Titanic

Prediction for Survived

EDA / Feature Engineering / Machine Learning

Titanic_Kaggle

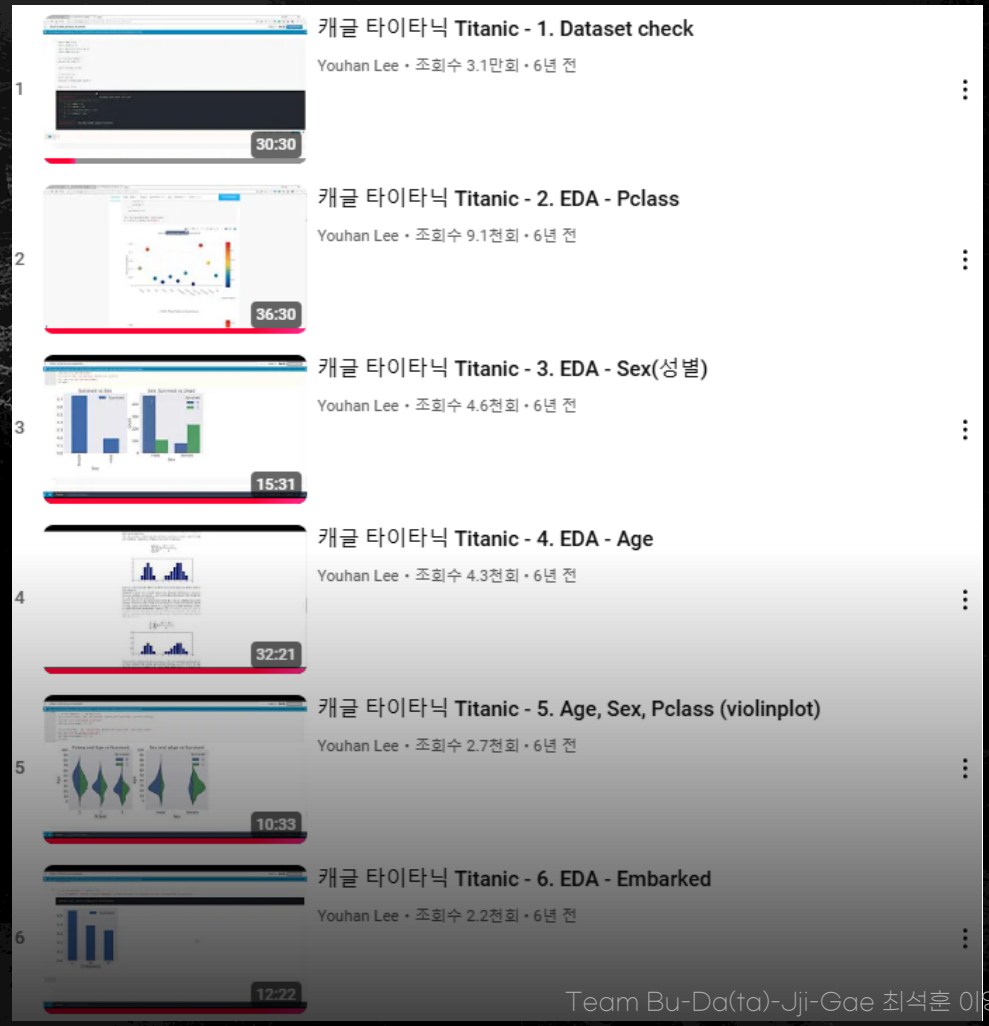
Learning

Titanic with Youhan Lee

게시자: Youhan Lee • 재생목록 • 동영상 15개 • 조회수 24,852회

1 - 14강 강의 수강

- 강의 수강 기록지 작성
- 강의 에러 기록지 작성

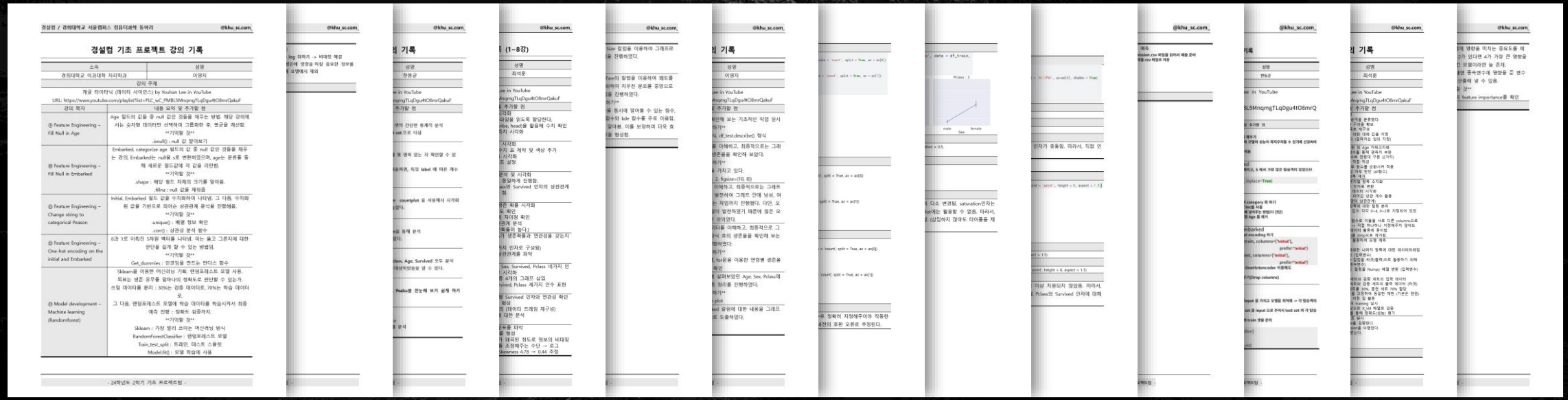


Titanic_Kaggle

Learning

1 - 14강 강의 수강

- 강의 수강 기록지 작성
- 강의 에러 기록지 작성



경성컴 기초 프로젝트 강의 기록

소속	성명
경희대학교 의과대학 지리학과	이영지

강의 주제

캐글 타이타닉 (데이터 사이언스) by Youhan Lee in YouTube

URL: https://www.youtube.com/playlist?list=PLC_wC_PMBL5MnqmgTLqDgu4tO8mrQakuF

강의 목차	내용 요약 및 추가할 점
⑤ Feature Engineering – Fill Null in Age	Age 필드의 값들 중 null 같은 것들을 채우는 방법. 해당 강의에서는 숫자형 데이터만 선택하여 그룹화한 후, 분균을 계산함. **기억할 것** .isnull() : null 값 알아보기
⑥ Feature Engineering – Fill Null in Embarked	Embarked, categorize age 필드의 값 중 null 같은 것들을 채우는 강의. Embarked는 null을 s로 변환하였으며, age는 분류를 통해 새로운 필드값에 각 값을 리턴함. **기억할 것** .shape : 해당 필드 자체의 크기를 알아봄. .fillna : null 값을 채워줌
⑦ Feature Engineering – Change string to categorical Peason	Initial, Embarked 필드 값을 수치화하여 나타냄. 그 다음, 수치화된 값을 기반으로 피어슨 상관관계 분석을 진행해줌. **기억할 것** .unique() : 배열 정보 확인 .corr() : 상관성 분석 함수
⑧ Feature Engineering – One-hot encoding on the initial and Embarked	0과 1로 이뤄진 5차원 벡터를 나타냄. 이는 좀고 그론지에 대한 판단을 쉽게 할 수 있는 방법임. **기억할 것** Get_dummies : 인코딩을 만드는 판다스 함수
⑨ Model development – Machine learning (Randomforest)	Sklearn을 이용한 머신러닝 기록. 랜덤포레스트 모델 사용. 목표는 생존 유무를 알아나의 정확도로 판단할 수 있는가. 쓰일 데이터를 분리 : 30%는 검증 데이터로, 70%는 학습 데이터로. 그 다음, 랜덤포레스트 모델에 학습 데이터를 학습시켜서 최종 예측 진행 ; 정확도 검증까지. **기억할 것** Sklearn : 가장 많이 쓰이는 머신러닝 방식 RandomForestClassifier : 랜덤포레스트 모델 Train_test_split : 트레인, 테스트 스플릿 Model.fit() : 모델 학습에 사용

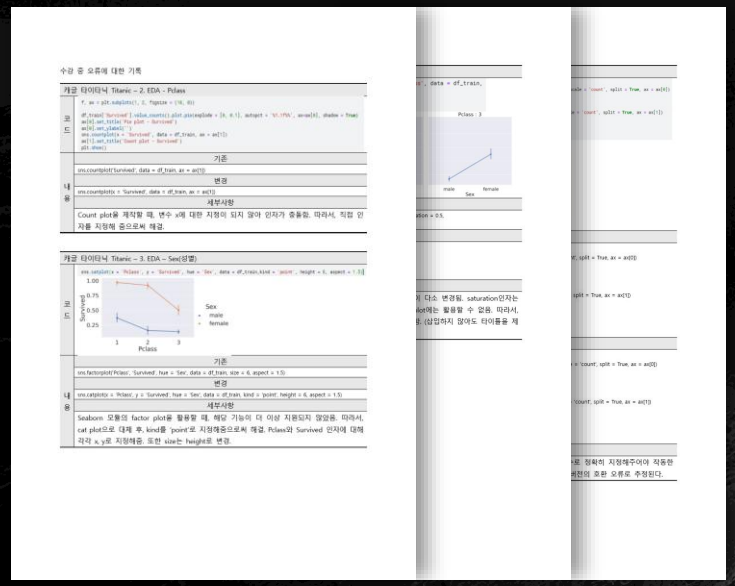
- 24학년도 2학기 기초 프로젝트팀 -

Titanic_Kaggle

Learning

1 - 14강 강의 수강

- 강의 수강 기록지 작성
- 강의 에러 기록지 작성



경설컴 / 경희대학교 서울캠퍼스 컴퓨터과학 동아리 @khu_sc.com_

경설컴 기초 프로젝트 강의 기록 (1~8강)

소속	성명
경희대학교 경영대학 경영학과	최석훈
강의 주제	
캐글 타이타닉 (데이터 사이언스) by Youhan Lee in YouTube URL: https://www.youtube.com/playlist?list=PLC_wC_PMBL5MnqmgTLqDgu4tO8mrQakuF	
강의 목차	내용 요약 및 추가할 점
① Dataset Check	데이터셋 구성 확인 및 결측치 시각화 1. df_train에 pandas로 csv파일을 읽도록 할당한다. 2. Pandas 자체 기능 (describe, head)을 활용해 수치 확인 3. Missingno를 활용한 결측치 시각화
② EDA – Pclass	Pclass와 Survived 데이터 활용해 시각화 1. Pandas 활용해 데이터 수치 표 제작 및 색상 추가 2. 파이 차트, 바 차트 활용 시각화 3. 차트에 대한 색상 및 구조 설정
③ EDA – Sex	Pclass와 Sex(성별) 간 상관관계 분석 및 시각화 1. 이전 강의와 대체적으로 동일하게 진행함. 2. Point plot을 활용해 Pclass와 Survived 인자의 상관관계 분석을 진행하고 차트화 함.
④ EDA – Age	Age(나이) 기준, Pclass 분포 및 생존 확률 시각화 1. KDE plot을 활용한 분포도 확인 2. 히스토그램과 KDE plot의 차이점 확인 3. Age와 Survival과의 상관관계 분석 (▶ 연령이 낮을수록 생존확률이 높다.)
⑤ Age, Sex, Pclass (Violinplot)	Violin plot을 활용해 Age와 Sex가 생존확률과 연관성을 갖는지 Pclass로 나누어 분석 1. Violin plot의 시각화 (3가지 인자로 구성됨) 2. 각각의 인자와 생존의 상관관계를 파악
⑥ EDA – Embarked	Count plot을 활용해, Embarked, Sex, Survived, Pclass 네가지 인수를 2x2 구조 (figsize = (2, 2))로 시각화 1. 하나의 결과 값에 2*2, 총 4개의 그래프 삽입 2. Embarked 기준, Sex, Survived, Pclass 세가지 인수 표현
⑦ EDA – Family Size	Count plot을 활용해 Family size별 Survived 인자와 연관성 확인 1. Count plot을 1*3구조로 형성 2. Family size에 대한 재정의 (데이터 프레임 재구성) 3. Family size별 생존확률에 대한 분석
⑧ EDA – Fare, Cabin, Ticket	Distplot을 활용해 Fare(요금)의 분포를 파악 1. Fare를 기준으로 분포도를 형성 2. Skewness(왜도): 데이터가 왜곡된 정도로 정보의 비대칭성을 의미. 과도한 왜도를 조정해주는 수단 → 로그 3. Lambda로 Log를 취해 Skewness 4.78 → 0.44 조정

- 24학년도 2학기 기초 프로젝트팀 -

경설컴 / 경희대학교 서울캠퍼스 컴퓨터과학 동아리 @khu_sc.com_

경설컴 기초 프로젝트 강의 기록

소속	성명
경희대학교 경영대학 경영학과	최석훈
강의 주제	
캐글 타이타닉 (데이터 사이언스) by Youhan Lee in YouTube URL: https://www.youtube.com/playlist?list=PLC_wC_PMBL5MnqmgTLqDgu4tO8mrQakuF	
강의 목차	내용 요약 및 추가할 점
⑨ Feature Engineering – Fill Null in Age	Feature engineering – Age의 결측치 보완 1. Name에서 Initial을 추출하여 승객을 분류했다. 2. Crosstab 시각화를 통해 Initial 구성을 확보 3. replace함수를 통해 5가지 분류로 재구성 4. loc(로케이션) 함수로 Null값에 대한 대체 값을 지정 5. Age에 대한 결측치를 해결한다. (결측치는 임의 지정)
⑩ Feature Engineering – Fill Null in Embarked	Feature engineering – Embarked 결측 보완 및 Age 카테고리화 1. Embarked 항목에 대해 fillna함수를 통해 결측치 보완 2. Age 항목에 대해 0~7까지 지수로 연령대 구분 (2가지) A. loc(로케이션) 활용하여 직접 작성 B. apply함수 활용하여 내부 함수를 순환시켜 적용 3. Age 카테고리 적용법의 동일성 여부 판단 (all함수) 4. drop함수를 통해 필요 없는 항목 제거
⑪ Feature Engineering – Change string to categorical Peason	Feature engineering – Initial, Embarked 문자열 항목 수치화 1. Initial과 Embarked의 문자열을 숫자로 변환 2. seaborn스타일 히트맵을 통한 데이터 시각화 3. Peason Correlation Coefficient 피어슨 상관 계수 활용 A. (-1: 음의 상관관계, 1: 양의 상관관계)
⑫ Feature Engineering – One-hot encoding on the initial and Embarked	Feature engineering – Initial, Embarked 항목에 대한 컬럼 분리 1. 기존: Initial, Embarked 항목의 값이 각각 0~4, 0~2로 지정되어 있었음. 2. 변환: pandas의 get_dummies 함수로 이들을 서로 다른 columns으로 분리함 (예: initial0, initial1, ...) → 직접 하나하나 지정해주지 않아도 자동으로 작업할 수 있어 큰 데이터 활용에 용이함. 3. 불필요한 항목(Passengerid 등)을 drop으로 제거함.
⑬ Model development – Machine learning (Randomforest)	Sklearn의 ensemble 중 RandomForest를 활용하여 모델 예측 1. 데이터 전처리 A. X_train에 Survived를 제외한 나머지 항목에 대한 데이터프레임을 Numpy 배열로 변환 (입력변수) B. target_label에 Survived 컬럼을 타겟(출력)으로 활용하기 위해 Numpy 배열로 변환 (종속변수) C. X_test에 test.csv의 모든 컬럼을 Numpy 배열 변환 (입력변수) 2. 훈련 세트와 검증 세트 분리 A. X_tr, X_vld: 각각 훈련 세트와 검증 세트의 입력 데이터 B. Y_tr, y_vld: 각각 훈련 세트와 검증 세트의 출력 데이터 (타겟) C. Test_size = 0.3: 검증 세트를 30%, 훈련 세트 70% 할당 D. Random_state: 시드 값을 고정하여 동일한 재현 (기본은 랜덤) 3. Random Forest Classifier 모델 지정 및 활용 A. fit(): 훈련 세트 입력하여 training 실시 4. predict(): 훈련된 model을 바탕으로 X_vld 배열로 검증 5. metrics의 accuracy_score함수를 통해 정확도(정확) 평가
⑭ Machine learning prediction – feature importance and prediction on test set	최종적으로 test.csv파일을 기반으로 테스트 실시. 1. Test.csv파일을 대상으로 model을 검증한다. 2. 검증한 모델로 주어진 submission을 수행한다. 3. 수행한 결과를 캐글에서 확인받는다.

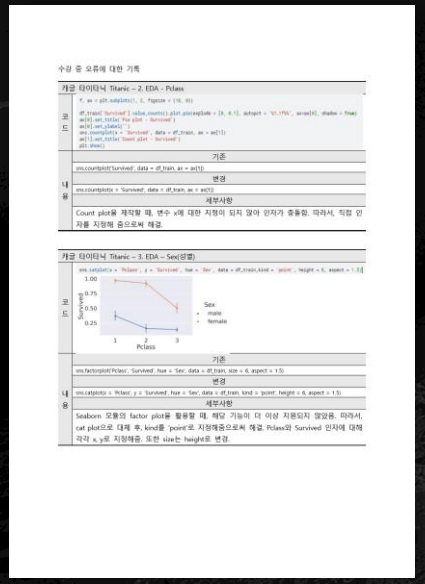
- 24학년도 2학기 기초 프로젝트팀 -

Titanic_Kaggle

Learning

1 - 14강 강의 수강

- 강의 수강 기록지 작성
- 강의 에러 기록지 작성


수강 중 오류에 대한 기록

캐글 타이타닉 Titanic – 2. EDA - Pclass

코드

내용

기존

변경

세부사항

Count plot을 제작할 때, 변수 x에 대한 지정이 되지 않아 인자가 충돌함. 따라서, 직접 인자를 지정해 줌으로써 해결.

캐글 타이타닉 Titanic – 3. EDA – Sex(성별)

코드

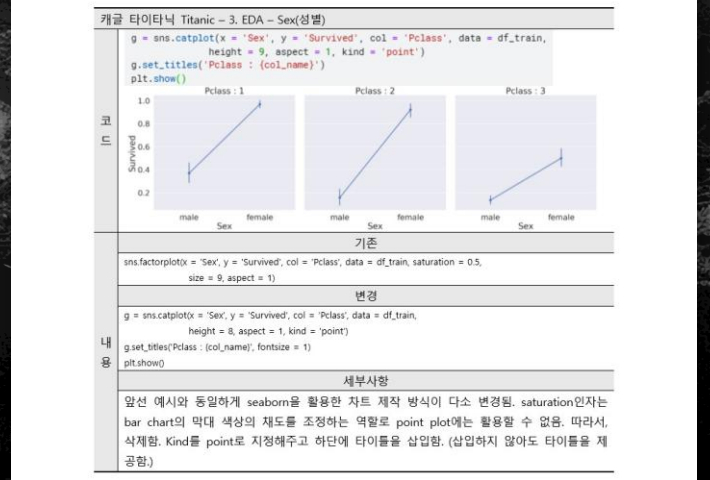
내용

기존

변경

세부사항

Seaborn 모듈의 factor plot을 활용할 때, 해당 기능이 더 이상 지원되지 않았음. 따라서, cat plot으로 대체 후, kind를 'point'로 지정해줌으로써 해결. Pclass와 Survived 인자에 대해 각각 x, y로 지정해줌. 또한 size는 height로 변경.


캐글 타이타닉 Titanic – 3. EDA – Sex(성별)

코드

```
g = sns.catplot(x = 'Sex', y = 'Survived', col = 'Pclass', data = df_train,
                height = 9, aspect = 1, kind = 'point')
g.set_titles('Pclass : {col_name}')
plt.show()
```

내용

기존

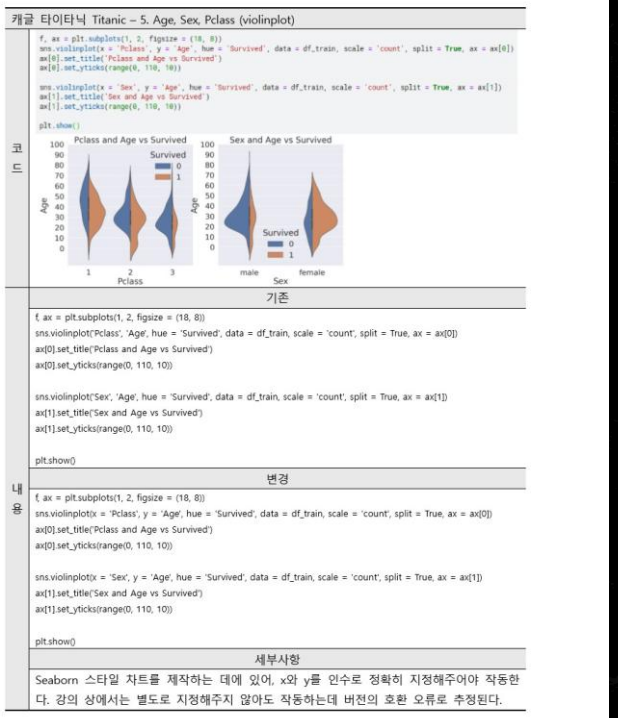
```
sns.factorplot(x = 'Sex', y = 'Survived', col = 'Pclass', data = df_train, saturation = 0.5,
               size = 9, aspect = 1)
```

변경

```
g = sns.catplot(x = 'Sex', y = 'Survived', col = 'Pclass', data = df_train,
                height = 8, aspect = 1, kind = 'point')
g.set_titles('Pclass : {col_name}', fontsize = 1)
plt.show()
```

세부사항

앞선 예시와 동일하게 seaborn을 활용한 차트 제작 방식이 다소 변경됨. saturation인자는 bar chart의 막대 색상의 채도를 조정하는 역할로 point plot에는 활용할 수 없음. 따라서, 삭제함. Kind를 point로 지정해주고 하단에 타이틀을 삽입함. (삽입하지 않아도 타이틀을 제공함.)


캐글 타이타닉 Titanic – 5. Age, Sex, Pclass (violinplot)

코드

```
f, ax = plt.subplots(1, 2, figsize = (18, 8))
sns.violinplot(x = 'Pclass', y = 'Age', hue = 'Survived', data = df_train, scale = 'count', split = True, ax = ax[0])
ax[0].set_title('Pclass and Age vs Survived')
ax[0].set_yticks(range(0, 110, 10))

sns.violinplot(x = 'Sex', y = 'Age', hue = 'Survived', data = df_train, scale = 'count', split = True, ax = ax[1])
ax[1].set_title('Sex and Age vs Survived')
ax[1].set_yticks(range(0, 110, 10))

plt.show()
```

내용

기존

```
f, ax = plt.subplots(1, 2, figsize = (18, 8))
sns.violinplot('Pclass', 'Age', hue = 'Survived', data = df_train, scale = 'count', split = True, ax = ax[0])
ax[0].set_title('Pclass and Age vs Survived')
ax[0].set_yticks(range(0, 110, 10))

sns.violinplot('Sex', 'Age', hue = 'Survived', data = df_train, scale = 'count', split = True, ax = ax[1])
ax[1].set_title('Sex and Age vs Survived')
ax[1].set_yticks(range(0, 110, 10))

plt.show()
```

변경

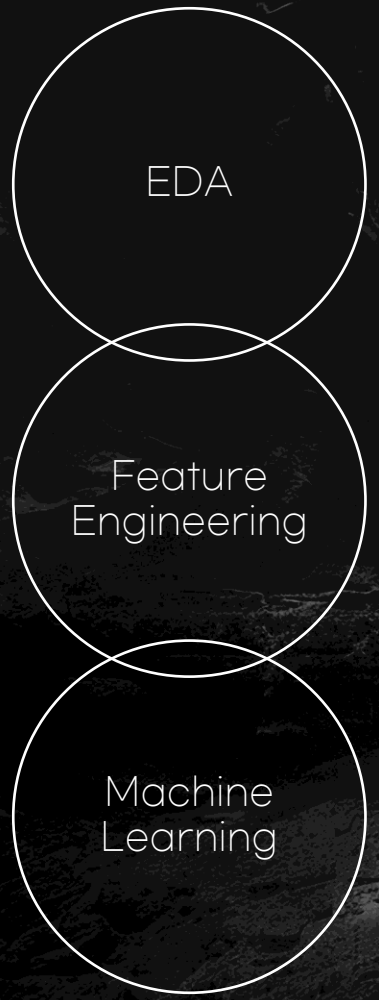
```
f, ax = plt.subplots(1, 2, figsize = (18, 8))
sns.violinplot(x = 'Pclass', y = 'Age', hue = 'Survived', data = df_train, scale = 'count', split = True, ax = ax[0])
ax[0].set_title('Pclass and Age vs Survived')
ax[0].set_yticks(range(0, 110, 10))

sns.violinplot(x = 'Sex', y = 'Age', hue = 'Survived', data = df_train, scale = 'count', split = True, ax = ax[1])
ax[1].set_title('Sex and Age vs Survived')
ax[1].set_yticks(range(0, 110, 10))

plt.show()
```

세부사항

Seaborn 스타일 차트를 제작하는 데에 있어, x와 y를 인수로 정확히 지정해주어야 작동한다. 강의 상에서는 별도로 지정해주지 않아도 작동하는데 버전의 호환 오류로 추정된다.

Titanic_Kaggle

Process for Machine Learning



EDA
탐색적 데이터 분석으로, 가공하지 않은 일차적 Raw Data를 통해,
주어진 데이터의 구성과 형태를 확인하고 ML까지의 과정을 계획한다.

Feature Engineering
주어진 데이터를 ML에 적용하기 위해 적절한 형태로
가공, 변환, 통합, 축소 등 데이터 전처리를 수행한다.

Machine Learning with Random Forest
Feature Engineering을 수행한 정제된 데이터를 통해, 선택한 ML모델에
적용한다. 결과를 수치로 도출하고 정확도를 측정해 모델의 완성도를 확인한다.
결과적으로, Test파일을 통해 Target을 예측한다.

Titanic_Kaggle

Structure of Dataset

*** Data Dictionary** Columns: 10, Row: 891 (train.csv)

Target: Survived

Variable	Definition	Key
Survival	Survival	0 = No, 1 = Yes
Pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
Sex	Sex	
Age	Age in years	
SibSp	# of siblings / spouses aboard the Titanic	
Parch	# of parents / children aboard the Titanic	
Ticket	Ticket number	
Fare	Passenger fare	
Cabin	Cabin number	
Embarked	Port of embarked	C = Cherbourg, Q = Queenstown, S = Southampton

Titanic_Kaggle

Before Beginning

```
1. import numpy as np
2. import pandas as pd
3. import matplotlib.pyplot as plt
4. import seaborn as sns
5. plt.style.use('seaborn')
6. sns.set(font_scale = 2.5)
7.
8. import missingno as msno
9. import warnings
10.warnings.filterwarnings('ignore')
11.
12.%matplotlib inline
```

Numpy & Pandas

데이터 분석을 위한 도구로서 Pearson Correlation Coefficient 등에 활용.

Matplotlib & Seaborn

분석한 데이터 셋을 바탕으로 시각화에 도움을 준다. Heatmap 등에 활용.

Missingno

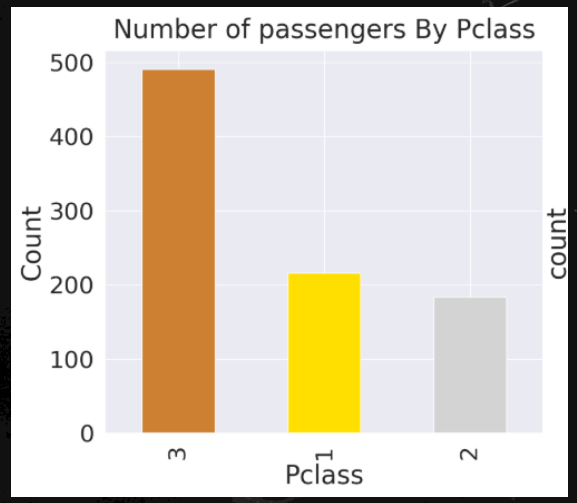
데이터 분석을 위한 도구로서 Pearson Correlation Coefficient 등에 활용.

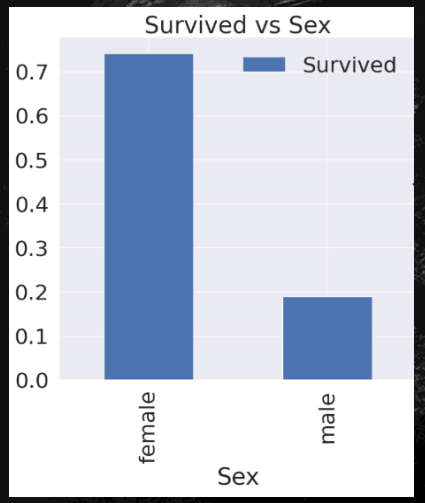
Warnings

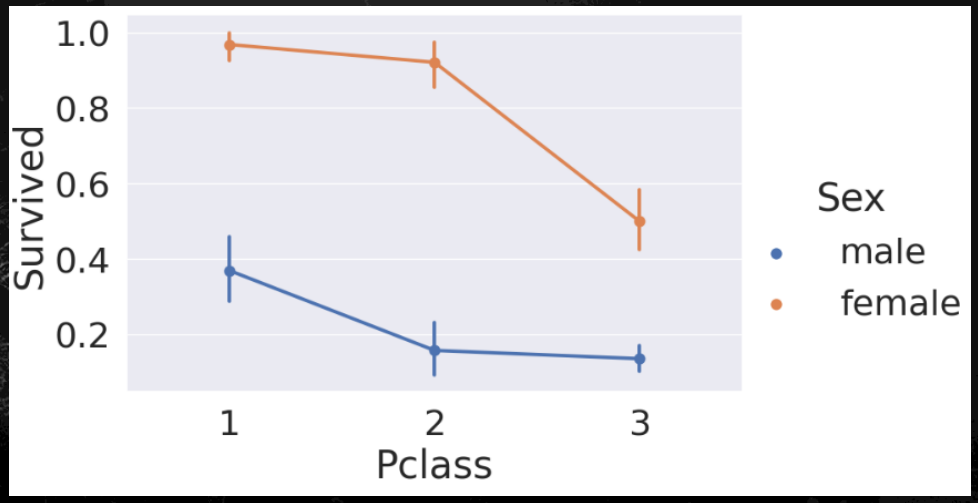
코드 작성 시 발생하는 경고에 대해 무시하는 명령.

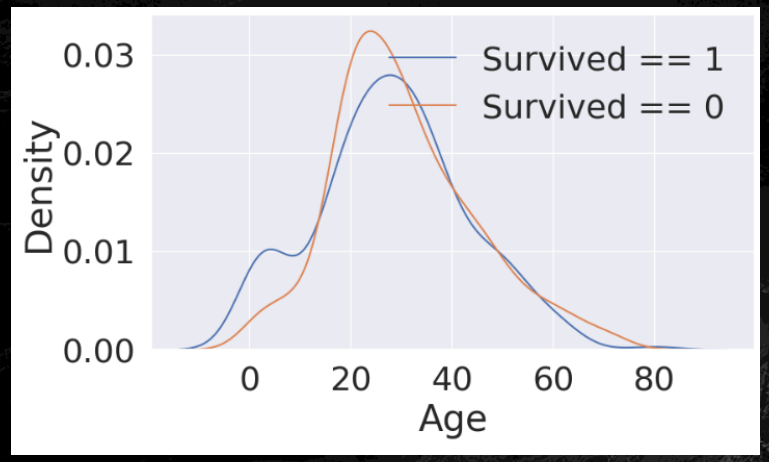
Titanic_Kaggle

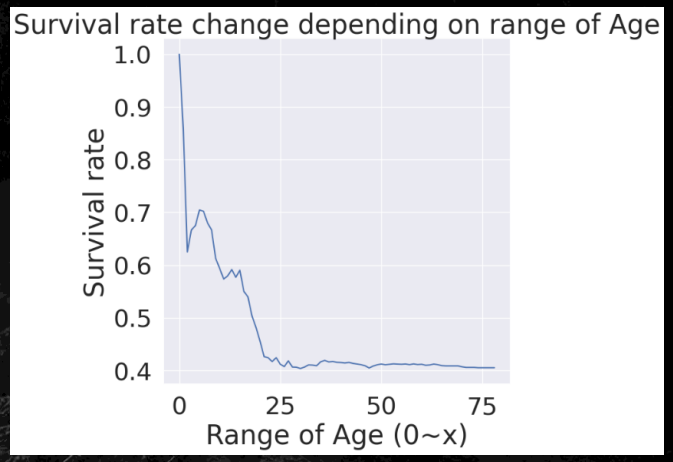
EDA

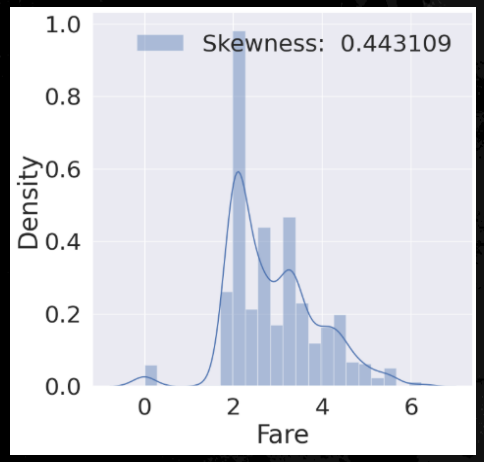












Survived	0	1	All
Pclass			
1	80	136	216
2	97	87	184
3	372	119	491
All	549	342	891

Titanic_Kaggle

Preprocess

```
#train, test파일 불러오기
df_train = pd.read_csv('/kaggle/input/titanic/train.csv')
df_test = pd.read_csv('/kaggle/input/titanic/test.csv')

#데이터 전처리
#Familysize 통합
df_train['Familysize'] = df_train['SibSp'] + df_train['Parch'] + 1
df_test['Familysize'] = df_test['SibSp'] + df_test['Parch'] + 1

#test.csv 결측치 제거_Fare
#Pclass별 평균값으로 대체
fare_by_pclass = df_test.groupby('Pclass')['Fare'].transform('mean')
df_test['Fare'].fillna(fare_by_pclass, inplace=True)
#로그 변환
df_train['Fare'] = df_train['Fare'].map(lambda i: np.log(i) if i > 0 else 0)
df_test['Fare'] = df_test['Fare'].map(lambda i: np.log(i) if i > 0 else 0)

#Initial 분류
df_train['Initial'] = df_train.Name.str.extract('([A-Za-z]+)\.')
df_test['Initial'] = df_test.Name.str.extract('([A-Za-z]+)\.')

#Initial 수합 (5가지)
df_train['Initial'].replace(['Mlle','Mme','Ms','Dr','Major','Lady','Countess','Jonkheer','Col','R
ev','Capt','Sir','Don', 'Dona'],
                            ['Miss','Miss','Miss','Mr','Mr','Mrs','Mrs','Other','Other','Other','M
r','Mr','Mr', 'Mr'],inplace=True)

df_test['Initial'].replace(['Mlle','Mme','Ms','Dr','Major','Lady','Countess','Jonkheer','Col','Re
v','Capt','Sir','Don', 'Dona'],
                            ['Miss','Miss','Miss','Mr','Mr','Mrs','Mrs','Other','Other','Other','M
r','Mr','Mr', 'Mr'],inplace=True)

#Age의 Null값 대체
df_train.loc[(df_train.Age.isnull())&(df_train.Initial=='Mr'),'Age'] = 33
df_train.loc[(df_train.Age.isnull())&(df_train.Initial=='Mrs'),'Age'] = 36
df_train.loc[(df_train.Age.isnull())&(df_train.Initial=='Master'),'Age'] = 5
df_train.loc[(df_train.Age.isnull())&(df_train.Initial=='Miss'),'Age'] = 22
df_train.loc[(df_train.Age.isnull())&(df_train.Initial=='Other'),'Age'] = 46

df_test.loc[(df_test.Age.isnull())&(df_test.Initial=='Mr'),'Age'] = 33
df_test.loc[(df_test.Age.isnull())&(df_test.Initial=='Mrs'),'Age'] = 36
df_test.loc[(df_test.Age.isnull())&(df_test.Initial=='Master'),'Age'] = 5
df_test.loc[(df_test.Age.isnull())&(df_test.Initial=='Miss'),'Age'] = 22
df_test.loc[(df_test.Age.isnull())&(df_test.Initial=='Other'),'Age'] = 46

#Embarked의 Null값을 'S'로 대체
df_train['Embarked'].fillna('S', inplace = True)
```

```
#Embarked의 Null값을 'S'로 대체
df_train['Embarked'].fillna('S', inplace = True)

#Age_cat에 Age를 카테고리화 하여 입력 (10살 단위로 구분)
def category_age(x):
    for i in range(7):
        if x < ((i+1)*10):
            return i
    return 7

df_train['Age_cat'] = df_train['Age'].apply(category_age)
df_test['Age_cat'] = df_test['Age'].apply(category_age)
df_train.drop(['Age'], axis = 1, inplace = True) #기존 Age 레이블 삭제
df_test.drop(['Age'], axis = 1, inplace = True)

#Initial을 int값으로 대체 (이후, 머신러닝을 통한 결과 예측을 위해)
#또한, Initial을 위에서 Age의 Null값 처소에 사용하므로 나중에 정리한다.
df_train['Initial'] = df_train['Initial'].map({'Master': 0, 'Miss': 1, 'Mr': 2, 'Mrs': 3, 'Othe
r': 4})
df_test['Initial'] = df_test['Initial'].map({'Master': 0, 'Miss': 1, 'Mr': 2, 'Mrs': 3, 'Other':
4})

#Embarked도 동일하게 int값으로 변환
df_train['Embarked'] = df_train['Embarked'].map({'C': 0, 'Q': 1, 'S': 2})
df_test['Embarked'] = df_test['Embarked'].map({'C': 0, 'Q': 1, 'S': 2})

#Sex도 동일하게 int값으로 변환
df_train['Sex'] = df_train['Sex'].map({'female': 0, 'male': 1})
df_test['Sex'] = df_test['Sex'].map({'female': 0, 'male': 1})

#get_dummies함수를 활용한 Initial 데이블 분리 (Initial0, Initial1, ...)
df_train = pd.get_dummies(df_train, columns=['Initial'], prefix='Initial')
df_test = pd.get_dummies(df_test, columns=['Initial'], prefix='Initial')

#get_dummies함수를 활용한 Embarked 데이블 분리 (Embarked0, Embarked1, Embarked2)
df_train = pd.get_dummies(df_train, columns=['Embarked'], prefix='Embarked')
df_test = pd.get_dummies(df_test, columns=['Embarked'], prefix='Embarked')

#불필요한 columns 제거
df_train.drop(['PassengerId', 'Name', 'SibSp', 'Parch', 'Ticket', 'Cabin'], axis=1, inplace=True)
df_test.drop(['PassengerId', 'Name',  'SibSp', 'Parch', 'Ticket', 'Cabin'], axis=1, inplace=True)
```

Null data 정리

카테고리화(범주)

레이블 병합

etc.

Titanic_Kaggle

Machine Learning

```
#sklean 모듈 설치
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
from sklearn.model_selection import train_test_split

#X_train: Survived를 제외한 입력변수 설정
#target_label: Survived를 출력변수(타겟) 설정
#X_test: 테스트 파일을 Numpy배열로 변환
X_train = df_train.drop('Survived', axis=1).values
target_label = df_train['Survived'].values
X_test = df_test.values

#훈련 및 검증 데이터를 테스트 세트(70%)와 검증 세트(30%)로 분리
#시드값을 고정하여 동일한 결과
X_tr, X_vld, y_tr, y_vld = train_test_split(X_train, target_label, test_size=0.3, random_st
ate=2018)

#머신러닝 모델 설정 (RandomForestClassifier)
#fit함수로 데이터 입력
#prediction변수에 예측값 입력 (변수는 Test의 입력데이터(X_vld))
model = RandomForestClassifier()
model.fit(X_tr, y_tr)
prediction = model.predict(X_vld)

#Pandas를 활용한 Feature Importance 추출
from pandas import Series

feature_importance = model.feature_importances_
Series_feat_imp = Series(feature_importance, index=df_test.columns)
```

1. sklearn

2. Submission

```
#최종 제출을 위한 Submission.csv 불러오기
submission = pd.read_csv('/kaggle/input/titanic/gender_submission.csv')

#제작한 모델을 바탕으로 Submission의 Survived 예측값을 출력
prediction = model.predict(X_test)
submission['Survived'] = prediction

#최종 제출
submission.to_csv('./my_first_submission.csv', index=False)
```

3. Accuracy

```
print('총 {}명 중 {:.2f}% 정확도로 생존을 맞춤'.format(y_vld.shape[0], 100 * metrics.accuracy_s
core(prediction, y_vld)))
```

4. Feature Importance

```
plt.figure(figsize=(8, 8))
Series_feat_imp.sort_values(ascending=True).plot.barh()
plt.xlabel('Feature importance')
plt.ylabel('Feature')
plt.show()
```