

혁신성장 청년인재 집중양성
이미지 분석 인공지능 서비스 개발 실무 과정

Semi-Project for Part 4

Training & testing traditional ML algorithms (Titanic survival analysis)

Daeyeon Jo
repositorator@gmail.com

본 교안은 멀티캠퍼스 교육을 위해 제작되었으며, 교육 외 배포/게시/공개를 금합니다.

1. Blank notebook for this semi-project

ML for Titanic survival prediction (Blank)

Logout

FileEditViewInsertCellKernelWidgetsHelp

TrustedPython 3

Save

+

Undo

Copy

Paste

Up

Down

Run

Interrupt

Restart

Code

Kernel

In [1]:

```
1 import warnings
2 warnings.filterwarnings("ignore")
3
4 import pandas as pd
5 import numpy as np
6 import matplotlib.pyplot as plt
7
8 # from sklearn import ?
9 # from sklearn.metrics import ?
```

1. Preparing dataset (2번부터 실습)

In [12]:

```
1 data_df = pd.read_csv('titanic.csv')
2 data_df.head(3)
```

Data info

- **PassengerId** : Unique ID of passenger
- **Survived** : 0 = No, 1 = Yes
- **pclass** : Ticket class (1 = 1st, 2 = 2nd, 3 = 3rd)
- **sibsp** : # of siblings & spouses aboard the Titanic
- **parch** : # of parents / children aboard the Titanic
- **ticket** : Ticket number
- **cabin** : Cabin number
- **embarked** : Port of Embarkation (C = Cherbourg, Q = Queenstown, S = Southampton)

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S

2. Possible pathways for data preprocessing

- + Check & adjust data for handling **Missing data & Outlier**
- + Select important columns (or just use all columns & improve your model later)
- + Change characters to numbers (Binary num, Class num, One-hot vector, etc.)
- + (If applicable & useful) **Select features** with Tree-based models
- + (If applicable & useful) **Modify the scale of features** with StandardScaler / MinMaxScaler
- + (If applicable & useful) **Reduce dimension** with PCA

서울시 범죄현황 통계자료 분석 및 시각화

2. 서울시 범죄현황 통계자료 분석 및 시각화

Last Checkpoint: an hour ago (unsaved changes)

File Edit View Insert Cell Kernel Help Trusted

서울시 범죄현황 통계자료 분석 및 시각화

```
In [1]: 1 import numpy as np
2 import pandas as pd
3 import seaborn as sns
4
5 import matplotlib.pyplot as plt
6 from matplotlib import font_manager, rc
```

1. 데이터 입력 및 데이터 전처리

```
In [2]: 1 df = pd.read_excel('관서별 5대범죄 발생 및 건수.xlsx', encoding='utf-8')
2 df.head()
```

	관서명	소계(발생)	소계(검거)	살인(발생)	살인(검거)	강도(발생)	강도(검거)	강간(발생)	강간(검거)	절도(발생)	절도(검거)	폭력
0	계	126481	82680	163	156	276	257	5449	55387	21842	652	
1	중부서	2868	1716	2	2	3	2	185	65	1395	477	135
2	중로서	2472	1589	3	3	6	5	115	98	1878	413	127
3	남대문서	2094	1226	1	0	6	4	65	46	1153	382	809
4	서대문서	4829	2579	2	2	5	4	154	124	1812	738	205

Scikit-learn practices & CheatSheet

Python For Data Science Cheat Sheet

Scikit-Learn

Learn Python for Data Science (continued) at [www.DataCamp.com](#)

Scikit-learn

Scikit-learn is an open source Python library that implements a range of machine learning, preprocessing, cross-validation and visualization algorithms using a unified interface.

A Basic Example

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.cross_validation import train_test_split
from sklearn.metrics import accuracy_score
from sklearn import datasets
X, y = datasets.load_digits()
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
model = KNeighborsClassifier(n_neighbors=5)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy_score(y_test, y_pred)
```

Loading The Data

Your data needs to be numeric and stored as NumPy arrays or SciPy sparse matrices. Other types that are convertible to numeric arrays, such as Pandas DataFrames, are also acceptable.

```
import numpy as np
X = np.loadtxt('data.csv', dtype=float, delimiter=',')
y = np.loadtxt('target.csv', dtype=int, delimiter=',')
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
```

Training And Test Data

```
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
```

Preprocessing The Data

Standardization

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Normalization

```
from sklearn.preprocessing import Normalizer
normalizer = Normalizer()
X_train = normalizer.fit_transform(X_train)
X_test = normalizer.transform(X_test)
```

Encoding Categorical Features

```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
X_train = label_encoder.fit_transform(X_train)
X_test = label_encoder.transform(X_test)
```

Imputing Missing Values

```
from sklearn.preprocessing import Imputer
imputer = Imputer()
X_train = imputer.fit_transform(X_train)
X_test = imputer.transform(X_test)
```

Converting Raw Numerical Features

```
from sklearn.preprocessing import PolynomialFeatures
poly = PolynomialFeatures(2)
X_train = poly.fit_transform(X_train)
X_test = poly.transform(X_test)
```

Create Your Model

Supervised Learning Estimators

```
from sklearn.linear_model import LinearRegression
lr = LinearRegression(normalize=True)
from sklearn.svm import SVC
svm = SVC(kernel='linear')
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=5)
```

Unsupervised Learning Estimators

```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=3, random_state=0)
```

Model Fitting

Supervised Learning

```
from sklearn import svm
svm = svm.SVC(kernel='linear')
svm.fit(X_train, y_train)
```

Unsupervised Learning

```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
pca.fit(X_train)
```

Prediction

Supervised Estimators

```
y_pred = svm.predict(X_test)
y_pred = lr.predict(X_test)
y_pred = knn.predict(X_test)
```

Unsupervised Estimators

```
y_pred = pca.transform(X_test)
y_pred = kmeans.predict(X_test)
```

Evaluate Your Model's Performance

Classification Metrics

Accuracy Score

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)
```

Classification Report

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

Confusion Matrix

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

Regression Metrics

Mean Absolute Error

```
from sklearn.metrics import mean_absolute_error
mean_absolute_error(y_test, y_pred)
```

Mean Squared Error

```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test, y_pred)
```

R-Score

```
from sklearn.metrics import r2_score
r2_score(y_test, y_pred)
```

Clustering Metrics

Adjusted Rand Index

```
from sklearn.metrics import adjusted_rand_score
adjusted_rand_score(y_test, y_pred)
```

Homogeneity

```
from sklearn.metrics import homogeneity_score
homogeneity_score(y_test, y_pred)
```

Adjusted Rand Index

```
from sklearn.metrics import adjusted_rand_score
adjusted_rand_score(y_test, y_pred)
```

Cross-Validation

Grid Search

```
from sklearn.grid_search import GridSearchCV
grid = GridSearchCV(svm, param_grid, cv=5)
grid.fit(X_train, y_train)
```

Randomized Parameter Optimization

```
from sklearn.grid_search import RandomizedSearchCV
grid = RandomizedSearchCV(svm, param_grid, cv=5)
grid.fit(X_train, y_train)
```

파이썬을 활용한 기초 통계분석

2. Pandas Recap

```
[1]: import pandas as pd
```

Series

```
[2]: a = ?([1,3,5,7])
a
```

```
0    1
1    3
2    5
3    7
dtype: int64
```

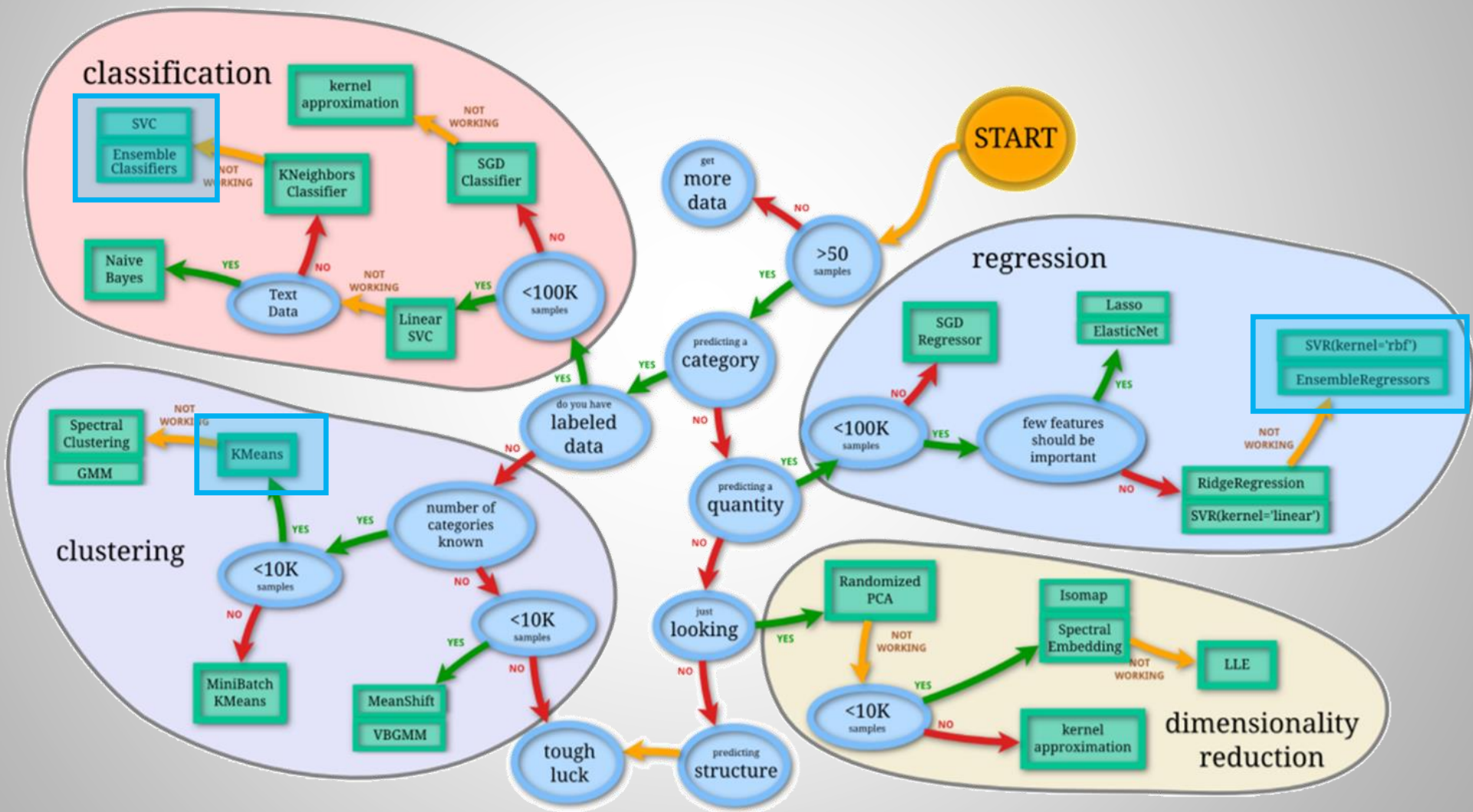
[3]: a.values

```
array([1, 3, 5, 7], dtype=int64)
```

본 교안은 멀티캠퍼스 교육을 위해 제작되었으며, 교육 외 배포/게시/공개를 금합니다.

풀어내려는 문제의 종류와 데이터의 타입(형태, 수)에 따른 ML 알고리즘 선택 가이드

http://scikit-learn.org/stable/tutorial/machine_learning_map/ (각 알고리즘 별 예시 코드 有)



수업 관련 공지사항

5팀

5/4팀

4팀

1팀

1/2팀

2팀

3팀

1팀 : 김은아, 문석호, 신광현, 장재원, 한상국 + 1人

2팀 : 김현하, 배소현, 윤영준, 이경희, 이재승, 하수진

3팀 : 김충희, 송윤성, 이가은, 이예랑, 이철희, 정수현

4팀 : 김채윤, 문희원, 이동규, 이주환, 이중기, 장준규

5팀 : 김지승, 이소은, 장희은, 한성준, 이대광 + 1人

* 세미프로젝트 시작 후 합류한 분들 중 파이썬이 익숙하지 않은 분들은 기존 Part 1~4의 내용들을 차례대로 실습해보고 팀 내에서 데이터에 모델을 적용하는 과정을 넓게 이해하는 것에 포커스를 맞춰주세요.

수업 관련 공지사항

* Part 1/2/4 에서 배운 지식들을 최대한 빠짐없이 활용하는데 초점을 맞춰주세요.

* 팀별로 자유로이 3층 내에서 자리를 옮겨서 논의하셔도 무방합니다.

* 월 ~ 화 : 팀별 분석 작업 -> 화요일 16:30 : 팀별 발표 및 질의응답 (15분 내외/팀)
: 화요일 16:25 까지 Jupyter notebook 제출 : repositivator@gmail.com

* 데이터 전처리 방법 / Model 선택 / Metric 선택 모두 자유입니다. (배운 내용의 복습에 Focus!)

* 발표 시 포함할 사항 : 데이터 전처리 방법 & 이유 / 모델 적용 프로세스 / 모델 적용 결과
발표 시 제출할 사항 : 주석이 포함된 전체 코드 (.ipynb 제출, PPT 발표자료 필수 X)

* 발표 시작 시간은 변동될 수 있습니다

혁신성장 청년인재 집중양성
이미지 분석 인공지능 서비스 개발 실무 과정

End of Document