Aichemist Train Session

CHAP 04 분류(1)

시작에 앞서

- **깃허브에 올라온 Notebook file 다운 받으시면, 코드 전체를 볼 수 있어요 :)**
- **조원들과 질문들에 함께 답하며, 공부한 내용들을 복습하고, 데이터 분석의 큰 흐름을 익혀봅시다 !**

+) 데이터 분석의 큰 과정과 데이터를 분석하는 힘을 길러드리고 싶은 마음(이라고 쓰고 욕심이라고 읽습니다.)에

어려운 용어나, 처음보는 개념들이 튀어나올 수도 있어요..ㅎㅎ 어렵다면 질문 대ㅐㅐㅐ환영 ^0^

ABOUT, Heart Attack Analysis & Prediction Dataset

age- 환자의 나이

sex- 환자의 성별

cp- 흉통 유형 ~ 0 = 전형적인 협심증, 1 = 비정형 협심증, 2 = 비협심증 통증, 3 = 무증상

trtbps- 안정시 혈압(mmHg 단위)

chol- BMI 센서를 통해 가져온 콜레스테롤(mg/dl)

fbs- (공복 혈당 > 120 mg/dl) ~ 1 = 참, 0 = 거짓

restecg- 안정시 심전도 결과 ~ 0 = 정상, 1 = ST-T파 정상, 2 = 좌심실 비대

thalachh - 최대 심박수 달성

oldpeak- 이전 피크

slp- 슬로프

caa- Number of major vessels

thall- 탈륨 스트레스 테스트 결과 ~ (0,3)

exng- 운동으로 인한 협심증 ~ 1=예, 0=아니요

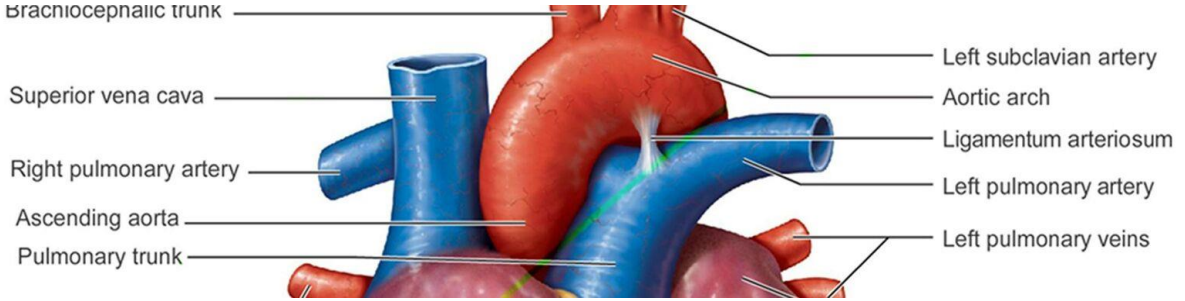
output- 1 = more chance / 0 = less chance of heart attack

캐글 데이터 / 노트북 링크

https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset/data

https://www.kaggle.com/code/namanmanchanda/heart-attack-eda-prediction-90-accuracy

심장마비 발병 여부 예측하기



Understanding Data

2.3.1 The shape of the data

In [3]:
```
print("The shape of the dataset is : ", df.shape)
```

```
The shape of the dataset is :  (303, 14)
```

2.3.2 Preview of the first 5 rows of the data

In [4]:
```
df.head()
```

Out[4]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Output=1 : 심장마비 발병확률 높음

Output=0 : 심장마비 발병확률 낮음

In [5]:
```
dict = {}
for i in list(df.columns):
    dict[i] = df[i].value_counts().shape[0]

pd.DataFrame(dict,index=["unique count"]).transpose()
```

Out[5]:

	unique count
age	41
sex	2
cp	4
trtbps	49
chol	152
fbs	2
restecg	3
thalachh	91
exng	2
oldpeak	40
slp	3
caa	5
thall	4
output	2

Q1. unique count 칼럼이 의미하는 바가 무엇일까요?

1. unique count : 중복 제거한 칼럼 값

Understanding Data

2.3.4 Separating the columns in categorical and continuous

```
cat_cols = ['sex','exng','caa','cp','fbs','restecg','slp','thall']
con_cols = ["age","trtbps","chol","thalachh","oldpeak"]
target_col = ["output"]
print("The categorial cols are : ", cat_cols)
print("The continuous cols are : ", con_cols)
print("The target variable is :  ", target_col)
```

```
The categorial cols are :  ['sex', 'exng', 'caa', 'cp', 'fbs', 'restecg', 'slp', 'thall']
The continuous cols are :  ['age', 'trtbps', 'chol', 'thalachh', 'oldpeak']
The target variable is :   ['output']
```

Q2. 분리한 두 칼럼 categorial col/continuous col를 비교해봅시다.

둘은 어떤 칼럼들을 구분한 것일까요 ?

범주형 column, 연속형 column

Q2 + (노트북에 답이 나와있진 않지만, 추론해봅시다 !)

위의 두 칼럼으로 분리하는 방법은 어떤 것이 있을까요? 어떻게 저 둘을 나눌 수 있을지 고민해봅시다 ~

unique count를 구했으니까 10개 미만인 카테고리들을 범주형으로 구분

Understanding Data

2.3.5 Summary statistics

```
df[con_cols].describe().transpose()
```

	count	mean	std	min	25%	50%	75%	max
age	303.0	54.366337	9.082101	29.0	47.5	55.0	61.0	77.0
trtbps	303.0	131.623762	17.538143	94.0	120.0	130.0	140.0	200.0
chol	303.0	246.264026	51.830751	126.0	211.0	240.0	274.5	564.0
thalachh	303.0	149.646865	22.905161	71.0	133.5	153.0	166.0	202.0
oldpeak	303.0	1.039604	1.161075	0.0	0.0	0.8	1.6	6.2

Q3. describe()를 con_cols에만 적용한 이유는 ?

수학적 통계는 연속형 피처에만 적용할 수 있기 때문

2.3.6 Missing values

```
df.isnull().sum()
```

```
age         0
sex         0
cp          0
trtbps      0
chol        0
fbs         0
restecg     0
thalachh    0
exng        0
oldpeak     0
slp         0
caa         0
thall       0
output      0
dtype: int64
```

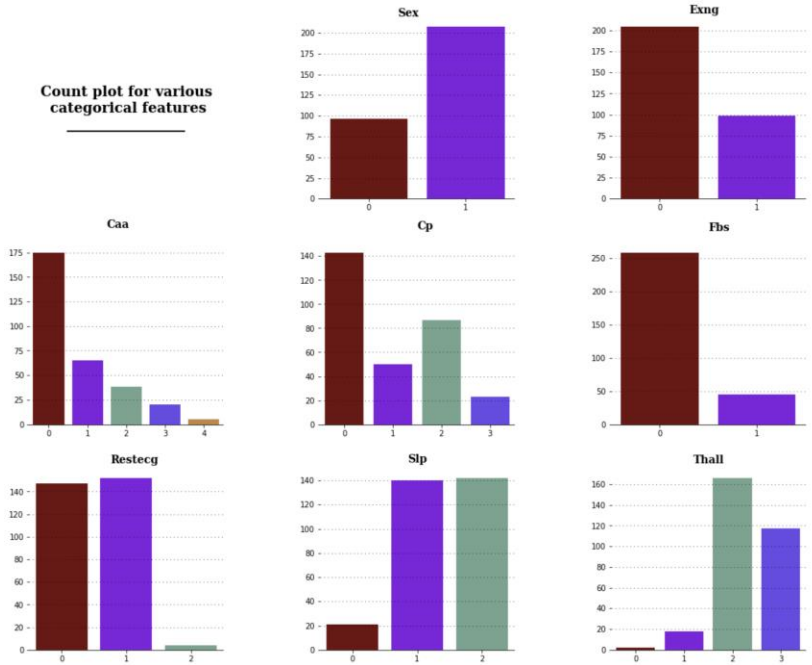
Q4. 위 코드의 목적은 무엇일까요?

null값 확인 용도. 결손값있는지 체크하기

Exploratory Data Analysis (EDA)

Q5. EDA란 무엇일까요 ? (*중요한 개념이니 검색해서 찾아봅시다)

탐색적 데이터 분석. EDA는 데이터 분석을 시작하기 전에 데이터를 다양한 각도에서 관찰하고 이해하는 과정입니다.
통계적 요약, 시각화, 데이터의 구조와 패턴 파악 등을 통해 데이터에 대한 직관을 얻고, 데이터의 특징과 내재된 문제
점들(예: 결측치, 이상치 등)을 발견하는 것이 목표입니다.



Restecg - 1(having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)이 가장 많고
2(showing probable or definite left ventricular hypertrophy by Estes' criteria)
Thall은 2가 가장 많고 0이 가장 적음. 탈륨 스트레스 테스트 결과

1/2 2/0

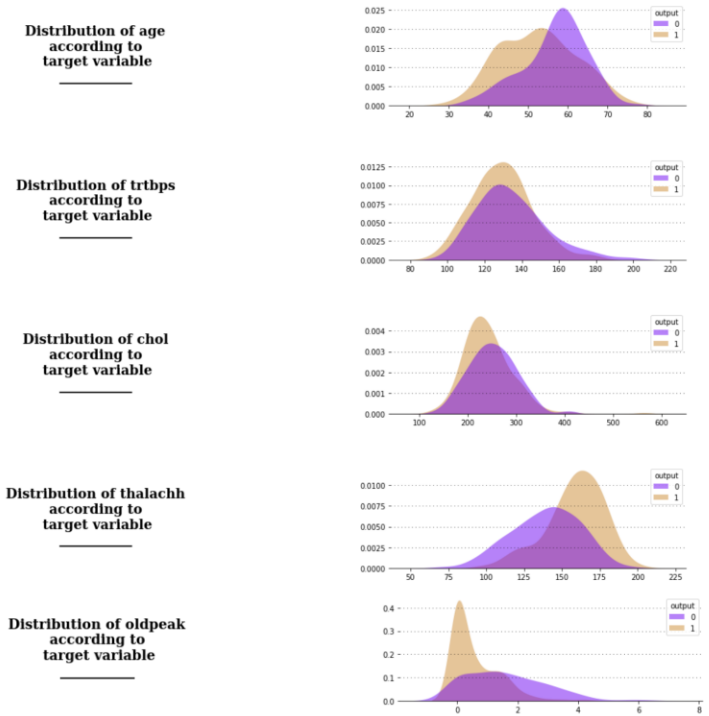
Q6. Restecg/Thall 카테고리형 변수의 count plot을 보고, 가장 빈도수가 높은 범주와 가장 낮은 범주를 찾아봅시다

Q6-1. Q6 답변을 기반으로, 두 변수가 target 변수에 어떤 영향을 미칠 수 있는지 설명해봅시다

데이터셋에서 대다수를 차지하는 범주에 모델이 과도하게 편향되어 학습될 수 있다. 모델이 다수 클래스 패턴에 지나치게 적응해 소수 클래스 패턴을 학습하는데 실패하거나 무시할 수 있음

타겟값 별로 연속형 피처 시각화

Q7. 아래 연속형 그래프 자료의 목적은 무엇일지 추측해봅시다



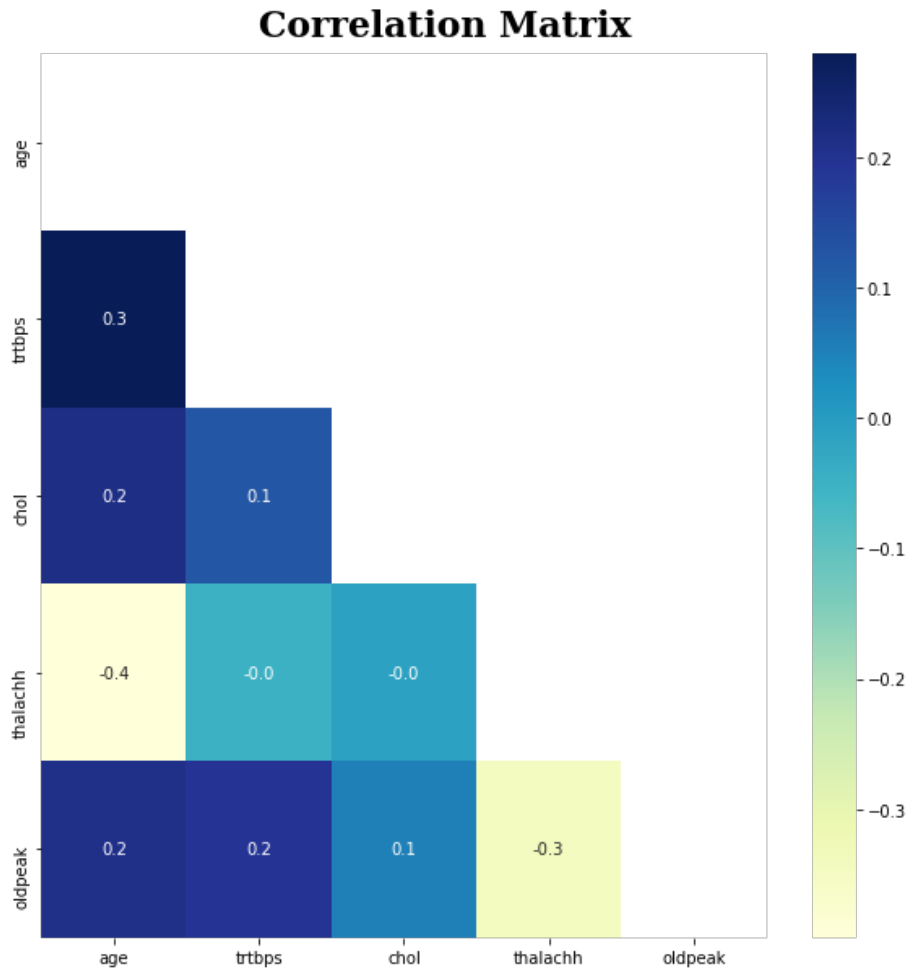
Bivariate Analysis

```
df_corr = df[con_cols].corr().transpose()
df_corr
```

	age	trtbps	chol	thalachh	oldpeak
age	1.000000	0.279351	0.213678	-0.398522	0.210013
trtbps	0.279351	1.000000	0.123174	-0.046698	0.193216
chol	0.213678	0.123174	1.000000	-0.009940	0.053952
thalachh	-0.398522	-0.046698	-0.009940	1.000000	-0.344187
oldpeak	0.210013	0.193216	0.053952	-0.344187	1.000000

Q8. 표와 그림은 상관계수에 대해 분석한 것입니다. 상관 계수가 무엇인지, 값이 클수록/작을수록 무엇을 의미하는 것인지 답해봅시다(*배운 적 없으니 검색하고 답해주세요!)

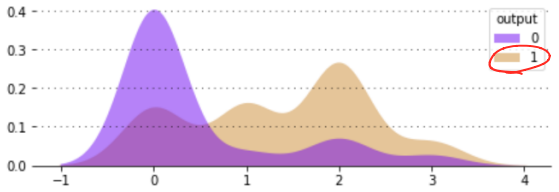
두 변수 간의 선형 관계의 강도와 방향을 측정하는 지표. 상관계수의 값은 -1부터 1까지의 범위를 가지며 절댓값이 클수록 강한 선형관계를 나타냄.



Exploratory Data Analysis (EDA)

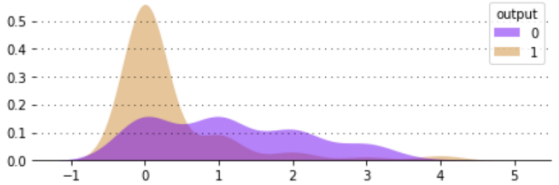
Chest pain distribution

0 - Typical Angina
1 - Atypical Angina
2 - Non-anginal Pain
3 - Asymptomatic



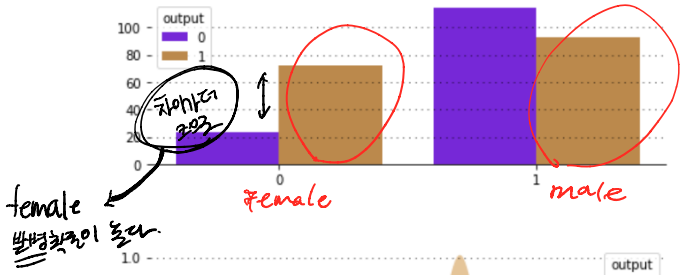
Number of major vessels

0 vessels
1 vessel
2 vessels
3 vessels
4vessels



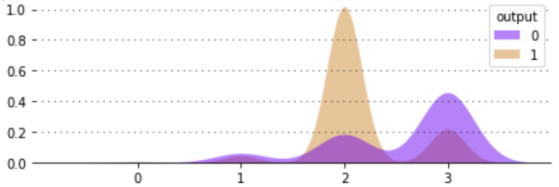
Heart Attack according to sex

0 - Female
1 - Male



Distribution of thall according to target variable

Thalium Stress Test Result
0, 1, 2, 3



Q9. 왼쪽 그래프 자료를 보고,

어떤 유형의 사람이 더 심장마비 발병 확률이 높은 지 추측해봅시다

non anginal Pain일수록 발병확률이 높다

0 vessels일 수록 발병확률이 높다.

여성일 수록 발병 확률이 높다.

Thalium stress가 2일 수록 높다 (근데 앞서 thall 2가 가장 많았기 때문에 발병률이 높아보이는 거일 수도 있다. 유의 부탁ㄱ)

Scaling and Encoding features

```
# creating a copy of df
df1 = df

# define the columns to be encoded and scaled
cat_cols = ['sex','exng','caa','cp','fbs','restecg','slp','thall']
con_cols = ["age","trtbps","chol","thalachh","oldpeak"]

# encoding the categorical columns
df1 = pd.get_dummies(df1, columns = cat_cols, drop_first = True)

# defining the features and target
X = df1.drop(['output'],axis=1)
y = df1[['output']]

# instantiating the scaler
scaler = RobustScaler()

# scaling the continuous featuree
X[con_cols] = scaler.fit_transform(X[con_cols])
print("The first 5 rows of X are")
X.head()
```

Q10. 어떤 전처리 기법들을 거쳤나요?

핫인코딩 사용 (pd.get_dummies), 표준 정규화 RobustScaler. StandardScaler보다 이상치의 영향에 강하다.

cat-> 원핫인코딩/ con->표준정규화(모든 변수들이 같은 스케일로 맞춰주기 위해서.)

Q10-2. 전처리 기법을 적용하는 대상은 어떤 칼럼 종류인지 답하고 그 이유는 무엇일지 답해주세요.

```
The first 5 rows of X are
```

	age	trtbps	chol	thalachh	oldpeak	sex_1	exng_1	caa_1	caa_2	caa_3	...	cp_2	cp_3	fbs_1	restecg_1	restecg_2	slp_1	slp_2	thall_1
0	0.592593	0.75	-0.110236	-0.092308	0.9375	1	0	0	0	0	...	0	1	1	0	0	0	0	1
1	-1.333333	0.00	0.157480	1.046154	1.6875	1	0	0	0	0	...	1	0	0	1	0	0	0	0
2	-1.037037	0.00	-0.566929	0.584615	0.3750	0	0	0	0	0	...	0	0	0	0	0	0	1	0
3	0.074074	-0.50	-0.062992	0.769231	0.0000	1	0	0	0	0	...	0	0	0	1	0	0	1	0
4	0.148148	-0.50	1.795276	0.307692	-0.1250	0	1	0	0	0	...	0	0	0	1	0	0	1	0

Modeling

```
# instantiating the object
dt = DecisionTreeClassifier(random_state = 42)

# fitting the model
dt.fit(X_train, y_train)

# calculating the predictions
y_pred = dt.predict(X_test)

# printing the test accuracy
print("The test accuracy score of Decision Tree is ", accuracy_score(y_test, y_pred))
```

```
The test accuracy score of Decision Tree is  0.7868852459016393
```

Q11. 결정 트리가 과적합에 취약한 이유는 무엇이었나요? 결정 트리의 기반이 어떤 값이었는지를 떠올리며 생각해봅시다.

트리가 너무 깊어질때까지 데이터를 완벽하게 분류하려고 하기 때문

Q11-2. 결정 트리에서 과적합을 방지하기 위해 조정할 수 있는 파라미터에는 어떤 것들이 있었나요?

max_depth, min_samples_split, min_samples_leaf, max_features, max_leaf_nodes

```
# instantiating the object
rf = RandomForestClassifier()

# fitting the model
rf.fit(X_train, y_train)

# calculating the predictions
y_pred = dt.predict(X_test)

# printing the test accuracy
print("The test accuracy score of Random Forest is ", accuracy_score(y_test, y_pred))
```

```
The test accuracy score of Random Forest is  0.7868852459016393
```

Q12. 랜덤 포레스트는 결정트리를 기반으로 한 알고리즘이었습니다.
랜덤 포레스트는 어떻게 학습 데이터로부터 결정 트리들을 생성했나요?

부트스트래핑으로 데이터 세트를 샘플링. 각 샘플링 된 개별 데이터 세트에 결정 트리 분류기를 각각 적용

Modeling

```
# instantiate the classifier
gbt = GradientBoostingClassifier(n_estimators = 300,max_depth=1,subsample=0.8,max_features=0.2,random_state=42)

# fitting the model
gbt.fit(X_train,y_train)

# predicting values
y_pred = gbt.predict(X_test)
print("The test accuracy score of Gradient Boosting Classifier is ", accuracy_score(y_test, y_pred))
```

```
The test accuracy score of Gradient Boosting Classifier is  0.8688524590163934
```

Q13 GBM의 각 하이퍼파라미터를 분석하고 각 값이 무엇을 의미하는 지 대답해보자 (n_estimators, max_depth, subsample, max_features)

그리고 이런 하이퍼 파라미터 튜닝을 도와주는 장치가 있었는데 무엇일까?

n_estimator : 약한 학습기 개수 지정 (약한 학습기 이때 결정 트리) max_depth : 결정트리 최대트리 깊이 subsample: 약한 학습기가 학습에 사용하는 데이터의 샘플링 비율.

max_features: 최대 피처 개수/ GridSearchCV -> 하이퍼 파라미터 튜닝

Q14. 여기서 결정트리와 관련된 파라미터가 나온 이유는 무엇일까요? GBM의 정의와 관련지어 생각해봅시다

약한 학습기가 결정트리(사용한 분류기가 결정트리라서)

```
# printing the test accuracy
print("The test accuracy score of Decision Tree is ", accuracy_score(y_test, y_pred))
```

```
The test accuracy score of Decision Tree is  0.7868852459016393
```

```
# printing the test accuracy
print("The test accuracy score of Random Forest is ", accuracy_score(y_test, y_pred))
```

```
The test accuracy score of Random Forest is  0.7868852459016393
```

Q15. 이 문제에 "분류" 모델들을 적용한 이유를 생각해봅시다

어떤 조건일 때 심장 발병 확률이 높은지 확인. 0/1 구분해내는게 우리 목표..!

```
# predicting values
y_pred = gbt.predict(X_test)
print("The test accuracy score of Gradient Boosting Classifier is ", accuracy_score(y_test, y_pred))
```

```
The test accuracy score of Gradient Boosting Classifier is  0.8688524590163934
```

수고하셨습니다