

Titanic using PCA, kNN

PinkLAB Edu

11 November, 2025

Table of Contents

1	Titanic	3
1.1	다시 타이타닉으로~	3
1.2	이름 분리해서 title 만들고	4
1.3	귀족과 평민 등급을 구별하고	5
1.4	gender 컬럼 생성	5
1.5	grade 컬럼 생성	5
1.6	현재까지 데이터 정리	6
1.7	null이 아닌 데이터만	7
1.8	데이터 나누고	7
1.9	pca 적용 준비	8
1.10	두 개의 축으로 변환	8
1.11	그려보자	9
1.12	그럼 세 개로	9
1.13	그러서 확인해볼 준비	10
1.14	plotly.express로	10
1.15	결과	11
1.16	pipe line 구축	11
1.17	다시 확인	12

1 Titanic

1.1 다시 타이타닉으로~

```
import pandas as pd

titanic_url = (
    "https://raw.githubusercontent.com/PinkWink/ML_tutorial" + \
    "/master/dataset/titanic.xls"
)

titanic = pd.read_excel(titanic_url)
titanic.head()
```

✓ 0.1s Python

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked
0	1	1	Allen, Miss. Elisabeth Walton	female	29.0000	0	0	24160	211.3375	B5	S
1	1	1	Allison, Master. Hudson Trevor	male	0.9167	1	2	113781	151.5500	C22 C26	S
2	1	0	Allison, Miss. Helen Loraine	female	2.0000	1	2	113781	151.5500	C22 C26	S
3	1	0	Allison, Mr. Hudson Joshua Creighton	male	30.0000	1	2	113781	151.5500	C22 C26	S
4	1	0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25.0000	1	2	113781	151.5500	C22 C26	S

1.2 이름 분리해서 title 만들고

```
import re

title = []

for idx, dataset in titanic.iterrows():
    title.append(re.search(r"\s\w+(\s\w+)?\.", dataset["name"]).group()[2:-1])

titanic["title"] = title

titanic.head()
```

✓ 0.0s

Python

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked
0	1	1	Allen, Miss. Elisabeth Walton	female	29.0000	0	0	24160	211.3375	B5	S
1	1	1	Allison, Master. Hudson Trevor	male	0.9167	1	2	113781	151.5500	C22 C26	S
2	1	0	Allison, Miss. Helen Loraine	female	2.0000	1	2	113781	151.5500	C22 C26	S
3	1	0	Allison, Mr. Hudson Joshua Creighton	male	30.0000	1	2	113781	151.5500	C22 C26	S
4	1	0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25.0000	1	2	113781	151.5500	C22 C26	S

1.3 귀족과 평민 등급을 구별하고

```
titanic["title"] = titanic["title"].replace("Mlle", "Miss")
titanic["title"] = titanic["title"].replace("Ms", "Miss")
titanic["title"] = titanic["title"].replace("Mme", "Mrs")

Rare_f = ["Dona", "Dr", "Lady", "the Countess"]
Rare_m = ["Capt", "Col", "Don", "Major", "Rev", "Sir", "Jonkheer", "Master"]

for each in Rare_f:
    titanic["title"] = titanic["title"].replace(each, "Rare_f")

for each in Rare_m:
    titanic["title"] = titanic["title"].replace(each, "Rare_m")

titanic["title"].unique()
```

✓ 0.0s

Python

```
array(['Miss', 'Rare_m', 'Mr', 'Mrs', 'Rare_f'], dtype=object)
```

1.4 gender 컬럼 생성

```
from sklearn.preprocessing import LabelEncoder

le_sex = LabelEncoder()
le_sex.fit(titanic["sex"])
titanic["gender"] = le_sex.transform(titanic["sex"])

le_sex.classes_
```

✓ 0.5s

Python

```
array(['female', 'male'], dtype=object)
```

1.5 grade 컬럼 생성

```
le_grade = LabelEncoder()
le_grade.fit(titanic["title"])
titanic["grade"] = le_grade.transform(titanic["title"])

le_grade.classes_
```

✓ 0.0s

Python

```
array(['Miss', 'Mr', 'Mrs', 'Rare_f', 'Rare_m'], dtype=object)
```

1.6 현재까지 데이터 정리

titanic.head()

✓ 0.0s Python

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked
0	1	1	Allen, Miss. Elisabeth Walton	female	29.0000	0	0	24160	211.3375	B5	S
1	1	1	Allison, Master. Hudson Trevor	male	0.9167	1	2	113781	151.5500	C22 C26	S
2	1	0	Allison, Miss. Helen Loraine	female	2.0000	1	2	113781	151.5500	C22 C26	S
3	1	0	Allison, Mr. Hudson Joshua Creighton	male	30.0000	1	2	113781	151.5500	C22 C26	S
4	1	0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25.0000	1	2	113781	151.5500	C22 C26	S

1.7 null이 아닌 데이터만

```
titanic = titanic[titanic["age"].notnull()]
titanic = titanic[titanic["fare"].notnull()]

titanic.info()
```

Python

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1045 entries, 0 to 1308
Data columns (total 17 columns):
#   Column      Non-Null Count  Dtype
---  -
0   pclass      1045 non-null   int64
1   survived    1045 non-null   int64
2   name        1045 non-null   object
3   sex         1045 non-null   object
4   age         1045 non-null   float64
5   sibsp       1045 non-null   int64
6   parch       1045 non-null   int64
7   ticket      1045 non-null   object
8   fare        1045 non-null   float64
9   cabin       272 non-null    object
10  embarked    1043 non-null   object
11  boat        417 non-null    object
12  body        119 non-null    float64
13  home.dest    685 non-null    object
14  title       1045 non-null   object
15  gender      1045 non-null   int32
16  grade       1045 non-null   int32
dtypes: float64(3), int32(2), int64(4), object(8)
memory usage: 138.8+ KB
```

1.8 데이터 나누고

```
from sklearn.model_selection import train_test_split

X = titanic[["pclass", "age", "sibsp", "parch", "fare", "gender", "grade"]].astype(
    "float"
)

y = titanic["survived"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=13
)
```

✓ 0.0s

Python

1.9 pca 적용 준비

```
from sklearn.model_selection import train_test_split

X = titanic[["pclass", "age", "sibsp", "parch", "fare", "gender", "grade"]].astype(
    "float"
)

y = titanic["survived"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=13
)
```

✓ 0.0s

Python

```
from sklearn.decomposition import PCA

def get_pca_data(ss_data, n_components=2):
    pca = PCA(n_components=n_components)
    pca.fit(ss_data)

    return pca.transform(ss_data), pca
```

✓ 0.0s

Python

```
def get_pd_from_pca(pca_data, col_num):
    cols = ["pca_" + str(n) for n in range(col_num)]

    return pd.DataFrame(pca_data, columns=cols)
```

✓ 0.0s

Python

```
import numpy as np

def print_variance_ratio(pca, only_sum=False):
    if not only_sum :
        print("variance_ratio : ", pca.explained_variance_ratio_)
    print("sum of variance_ratio : ", np.sum(pca.explained_variance_ratio_))
```

✓ 0.0s

Python

1.10 두 개의 축으로 변환

```
pca_data, pca = get_pca_data(X_train, n_components=2)
print_variance_ratio(pca)
```

✓ 0.0s

Python

```
variance_ratio : [0.93577394 0.06326916]
sum of variance_ratio : 0.9990431009511271
```


1.11 그려보자

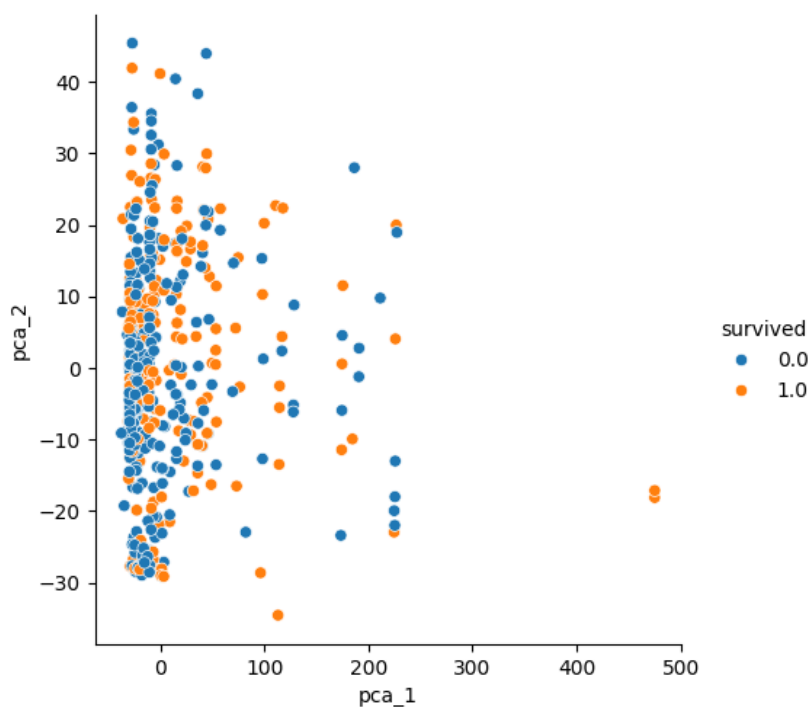
```
import seaborn as sns

pca_columns = ["pca_1", "pca_2"]
pca_pd = pd.DataFrame(pca_data, columns=pca_columns)
pca_pd["survived"] = y_train

sns.pairplot(pca_pd, hue="survived", height=5, x_vars=["pca_1"], y_vars=["pca_2"]);
```

✓ 0.2s

Python



1.12 그림 세 개로

```
pca_data, pca = get_pca_data(X_train, n_components=3)
print_variance_ratio(pca)
```

✓ 0.0s

Python

```
variance_ratio : [9.35773938e-01 6.32691630e-02 4.00903990e-04]
sum of variance_ratio : 0.9994440049413543
```

1.13 그래서 확인해볼 준비

```
pca_pd = get_pd_from_pca(pca_data, 3)

pca_pd["survived"] = y_train.values
pca_pd.head()
```

✓ 0.0s

Python

	pca_0	pca_1	pca_2	survived
0	-28.763184	4.479379	-0.451531	0
1	41.587362	22.084594	0.011834	0
2	-19.598979	-10.999936	0.558167	0
3	-28.232483	-6.559632	-1.349217	1
4	-29.055717	-1.510811	-0.538886	0

1.14 plotly.express로

```
import plotly.express as px

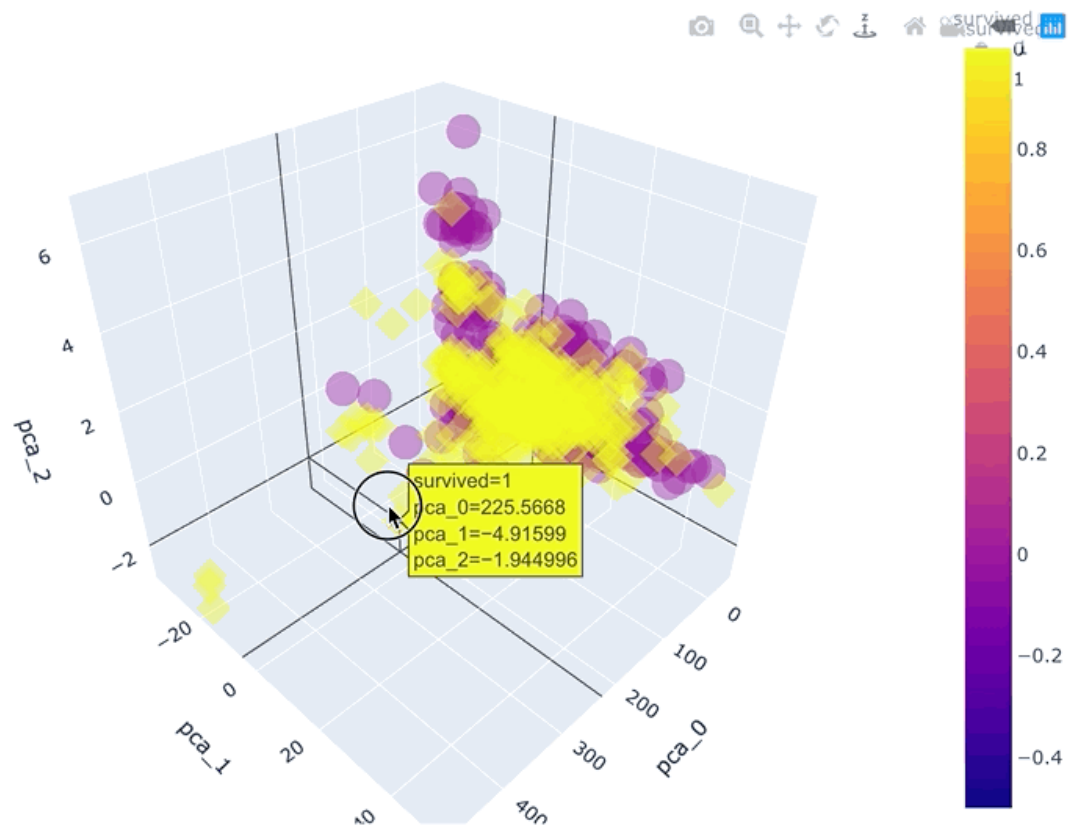
fig = px.scatter_3d(
    pca_pd,
    x="pca_0",
    y="pca_1",
    z="pca_2",
    color="survived",
    symbol="survived",
    opacity=0.4,
)

fig.update_layout(margin=dict(l=0, r=0, b=0, t=0))
fig.show()
```

✓ 0.0s

Python

1.15 결과



1.16 pipe line 구축

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler

estimators = [
    ("scaler", StandardScaler()),
    ("pca", PCA(n_components=3)),
    ("clf", KNeighborsClassifier(n_neighbors=20)),
]

pipe = Pipeline(estimators)
pipe.fit(X_train, y_train)

pred = pipe.predict(X_test)
print(accuracy_score(y_test, pred))
```

✓ 0.0s

Python

0.7703349282296651

1.17 다시 확인

```
dicaprio = np.array([[3, 18, 0, 0, 5, 1, 1]])  
print("Decapro : ", pipe.predict_proba(dicaprio)[0, 1])  
  
winslet = np.array([[1, 16, 1, 1, 100, 0, 3]])  
print("Winslet : ", pipe.predict_proba(winslet)[0, 1])
```

✓ 0.0s

Python

Decapro : 0.1
Winslet : 0.85