March 8, 2022

1 Rep 2. KNN

1.0.1

https://www.kaggle.com/rakeshrau/social-network-ads => Social_Network_Ads.csv

1.0.2

, ,

1.0.3

- User ID :
- Gender :
- Age :
- EstimatedSalary :
- Purchased : (0: / 1:)

1.0.4 1.

CSV

[ ]:
```
import pandas as pd
df = pd.read_csv('Social_Network_Ads.csv')
df
```

[ ]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
..	…	…	…	…	…
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398	15755018	Male	36	33000	0
399	15594041	Female	49	36000	1

[400 rows x 5 columns]

1.0.5 2.

input_data target_data

[ ]:
```
input_data = df[['Gender', 'Age', 'EstimatedSalary']].to_numpy()
target_data = df['Purchased'].to_numpy()
input_data
```

[ ]:
```
array([['Male', 19, 19000],
       ['Male', 35, 20000],
       ['Female', 26, 43000],
       …,
       ['Female', 50, 20000],
       ['Male', 36, 33000],
       ['Female', 49, 36000]], dtype=object)
```

- Zscore

[ ]:
```
#     (Z  )
import numpy as np

mean = np.mean(input_data, axis=0)
std = np.std(input_data, axis=0)

input_scaled = (input_data - mean) / std

input_scaled
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/Users/solstice/Desktop/Github/Inhatc-MachineLearning/ /2 /     .ipynb Cell 7' i␣
 ↪<cell line: 4>()
      <a href='vscode-notebook-cell:/Users/solstice/Desktop/Github/
 ↪Inhatc-MachineLearning/%EC%A0%95%EC%9C%A4/2%EC%A3%BC%EC%B0%A8/
 ↪%EC%86%8C%EC%85%9C%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC%EA%B4%91%EA%B3%A0.
 ↪ipynb#ch0000026?line=0'>1</a> #     (Z  )
      <a href='vscode-notebook-cell:/Users/solstice/Desktop/Github/
 ↪Inhatc-MachineLearning/%EC%A0%95%EC%9C%A4/2%EC%A3%BC%EC%B0%A8/
 ↪%EC%86%8C%EC%85%9C%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC%EA%B4%91%EA%B3%A0.
 ↪ipynb#ch0000026?line=1'>2</a> import numpy as np
----> <a href='vscode-notebook-cell:/Users/solstice/Desktop/Github/
 ↪Inhatc-MachineLearning/%EC%A0%95%EC%9C%A4/2%EC%A3%BC%EC%B0%A8/
 ↪%EC%86%8C%EC%85%9C%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC%EA%B4%91%EA%B3%A0.
 ↪ipynb#ch0000026?line=3'>4</a> mean = np.mean(input_data, axis=0)
      <a href='vscode-notebook-cell:/Users/solstice/Desktop/Github/
 ↪Inhatc-MachineLearning/%EC%A0%95%EC%9C%A4/2%EC%A3%BC%EC%B0%A8/
 ↪%EC%86%8C%EC%85%9C%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC%EA%B4%91%EA%B3%A0.
 ↪ipynb#ch0000026?line=4'>5</a> std = np.std(input_data, axis=0)
      <a href='vscode-notebook-cell:/Users/solstice/Desktop/Github/
 ↪Inhatc-MachineLearning/%EC%A0%95%EC%9C%A4/2%EC%A3%BC%EC%B0%A8/
 ↪%EC%86%8C%EC%85%9C%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC%EA%B4%91%EA%B3%A0.
 ↪ipynb#ch0000026?line=6'>7</a> input_scaled = (input_data - mean) / std
```

```
File <__array_function__ internals>:5, in mean(*args, **kwargs)

File /opt/homebrew/Caskroom/miniforge/base/envs/py39/lib/python3.9/site-package
 ↪numpy/core/fromnumeric.py:3440, in mean(a, axis, dtype, out, keepdims, where)
   <a href='file:///opt/homebrew/Caskroom/miniforge/base/envs/py39/lib/python3.
 ↪site-packages/numpy/core/fromnumeric.py?line=3436'>3437</a>     else:
   <a href='file:///opt/homebrew/Caskroom/miniforge/base/envs/py39/lib/python3.
 ↪site-packages/numpy/core/fromnumeric.py?line=3437'>3438</a>         return␣
 ↪mean(axis=axis, dtype=dtype, out=out, **kwargs)
-> <a href='file:///opt/homebrew/Caskroom/miniforge/base/envs/py39/lib/python3.
 ↪site-packages/numpy/core/fromnumeric.py?line=3439'>3440</a> return _methods.
 ↪_mean(a, axis=axis, dtype=dtype,
   <a href='file:///opt/homebrew/Caskroom/miniforge/base/envs/py39/lib/python3.
 ↪site-packages/numpy/core/fromnumeric.py?line=3440'>3441</a>                  ␣
 ↪    out=out, **kwargs)

File /opt/homebrew/Caskroom/miniforge/base/envs/py39/lib/python3.9/site-package
 ↪numpy/core/_methods.py:181, in _mean(a, axis, dtype, out, keepdims, where)
    <a href='file:///opt/homebrew/Caskroom/miniforge/base/envs/py39/lib/python3
 ↪9/site-packages/numpy/core/_methods.py?line=178'>179</a> ret = umr_sum(arr,␣
 ↪axis, dtype, out, keepdims, where=where)
    <a href='file:///opt/homebrew/Caskroom/miniforge/base/envs/py39/lib/python3
 ↪9/site-packages/numpy/core/_methods.py?line=179'>180</a> if isinstance(ret, m
 ↪ndarray):
--> <a href='file:///opt/homebrew/Caskroom/miniforge/base/envs/py39/lib/python3
 ↪9/site-packages/numpy/core/_methods.py?line=180'>181</a>     ret = um.
 ↪true_divide(
    <a href='file:///opt/homebrew/Caskroom/miniforge/base/envs/py39/lib/python3
 ↪9/site-packages/numpy/core/_methods.py?line=181'>182</a>             ret,␣
 ↪rcount, out=ret, casting='unsafe', subok=False)
    <a href='file:///opt/homebrew/Caskroom/miniforge/base/envs/py39/lib/python3
 ↪9/site-packages/numpy/core/_methods.py?line=182'>183</a>     if␣
 ↪is_float16_result and out is None:
    <a href='file:///opt/homebrew/Caskroom/miniforge/base/envs/py39/lib/python3
 ↪9/site-packages/numpy/core/_methods.py?line=183'>184</a>         ret = arr.
 ↪dtype.type(ret)

TypeError: unsupported operand type(s) for /: 'str' and 'int'
```

- TypeError: unsupported operand type(s) for /: ‘str’ and ‘int’
- .
- Male 0, Female 1 .
- LabelEncoder .
- https://steadiness-193.tistory.com/243
- encoder.fit() .
- Gender .

[ ]:
```
from sklearn.preprocessing import LabelEncoder

encoder = LabelEncoder()
encoder.fit(df['Gender'])
gender_encoded = encoder.transform(df['Gender'])
gender_encoded
```

[ ]:
```
array([1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0,
       1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1,
       0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1,
       1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0,
       1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0,
       0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1,
       1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0,
       1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0,
       0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0,
       1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1,
       0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0,
       1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0,
       0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0,
       1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1,
       0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1,
       0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0,
       1, 0, 1, 0])
```

- Gender input_data .

[ ]:
```
gender_encoded_df = pd.DataFrame(gender_encoded, columns=['Gender'])
df['Gender'] = gender_encoded_df
input_data = df[['Gender', 'Age', 'EstimatedSalary']].to_numpy()
input_data
```

[ ]:
```
array([[    1,    19, 19000],
       [    1,    35, 20000],
       [    0,    26, 43000],
       …,
       [    0,    50, 20000],
       [    1,    36, 33000],
       [    0,    49, 36000]])
```

Zscore

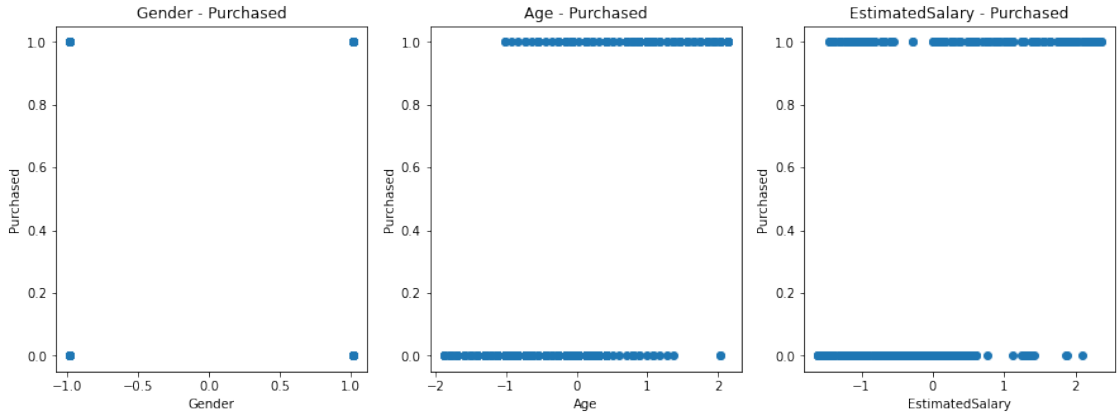
[ ]:
```
#     (Z  )
import numpy as np

mean = np.mean(input_data, axis=0)
```

```
std = np.std(input_data, axis=0)

input_scaled = (input_data - mean) / std

input_scaled
```

[ ]:
```
array([[ 1.02020406, -1.78179743, -1.49004624],
       [ 1.02020406, -0.25358736, -1.46068138],
       [-0.98019606, -1.11320552, -0.78528968],
       …,
       [-0.98019606,  1.17910958, -1.46068138],
       [ 1.02020406, -0.15807423, -1.07893824],
       [-0.98019606,  1.08359645, -0.99084367]])
```

1.0.6 3.

[ ]:
```
import matplotlib.pyplot as plt

plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
plt.title('Gender - Purchased')
plt.scatter(input_scaled[:, 0], target_data)
plt.xlabel('Gender')
plt.ylabel('Purchased')

plt.subplot(1, 3, 2)
plt.title('Age - Purchased')
plt.scatter(input_scaled[:, 1], target_data)
plt.xlabel('Age')
plt.ylabel('Purchased')

plt.subplot(1, 3, 3)
plt.title('EstimatedSalary - Purchased')
plt.scatter(input_scaled[:, 2], target_data)
plt.xlabel('EstimatedSalary')
plt.ylabel('Purchased')

plt.show()
```



1.0.7 4.

train test ,

[ ]:
```
from sklearn.model_selection import train_test_split
train_input, test_input, train_target, test_target = train_test_split(
    input_scaled, target_data, stratify=target_data, random_state=45)

from sklearn.neighbors import KNeighborsClassifier

kn = KNeighborsClassifier()
kn.fit(train_input, train_target)
kn.score(test_input, test_target)
```

[ ]:
```
0.94
```

1.0.8 5.

[ ]:
```
pred_data = np.array([[0, 35, 140000]])
pred_scaled = (pred_data - mean) / std

kn.predict(pred_scaled)
```

[ ]:
```
array([1])
```