Ujian Praktik Fundamental Artificial Intelligence, Machine Learning, and Deep Learning

AI MASTERY_KM 4

Identitas

Nama Lengkap: Giselle Halim
Kode File: AIM0116F2202_ML_02 (Kode Program ML 2)
Kelas: Tensor

Deskripsi Kode

Kode ini berisi perancangan model algoritma K-Nearest Neighbor (K-NN) yang dilakukan dengan 2 cara, penulisan kode manual dengan rumus euclidian distance dan dengan model K-NN bawaan library scikit-learn. Kedua model K-NN yang dibuat memiliki nilai K=5, berarti data akan berkelompok dengan 5 tetangganya. Tujuan dari pembuatan 2 model tersebut adalah untuk melakukan perbandingan antara 2 model tersebut.

Dataset yang digunakan adalah tentang social network advertisement. Data ini akan diproses oleh model algoritma untuk memprediksi apakah seseorang akan membeli suatu produk yang diiklankan di social media berdasarkan usia dan perkiraan gaji. Dataset yang digunakan dibagi menjadi data training dan data testing dengan rasio 75:25. Untuk meningkatkan akurasi model, dilakukan feature scaling dengan StandardScaler.

Deskripsi Kode

Setelah model dilatih dengan data training, dilakukan perhitungan confusion matrix untuk membandingkan akurasi kedua model.
Terakhir, model dijalankan pada data testing dan hasilnya divisualisasikan dengan scatter plot dari matplotlib.

Flowchart

Link Flowchart Kode Awal

Link Flowchart Kode Modifikasi

Analisis Perbandingan Hasil Modifikasi Program

Pada kode hasil modifikasi, ditambahkan hal-hal sebagai berikut:

- Ditambahkan kode untuk melihat 5 data awal dan terakhir, mengecek tipe data, menghitung jumlah baris dan kolom, serta melakukan data cleaning dengan melihat apakah ada baris yang kosong.

- Ditambahkan Exploratory Data Analysis yang berisi eksplorasi univariate pada variabel beserta visualisasinya, melihat nilai min-max, dan deskripsi statistik lainnya. Ada juga perhitungan dan visualisasi korelasi antar feature untuk melihat seberapa kuat hubungan antar feature.

Analisis Perbandingan Hasil Modifikasi Program

- Adanya tambahan library yang digunakan, yaitu Seaborn untuk melakukan visualisasi data.
- Pengukuran untuk mengevaluasi model ditambahkan adanya classification report yang menunjukkan pengukuran accuracy, precision, recall, dan F1 score.
- Ditambahkan penjabaran hasil prediksi dari kedua model.

Dokumentasi Modifikasi

- Melihat 5 row terakhir pada dataset
- Melihat informasi kolom pada dataset
- Memeriksa apakah ada data null (pada dataset tidak ada data null/kosong)

```
[5]  dataset.tail()
```

	Age	EstimatedSalary	Purchased
395	46	41000	1
396	51	23000	1
397	50	20000	1
398	36	33000	0
399	49	36000	1

```
[6]  #General dataset information
     dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 3 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Age              400 non-null    int64
 1   EstimatedSalary  400 non-null    int64
 2   Purchased        400 non-null    int64
dtypes: int64(3)
memory usage: 9.5 KB
```

```
[7]  #Check the data for null values
     dataset.isnull().sum()
```

```
Age              0
EstimatedSalary  0
Purchased        0
dtype: int64
```

Melakukan Exploratory Data Analysis (EDA):

- Melihat value min dan max pada kolom ‘Age’ dan ‘Salary’
- Melihat kalkulasi statistika pada dataset
- Melihat distribusi kelas target pada dataset
- Melihat distribusi kolom ‘Age’ dan ‘Salary’

```
[8]  #Minimum/Maximum Age
     min_age = min(dataset['Age'])
     max_age = max(dataset['Age'])

     print("Min age: ",min_age)
     print("Max age: ",max_age)
```

```
Min age:  18
Max age:  60
```

```
[9]  #Minimum/Maximum Estimated Salary
     min_salary = min(dataset['EstimatedSalary'])
     max_salary = max(dataset['EstimatedSalary'])

     print("Min salary: ",min_salary)
     print("Max salary: ",max_salary)
```

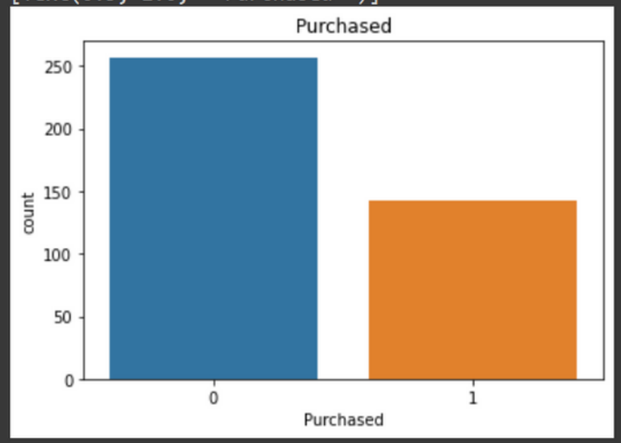
```
Min salary:  15000
Max salary:  150000
```

```
#Statistical Calculation
dataset[['Age','EstimatedSalary']].describe()
```

	Age	EstimatedSalary
count	400.000000	400.000000
mean	37.655000	69742.500000
std	10.482877	34096.960282
min	18.000000	15000.000000
25%	29.750000	43000.000000
50%	37.000000	70000.000000
75%	46.000000	88000.000000
max	60.000000	150000.000000

```
import seaborn as sns
#Univariate analysis purchase
sns.countplot(dataset['Purchased']).set (title=' Purchased ')
```

```
[Text(0.5, 1.0, ' Purchased ')]
```

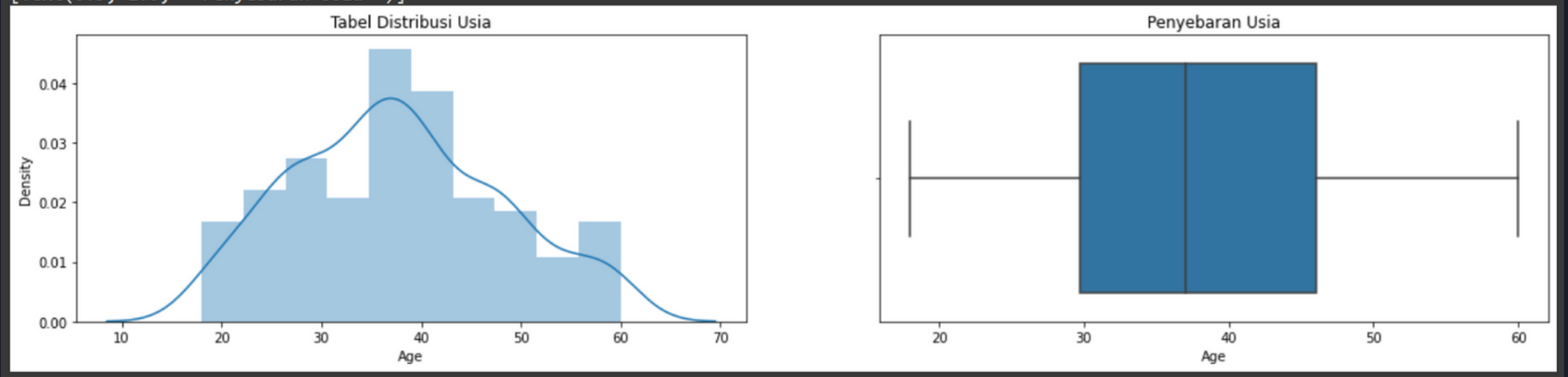


```
#Univariate analysis age
f = plt.figure(figsize=(20,4))

f.add_subplot(1,2,1)
sns.distplot(dataset['Age']).set (title=' Tabel Distribusi Usia ')

f.add_subplot(1,2,2)
sns.boxplot(dataset['Age']).set (title=' Penyebaran Usia ')
```

```
[Text(0.5, 1.0, ' Penyebaran Usia ')]
```

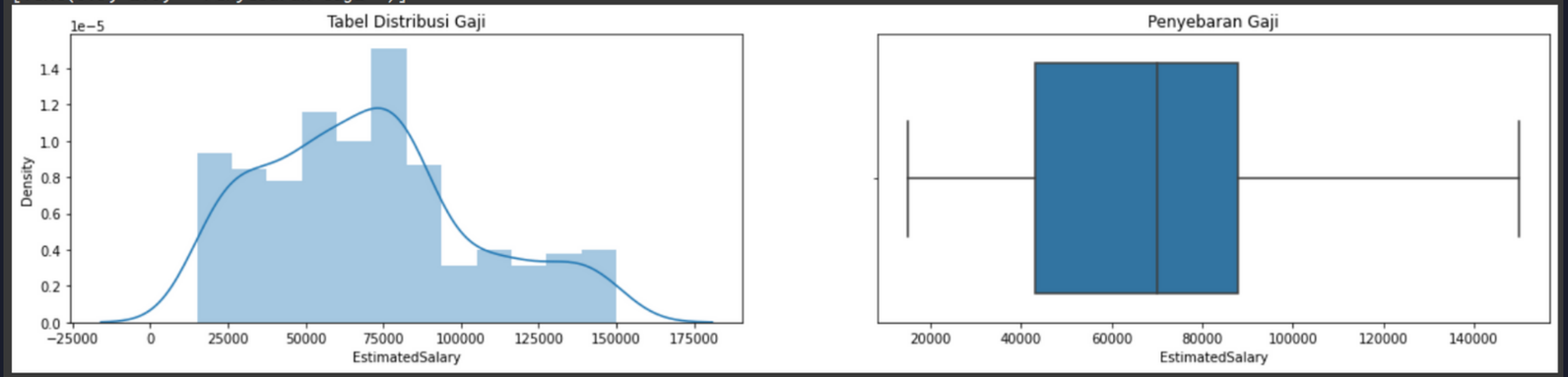



```
#Univariate analysis estimated salary
f = plt.figure(figsize=(20,4))

f.add_subplot(1,2,1)
sns.distplot(dataset['EstimatedSalary']).set (title=' Tabel Distribusi Gaji ')

f.add_subplot(1,2,2)
sns.boxplot(dataset['EstimatedSalary']).set (title=' Penyebaran Gaji ')
```

```
[Text(0.5, 1.0, ' Penyebaran Gaji ')]
```



```python
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test) #avoid data leakage
```

Melakukan feature scaling untuk algotima KNN

```python
from math import sqrt
class KNN():
  def __init__(self,k):
    self.k=k
    print(self.k)
  def fit(self,X_train,y_train):
    self.x_train=X_train
    self.y_train=y_train
  def calculate_euclidean(self,sample1,sample2):
    distance=0.0
    for i in range(len(sample1)):
      distance+=(sample1[i]-sample2[i])**2 #Euclidean Distance = sqrt(sum i to N (x1_i - x2_i)^2)
    return sqrt(distance)
  def nearest_neighbors(self,test_sample):
    distances=[]#calculate distances from a test sample to every sample in a training set
    for i in range(len(self.x_train)):
      distances.append((self.y_train[i],self.calculate_euclidean(self.x_train[i],test_sample)))
    distances.sort(key=lambda x:x[1])#sort in ascending order, based on a distance value
    neighbors=[]
    for i in range(self.k): #get first k samples
      neighbors.append(distances[i][0])
    return neighbors
```

Model KNN yang dikodekan sendiri

```
#Correlation across features
dataset.corr().style.background_gradient().set_precision(2)
```

	Age	EstimatedSalary	Purchased
Age	1.00	0.16	0.62
EstimatedSalary	0.16	1.00	0.36
Purchased	0.62	0.36	1.00

```
print('Our model predictions: ',predictions)
```

```
Our model predictions:  [0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0,
```

```
print('Sklearn model predictions: ',y_pred)
```

```
Sklearn model predictions:  [0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 1 0 (
 0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 1 1 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 1 (
 0 0 0 0 1 1 1 1 0 0 1 0 0 1 1 0 0 1 0 0 0 0 0 1 1 1]
```

```
#Classification report
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.96      0.94      0.95        68
           1       0.88      0.91      0.89        32

    accuracy                           0.93       100
   macro avg       0.92      0.92      0.92       100
weighted avg       0.93      0.93      0.93       100
```

```
from sklearn.metrics import classification_report
print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           0       0.96      0.94      0.95        68
           1       0.88      0.91      0.89        32

    accuracy                           0.93       100
   macro avg       0.92      0.92      0.92       100
weighted avg       0.93      0.93      0.93       100
```

- *y_pred: prediksi model sklearn, predictions: prediksi model yang ditulis sendiri

Kesimpulan

Kesimpulan dari hasil kode yang telah dimodifikasi adalah:
1 .Tidak ada data yang kosong/bernilai null pada dataset, sehingga dataset bersih. Jumlah data pada dataset adalah 400 rows (tidak termasuk header).

2 .Hasil EDA menunjukkan bahwa:

- Rata-rata usia adalah 37.7 tahun dan rata-rata gaji adalah $69,74k
- Usia termuda pada dataset adalah 18 tahun dan yang tertua adalah 60 tahun.
- Kebanyakan orang pada dataset berusia sekitar 30-45 tahun dan paling sedikit berusia sekitar 50-55 tahun.
- Gaji terendah pada dataset adalah $15k dan yang tertinggi adalah $150k
- Untuk gaji, kebanyakan orang pada dataset memiliki gaji sekitar $40k-80k dan menjadi semakin sedikit pada gaji >$90k.

Kesimpulan

3. Jumlah orang yang tidak membeli (0) dan membeli (1) cukup seimbang dengan perbedaan sekitar 100 orang dari keseluruhan jumlah 400 orang.

4. Hasil perhitungan korelasi antar features menunjukkan bahwa adanya hubungan linear positif yang cukup kuat antar Purchased dan Age (usia dan pembelian).

5. Model yang dibuat sendiri dan yang diimport dari sklearn memiliki hasil prediksi dan classification report yang sama persis. Confusion Matrix juga menunjukkan hasil yang sama. Ini berarti model yang dibuat tidak ada bedanya dengan model yang dipanggil oleh library.

THANK YOU

Google Colab Link