan prediksi dengan model Random Forest Classifier
y_predicted = random_forest_model.predict(x_test)

# Menampilkan 10 prediksi pertama
print(y_predicted[:10])
```

```
[3 3 3 3 3 3 3 3 3 3]
```

```
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report

model1=RandomForestClassifier()
# Matriks kebingungan (confusion matrix)
confusion = confusion_matrix(y_test, y_predicted)
print('\nConfusion Matrix:')
print(confusion)

# Akurasi
accuracy = accuracy_score(y_test, y_predicted)
print('\nAccuracy:')
print(accuracy)

# Laporan klasifikasi
classification_rep = classification_report(y_test, y_predicted)
print('\nClassification Report:')
print(classification_rep)
```

UNIVERSITAS TRISAKTI

HASIL PREDIKSI
RANDOM FOREST

```
Confusion Matrix:
[[130   0]
 [ 17   1]]

Accuracy:
0.8851351351351351

Classification Report:
              precision    recall  f1-score   support

           3       0.88      1.00      0.94       130
           4       1.00      0.06      0.11        18

    accuracy                           0.89       148
   macro avg       0.94      0.53      0.52       148
weighted avg       0.90      0.89      0.84       148
```

UNIVERSITAS TRISAKTI

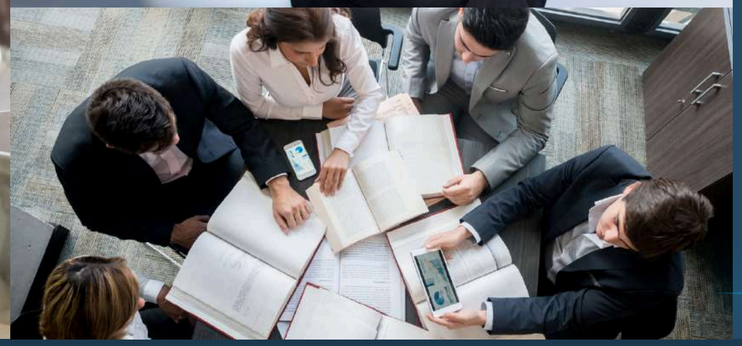
PREDIKSI MENGGUNAKAN DECISION TREE

```python
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report

# Matriks kebingungan (confusion matrix)
confusion = confusion_matrix(y_test, y_predicted)
print('\nConfusion Matrix:')
print(confusion)

# Akurasi
accuracy = accuracy_score(y_test, y_predicted)
print('\nAccuracy:')
print(accuracy)

# Laporan klasifikasi
classification_rep = classification_report(y_test, y_predicted)
print('\nClassification Report:')
print(classification_rep)
```

HASIL PREDIKSI DECISION TREE

```
Confusion Matrix:
[[216  38]
 [ 34   8]]

Accuracy:
0.7567567567567568

Classification Report:
              precision    recall  f1-score   support

           3       0.86      0.85      0.86       254
           4       0.17      0.19      0.18        42

    accuracy                           0.76       296
   macro avg       0.52      0.52      0.52       296
weighted avg       0.77      0.76      0.76       296
```

PEMODELAN
NN & DEEP LEARNING



UNIVERSITAS TRISAKTI

PREDIKSI MENGGUNAKAN MLP CLASSIFIER

UNIVERSITAS TRISAKTI

```python
#model MLP Classifier
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

# Misalkan Anda memiliki data X dan y
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20)

# Inisialisasi dan melatih model MLPClassifier
mlp = MLPClassifier(hidden_layer_sizes=(4,25,85), max_iter=1000)

#melakukan training
mlp.fit(x_train, y_train)

# Melakukan prediksi pada data uji
y_predicted = mlp.predict(x_test)

# Matriks kebingungan (confusion matrix)
print('\nConfusion Matrix:')
print(confusion_matrix(y_test, y_predicted))

# Akurasi
print('\nAccuracy:')
print(accuracy_score(y_test, y_predicted))

# Laporan klasifikasi
print('\nClassification Report:')
print(classification_report(y_test, y_predicted))
```

HASIL PREDIKSI
MLP CLASSIFIER

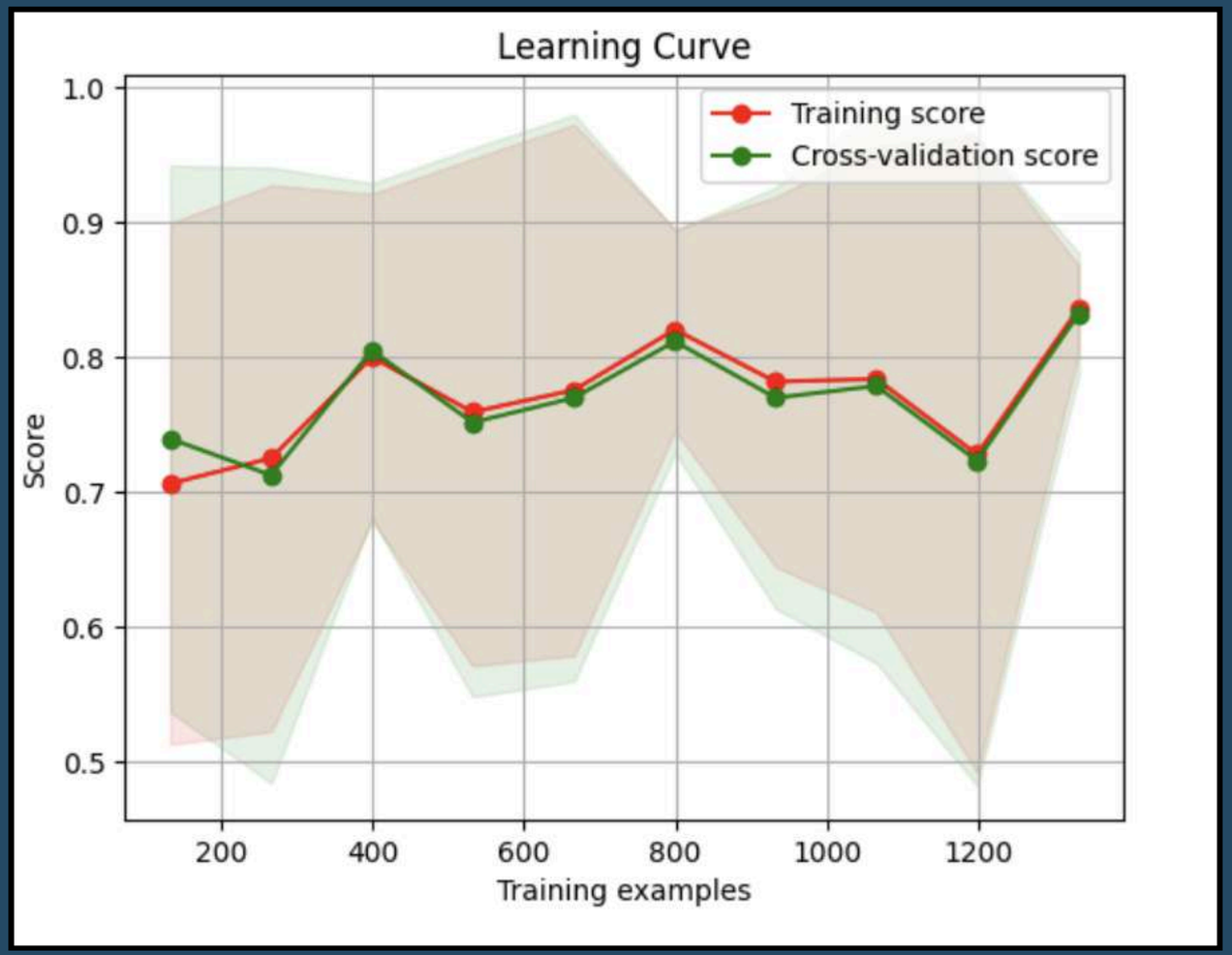
UNIVERSITAS TRISAKTI

```
Confusion Matrix:
[[263   0]
 [ 33   0]]

Accuracy:
0.8885135135135135

Classification Report:
              precision    recall  f1-score   support

           3       0.89      1.00      0.94       263
           4       0.00      0.00      0.00        33

    accuracy                           0.89       296
   macro avg       0.44      0.50      0.47       296
weighted avg       0.79      0.89      0.84       296
```

UNIVERSITAS TRISAKTI

HASIL EVALUASI MENGGUNAKAN LEARNING CURVE

MODEL

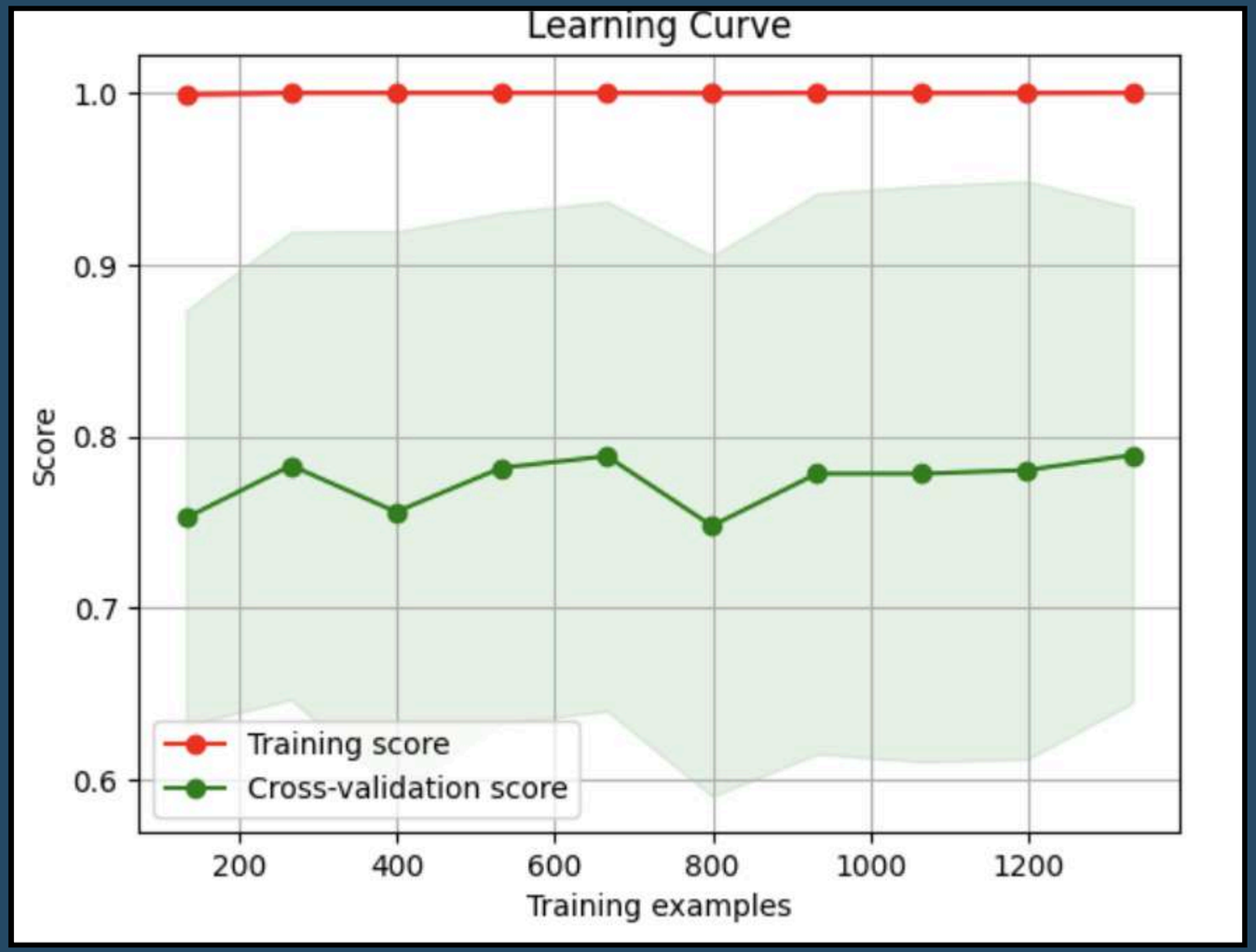


MLP CLASSIFIER

UNIVERSITAS TRISAKTI

HASIL EVALUASI MENGGUNAKAN LEARNING CURVE

MODEL



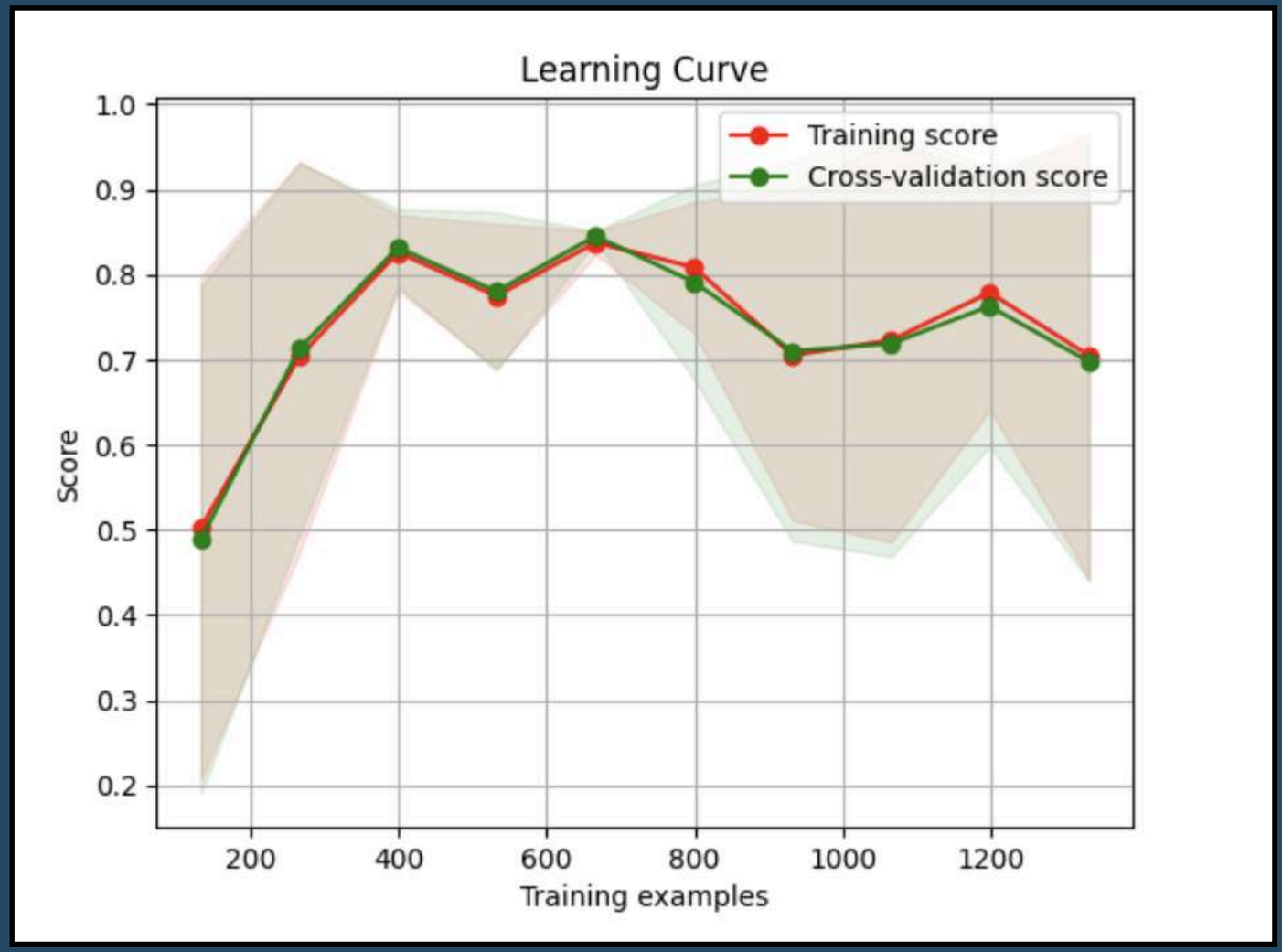
RANDOM FOREST

UNIVERSITAS TRISAKTI

HASIL EVALUASI MENGGUNAKAN LEARNING CURVE

MODEL

DECISION TREE



REFINING
SISTEM CERDAS

UNIVERSITAS TRISAKTI

OVERSAMPLING PADA KOLOM TARGET

UNIVERSITAS TRISAKTI

```python
from sklearn.utils import resample

# Membuat dua DataFrame terpisah untuk kelas mayoritas dan minoritas
df_majority = df[df['PerformanceRating'] == 3]
df_minority = df[df['PerformanceRating'] == 4]

# Oversampling pada kelas mayoritas
df_minority_upsampled = resample(df_minority,
                                 replace=True,      # Sampling tanpa penggantian
                                 n_samples=1253,  # Memastikan jumlah sampel yang sama
                                 random_state=42)    # Untuk hasil yang dapat direprodu

# Menggabungkan kelas mayoritas yang telah dioversample dengan kelas minoritas
df_up = pd.concat([df_minority_upsampled, df_majority])
```

```python
df_raw['PerformanceRating'].value_counts()
```

```
PerformanceRating
3    1253
4     227
Name: count, dtype: int64
```

```python
df_up['PerformanceRating'].value_counts()
```

```
PerformanceRating
4    1253
3    1253
Name: count, dtype: int64
```

UNDERSAMPLING PADA KOLOM TARGET

UNIVERSITAS TRISAKTI

```python
from sklearn.utils import resample
import pandas as pd

# Membuat dua DataFrame terpisah untuk kelas mayoritas dan minoritas
df_majority = df[df['PerformanceRating'] == 3]
df_minority = df[df['PerformanceRating'] == 4]

# Menentukan jumlah sampel yang ingin dihasilkan untuk kelas mayoritas (di sini, kita
n_samples_minority = len(df_minority)
n_samples_majority = n_samples_minority

# Melakukan undersampling pada kelas mayoritas
df_majority_undersampled = resample(df_majority,
                                    replace=False,     # Sampling tanpa penggantian (
                                    n_samples=n_samples_majority,  # Jumlah sampel ya
                                    random_state=42)   # Untuk hasil yang dapat direp

# Menggabungkan kelas mayoritas yang telah diundersample dengan kelas minoritas
df_up = pd.concat([df_majority_undersampled, df_minority])
```

```python
df_raw['PerformanceRating'].value_counts()
```

```
PerformanceRating
3    1253
4     227
Name: count, dtype: int64
```

```python
df_up['PerformanceRating'].value_counts()
```

```
PerformanceRating
3    227
4    227
Name: count, dtype: int64
```

DIAGRAM BARPLOT SETELAH OVERSAMPLING

UNIVERSITAS TRISAKTI

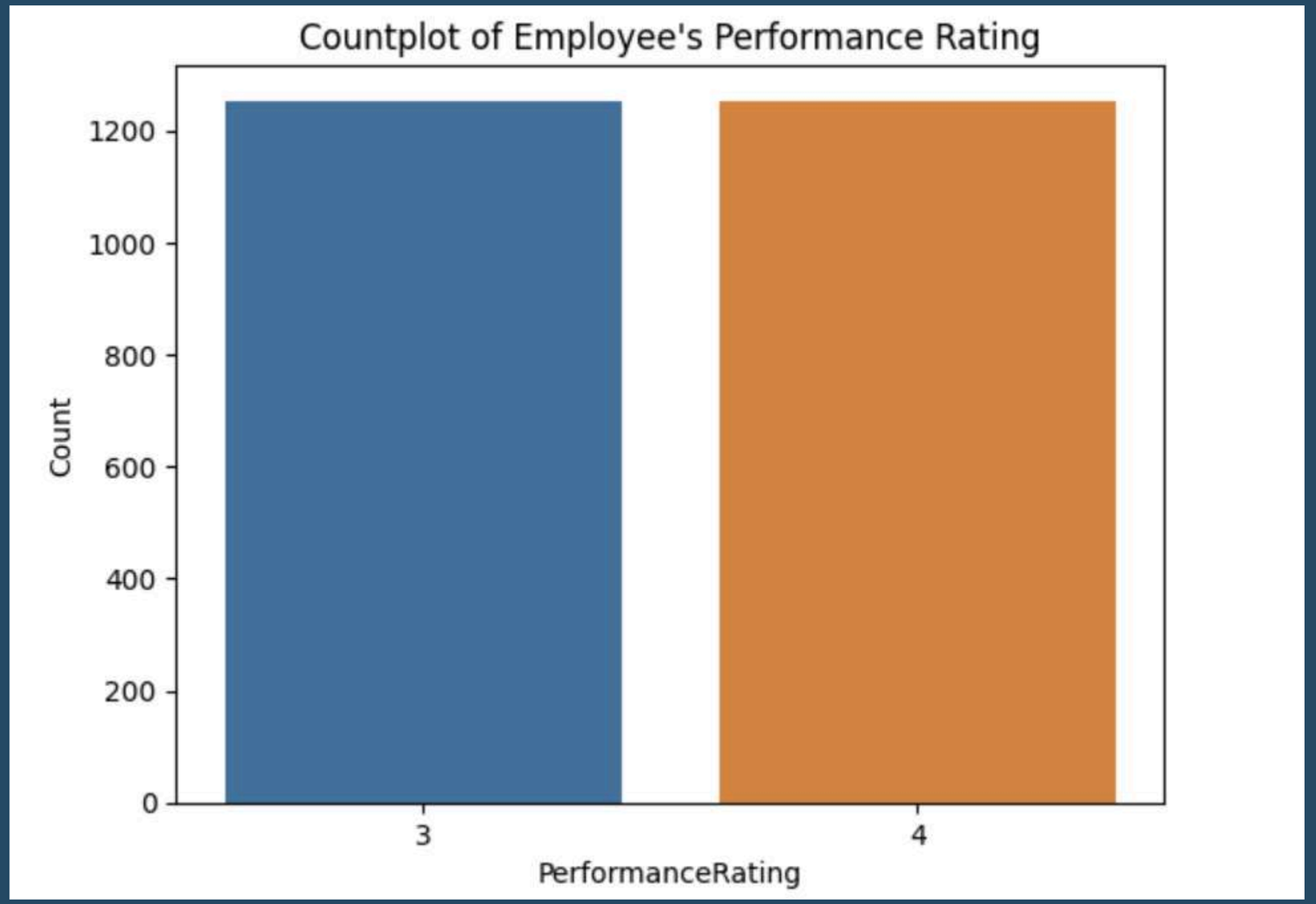
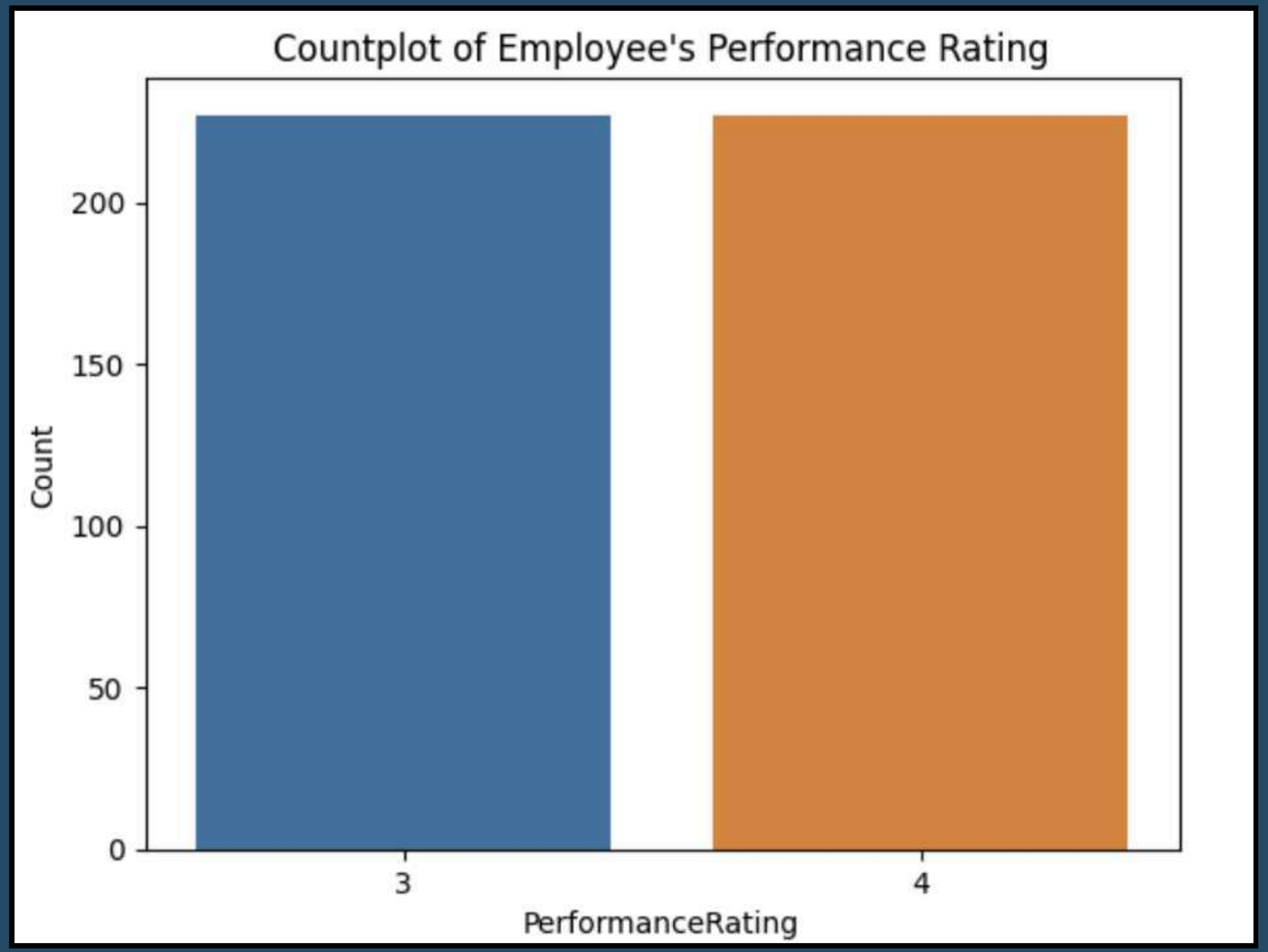


DIAGRAM BARPLOT SETELAH UNDERSAMPLING

UNIVERSITAS TRISAKTI



ANALISIS HASIL

UNIVERSITAS TRISAKTI

PERBANDINGAN AKURASI

UNIVERSITAS TRISAKTI

Model ML	Sebelum Di-Refining	Oversampling	Undersampling
Random Forest	88%	96%	47%
Decision Tree	75%	92%	56%
Naïve Bayes	87.5%	54.18%	42%
MLP Classifier	88%	49%	57%
KNN-Neighbors	79%	80%	47%
Logistic Regression	85%	49%	45%

UNIVERSITAS TRISAKTI

KOMPARASI MODEL ML

Decision Tree

Menghasilkan accuracy sebesar 0.75 atau **75%,** kemudian setelah dilakukan teknik oversampling pada dataset didapatkan hasil accuracy sebesar **92%**

Random Forest

Menghasilkan accuracy sebesar **88.51%**, kemudian setelah dilakukan teknik oversampling pada dataset didapatkan hasil accuracy sebesar **96%**

MLP Classifier

Menghasilkan accuracy sebesar **88.85%,** setelah dilakukan teknik oversampling pada dataset didapatkan hasil accuracy sebesar **54%**

UNIVERSITAS TRISAKTI

KESIMPULAN

Kesimpulan dari hasil percobaan prediksi Rating Karyawan yang kami lakukan pada dataset ini, kami memutuskan untuk memilih model Machine Learning **Random Forest** (sesudah dataset di-oversampling) untuk memprediksi Rating Karyawan, karena model tersebut memiliki nilai **akurasi tertinggi**, yang mana model tersebut memberikan hasil Rating Karyawan dengan baik.

VALIDASI ANTARA HASIL DENGAN MEANINGFUL OBJECTIVE

Meaningful Objective :

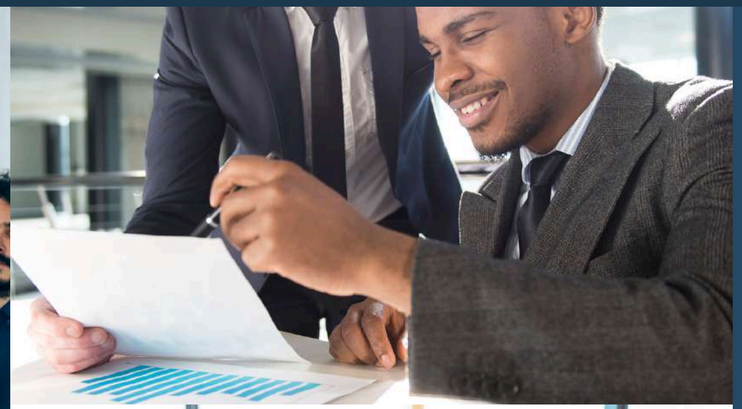
Menghasilkan hasil prediksi Rating Karyawan dengan baik agar **meningkatkan produktivitas** dan **kinerja karyawan**, serta membantu perusahaan dalam **mencapai target bisnis** dengan mengukur kinerja karyawan - karyawannya.

Hasil :

Berdasarkan model Machine Learning yang kita gunakan didapatkan **hasil akurasi** sebesar **96%**, yang menunjukkan bahwa tercapainya elemen Meaningful Objective yang kami inginkan.

UNIVERSITAS TRISAKTI

RESIKO & MITIGASI



UNIVERSITAS TRISAKTI

RESIKO DAN MITIGASI

Resiko

Kemungkinan terjadi **overfitting** terutama jika terlalu banyak pohon yang digunakan di dalam ensemble

Mitigasi

- Menetapkan jumlah pohon (**n_estimators**) untuk memastikan tidak ada overfitting yang signifikan
- Gunakan parameter **Max Depth** untuk mengendalikan tingkat overfitting

RESIKO DAN MITIGASI

Resiko

Ketidakseimbangan yang Berlebihan

Mitigasi

- Pertimbangkan teknik oversampling yang lebih moderat dan evaluasi dampaknya pada hasil model.

RESIKO DAN MITIGASI

Resiko

Kesalahan dalam Situasi yang Kritis

Mitigasi

- Gunakan metrik evaluasi seperti F1-score, precision, dan recall untuk memahami kinerja model dengan lebih baik.

DEPLOY MODEL

UNIVERSITAS TRISAKTI

TEAM 5

Thank You

UNIVERSITAS TRISAKTI