2021. 인공지능

Placements prediction

컴퓨터공학전공 2017106119 염상권

CONTENTS

01

개요

01 개요

01 소개

02. 알고자 하는 것

03. 데이터 셋

소개

공대생을 학교에서 면접을 실시하고 회사에서 일할 지원자를 고용하는 자리가 마련 되었습니다.

학교에서는 학생들의 취업률를 알고자 학생들의 데이터를 가지고 이들의 **취업률을 예측**하고자 합니다.

Engineering Placements Predictions

01 개요

01. 소개

02 알고자 하는 것

알고자 하는 것

What's this kernel is about?

학생의 "Stream", "HistoryOfBacklogs", "CGPA"를 기준으로 배치 여부를 분류하는 지도 예측을 수행할 것입니다.
또한, 우리는 데이터를 분석하고 통찰력을 찾기 위해 노력할 것입니다.

02

데이터 셋

02 데이터 셋

01 데이터 읽어오기

02. 구성

03. 통계

데이터 읽어오기

```
import pandas as pd
df=pd.read_csv("../input/engineering-placements-prediction/collegePlace.csv")
df.head(10).style.set_properties(**{"background-color": "black","color": "white", "border-color":
"black","font-size":"11.5pt",'width': 200})
```

	Age	Gender	Stream	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
0	22	Male	Electronics And Communication	1	8	1	1	1
1	21	Female	Computer Science	0	7	1	1	1
2	22	Female	Information Technology	1	6	0	0	1
3	21	Male	Information Technology	0	8	0	1	1
4	22	Male	Mechanical	0	8	1	0	1
5	22	Male	Electronics And Communication	0	6	0	0	0
6	21	Male	Computer Science	0	7	0	1	0
7	21	Male	Information Technology	1	7	0	0	0
8	21	Male	Computer Science	2	6	0	0	1
9	21	Female	Computer Science	1	6	1	0	0

02 데이터 셋

01. 데이터 읽어오기

02 구성

03. 통계

구성

Serial no.	Column_Name	Description
1.	Age	Age of the student.
2.	Gender	Gender of the student.
3.	Stream	Respective course of the students.
4.	Internships	No. of the internships,students worked.
5.	CGPA (Cumulative Grade Points Average)	Results of the students.
6.	Hostel	Whether the student stays in the hostel or not?
7.	HistoryOfBacklogs	Whether the student have backlogs before?
8.	PlacedOrNot	Whether a student gets placed or not?

컬럼

데이터 사이즈

데이터 타입

데이터 정보

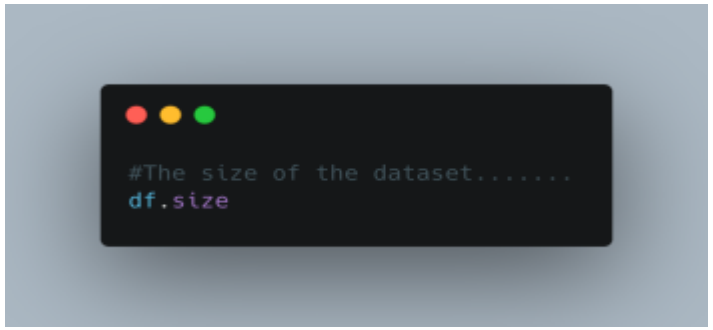
02 데이터 셋

01. 데이터 읽어오기

02 구성

03. 통계

구성



23728

컬럼

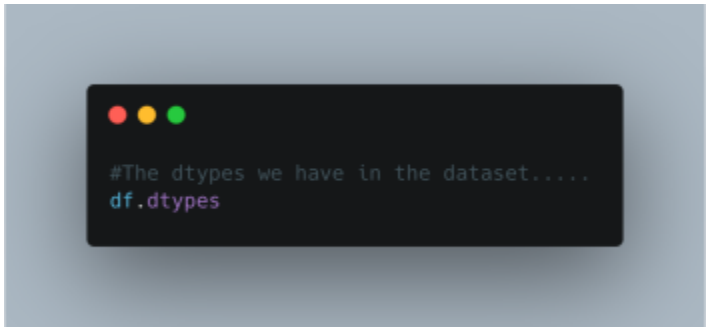
데이터 사이즈

데이터 타입

데이터 정보

02 데이터 셋

구성




```
Age                    int64
Gender                object
Stream                object
Internships            int64
CGPA                   int64
Hostel                 int64
HistoryOfBacklogs      int64
PlacedOrNot            int64
dtype: object
```

컬럼

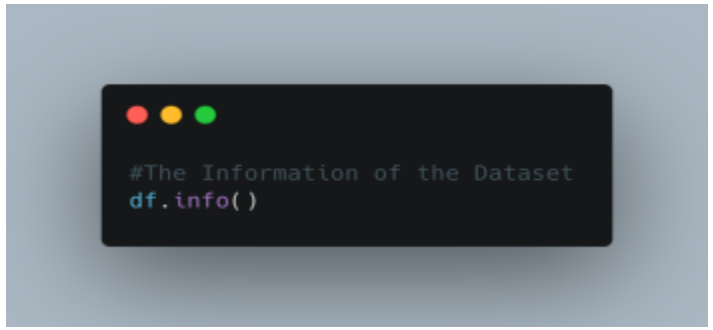
데이터 사이즈

데이터 타입

데이터 정보

02 데이터 셋

구성




```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2966 entries, 0 to 2965
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Age                2966 non-null   int64
 1   Gender             2966 non-null   object
 2   Stream             2966 non-null   object
 3   Internships        2966 non-null   int64
 4   CGPA               2966 non-null   int64
 5   Hostel             2966 non-null   int64
 6   HistoryOfBacklogs  2966 non-null   int64
 7   PlacedOrNot        2966 non-null   int64
dtypes: int64(6), object(2)
memory usage: 185.5+ KB
```

컬럼

데이터 사이즈

데이터 타입

데이터 정보

02 데이터 셋

01. 데이터 읽어오기

02. 구성

03 통계

통계

	Age	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
count	2966.000000	2966.000000	2966.000000	2966.000000	2966.000000	2966.000000
mean	21.490000	0.700000	7.070000	0.270000	0.190000	0.550000
std	1.320000	0.740000	0.970000	0.440000	0.390000	0.500000
min	19.000000	0.000000	5.000000	0.000000	0.000000	0.000000
25%	21.000000	0.000000	6.000000	0.000000	0.000000	0.000000
50%	21.000000	1.000000	7.000000	0.000000	0.000000	1.000000
75%	22.000000	1.000000	8.000000	1.000000	0.000000	1.000000
max	30.000000	3.000000	9.000000	1.000000	1.000000	1.000000

데이터에 대한 통계 입니다.

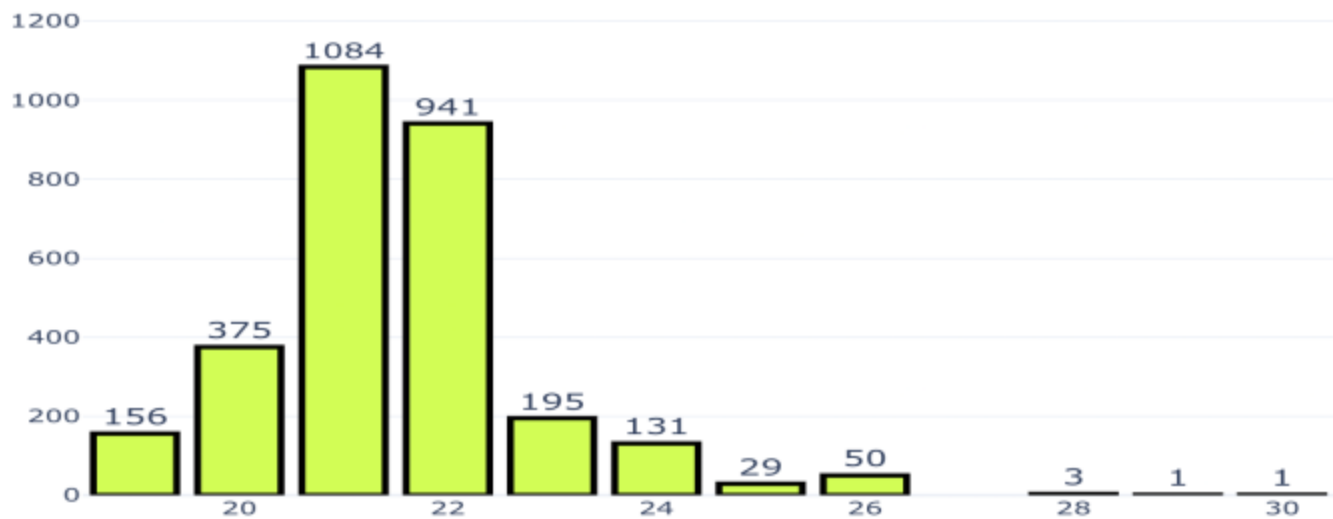
각 컬럼에 대한 평균 갯수, 평균, 표준, 최소 , 25%, 50%, 75%, 최대의 결과 값을 테이블로 나타내봤습니다.

03

분포도

03 분포도

Age



1. 지원자가 가장 많은 나이대는 21살 입니다.
2. 두 번째로 가장 많은 나이대는 22살 입니다.

주로, 공대생의 나이는 19 - 25살로 구성됨을 알 수 있으며 26살 이상의 학생은 몇 없는 것을 알 수 있습니다.

03 분포도

01. Age

02 Internships

03. CGPA

04. Hostel

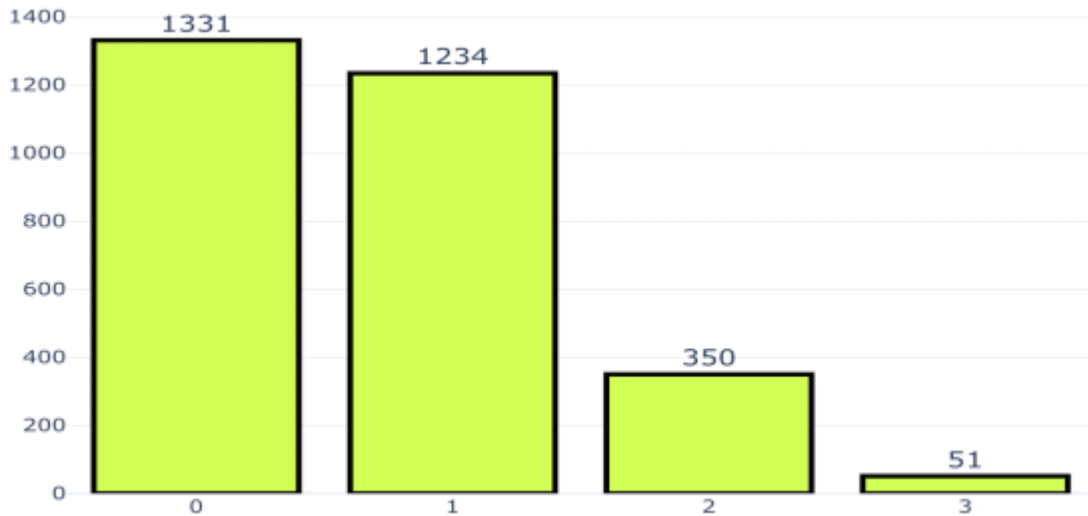
05. History of backlogs

06. Placed or Not

07. Gender

08. Stream

Internships



1. 대다수의 학생들은 어떠한 인턴쉽을 하지 않았습니다.
2. 그 다음으로 한 번의 인턴쉽을 한 학생이 주류였습니다.
3. 3번의 인턴쉽을 한 학생은 거의 없는 것을 알 수 있습니다.

인턴쉽을 더 많이 한 학생일 수록 좋은 업무 경험을 갖고 있을 것 입니다.

따라서 이는 취업 또는 취업의 기회의 확률이 더 높을 것으로 추정됩니다.

03 분포도

01. Age

02. Internships

03 CGPA

04. Hostel

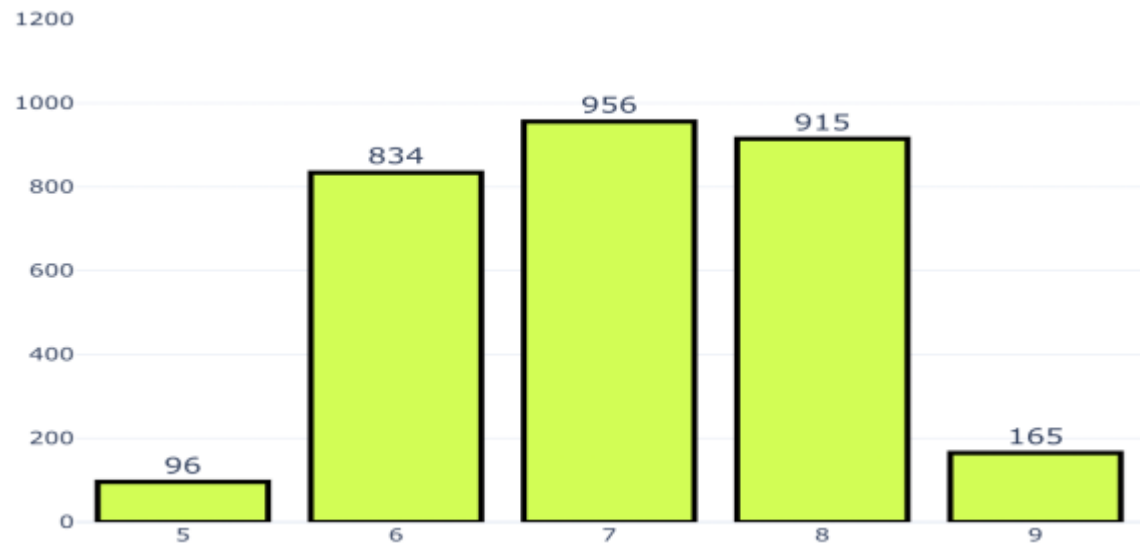
05. History of backlogs

06. Placed or Not

07. Gender

08. Stream

CGPA



1. 가장 많은 학생은 CGPA가 7점 인 것을 알 수 있습니다.
2. 두 번째로 많은 학생의 CGPA는 8점 인 것을 알 수 있습니다.

CGPA 는 채용에 중요한 역할을 합니다.

03 분포도

01. Age

02. Internships

03. CGPA

04 Hostel

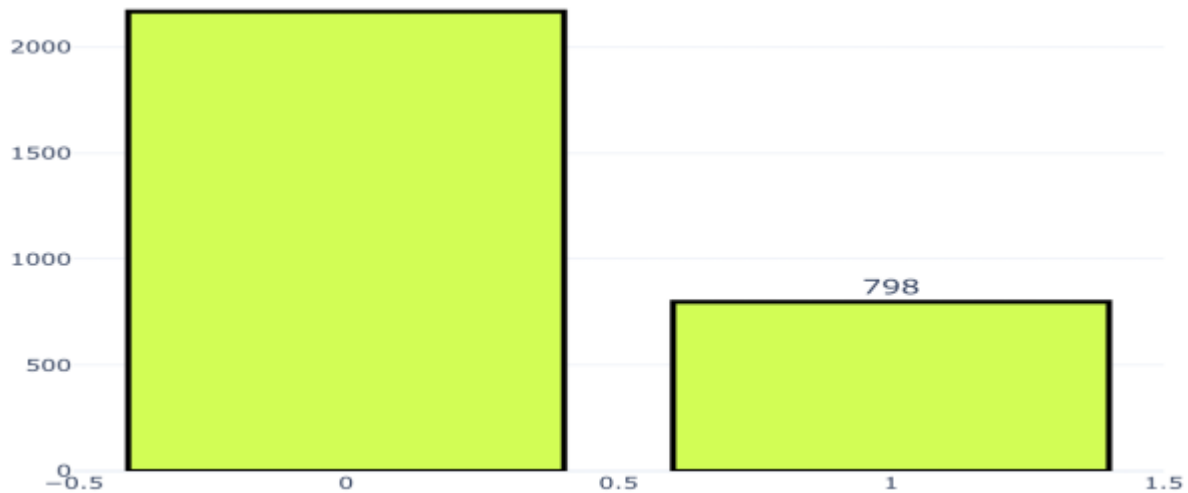
05. History of backlogs

06. Placed or Not

07. Gender

08. Stream

Hostel



1. 가장 많은 학생들이 기숙사에 지내지 않은 것을 알 수 있습니다.
2. 상당수의 나머지 학생들이 기숙사에 지내는 것을 알 수 있습니다.

03 분포도

01. Age

02. Internships

03. CGPA

04. Hostel

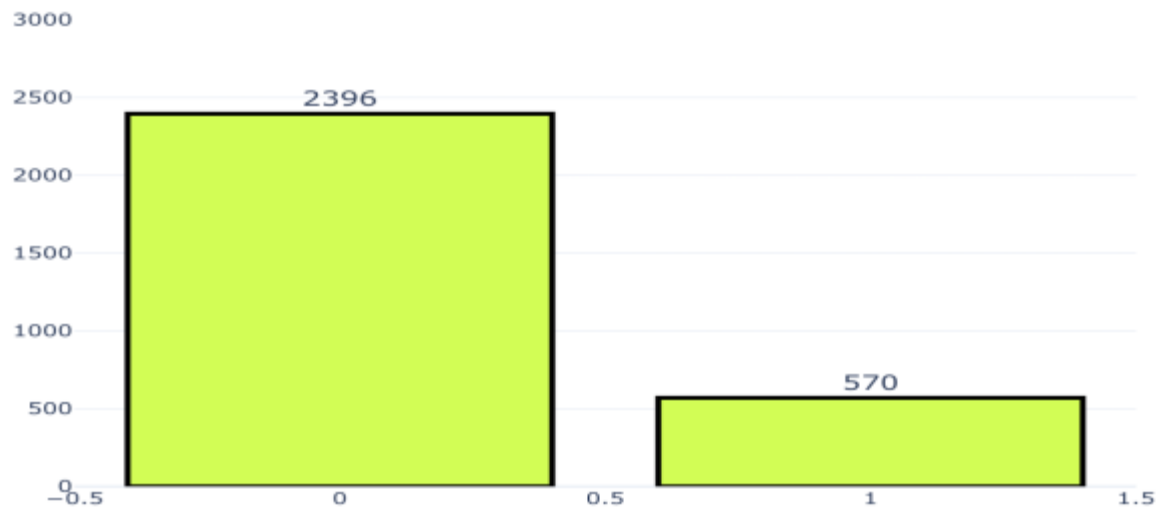
05 HistoryOfBacklogs

06. Placed or Not

07. Gender

08. Stream

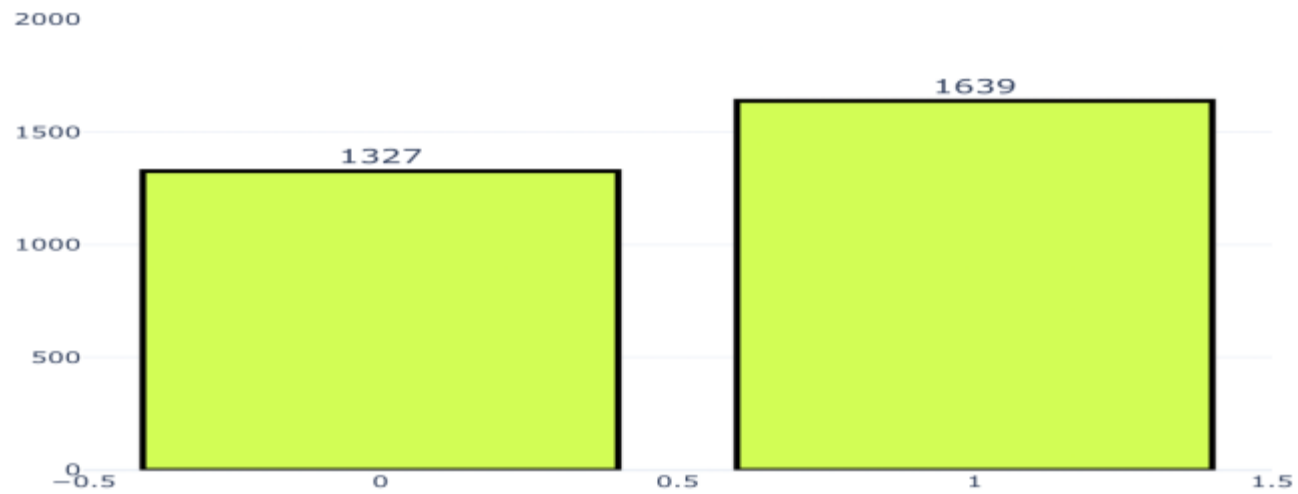
Hostel



1. 가장 많은 학생들이 back log를 가지고 있지 않습니다.
2. 적은 학생들이 back log를 가지고 있지 않습니다.

03 분포도

Placed or not



1. 많은 학생들이 직업을 가진 것을 알 수 있습니다.
2. 취직을 하지 못한 학생들은 비교적 적었습니다.

03 분포도

Gender

1. 가장 많은 학생은 남자입니다.
2. 여학생은 소수임을 알 수 있습니다.

03 분포도

01. Age

02. Internships

03. CGPA

04. Hostel

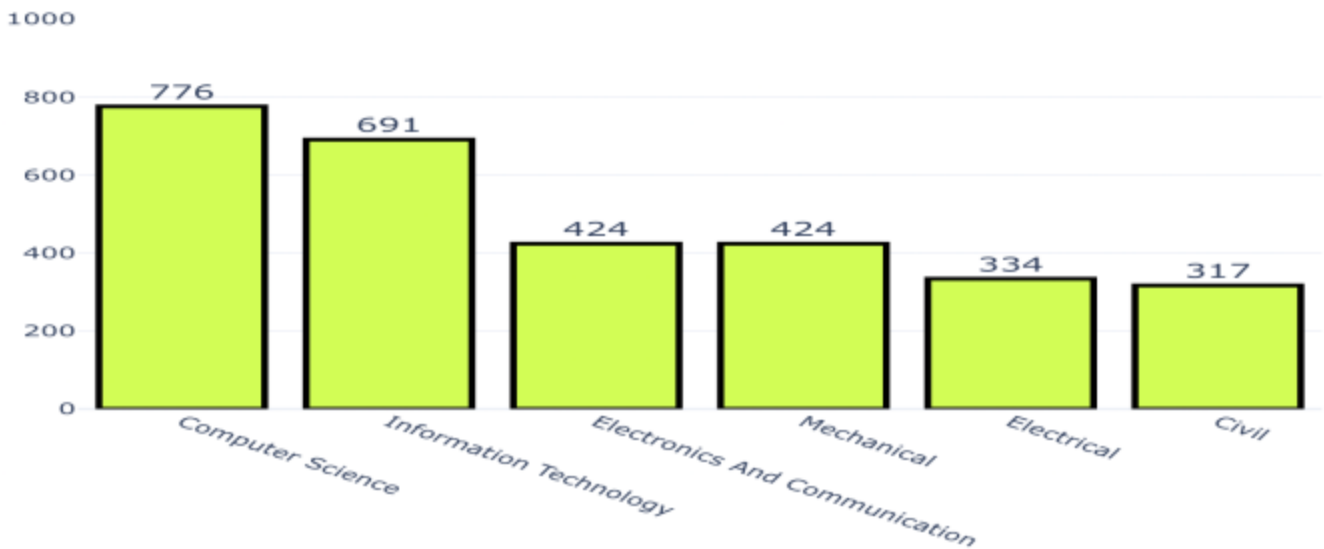
05. History of backlogs

06. Placed or not

07. Gender

08 Stream

Stream



1. 가장 많은 학생들이 컴퓨터 과학 전공을 하고 있습니다.
2. 두 번째로 많은 학생들이 정보 기술을 전공하고 있습니다.
3. 가장 적은 학생이 토목 공학을 전공하고 있습니다.

04

EDA 탐색

04 EDA 탐색

EDA 란?

EDA

수집한 데이터가 들어왔을 때, **이를 다양한 각도에서 관찰하고 이해하는 과정**입니다.

즉, 데이터를 분석하기 전에 그래프나 통계적인 방법으로 자료를 직관적으로 바라보는 과정입니다.

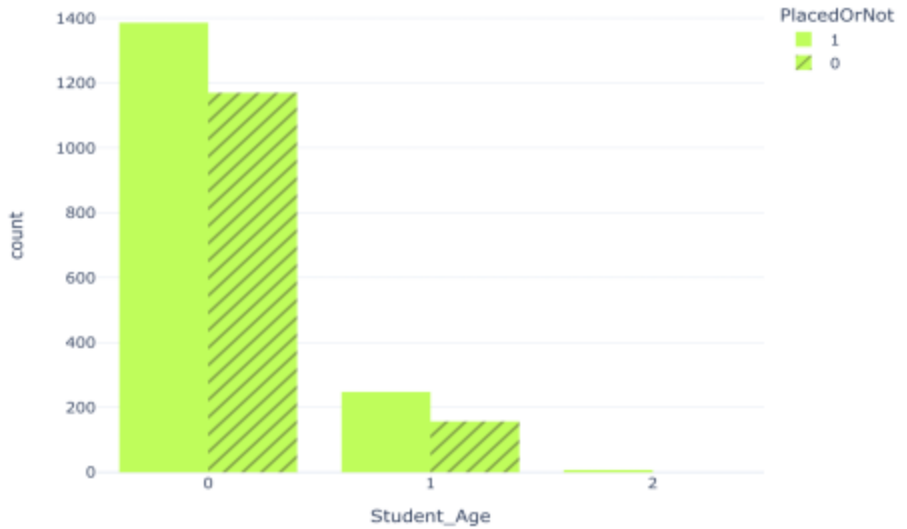
EDA는 데이터를 시각화 해, 중요한 특징을 찾아보는 과정이다!!

04 EDA 탐색

Age vs Placed or Not

나이는 연속적인 값입니다. 따라서 이를 정규화를 통해 범주형 값으로 변환할 필요가 있습니다.

따라서 binning을 통해 다양한 연령을 단일 bin으로 그룹화 하거나 단일 값으로 할당합니다.

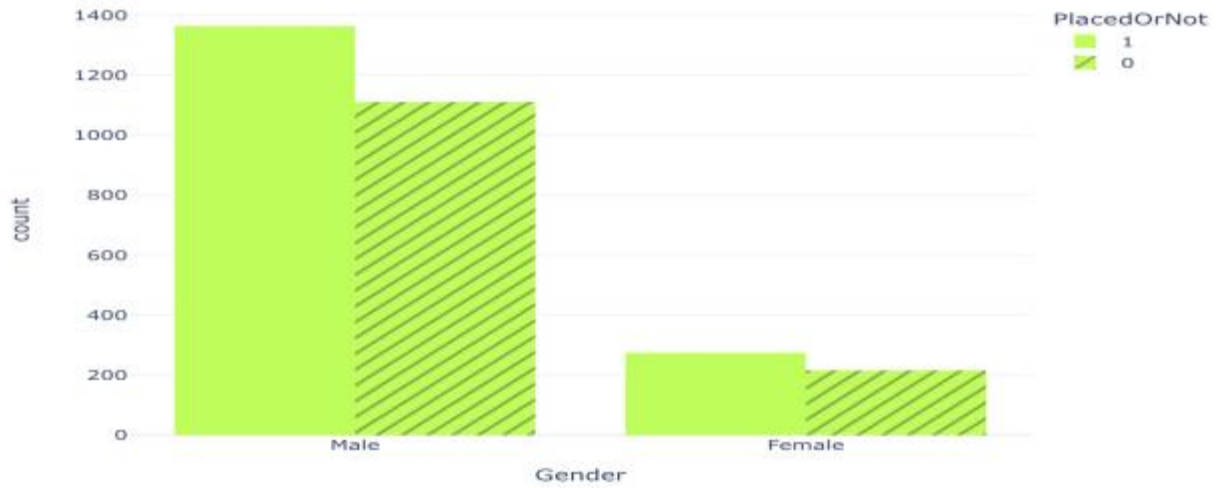


21 - 22살인 학생들은 캠퍼스 채용을 통해 많은 기회를 받은 반면에,

28 - 30사이에 있는 학생들은 채용 기회가 매우 적었습니다.

04 EDA 탐색

Gender vs Placed or Not

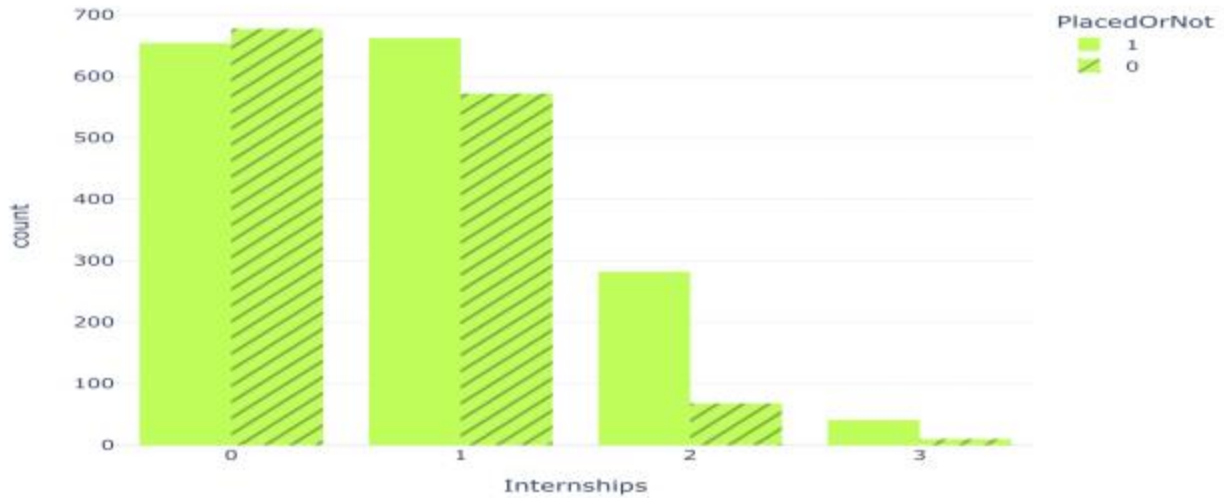


데이터 상으로는 남성이 여성의 취업률 보다

높은 것을 알 수 있습니다.

04 범주형 vs 연속형

Internships vs Placed or Not

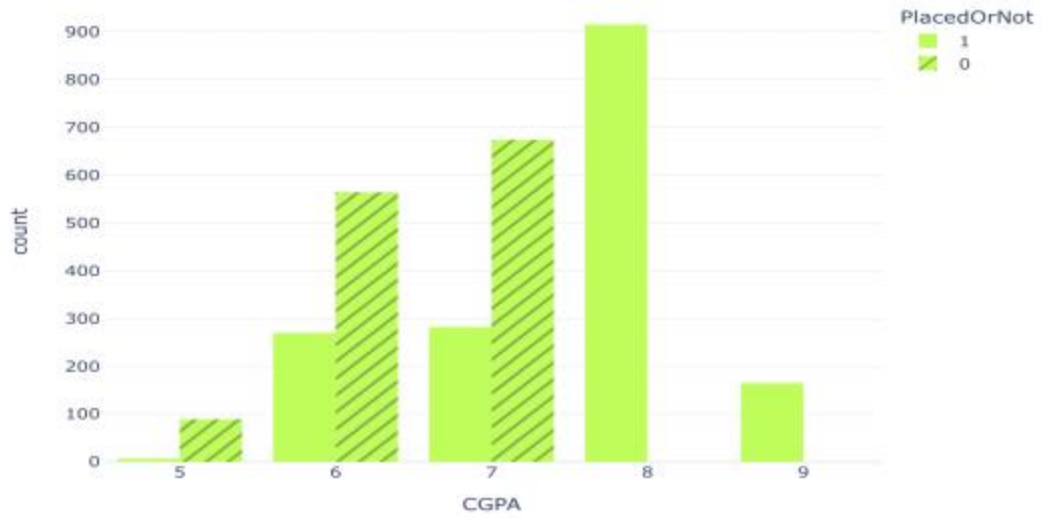


인턴쉽을 한번도 안해봤거나 한번 해본 학생이

가장 많이 취업에 성공하였습니다.

04 EDA 탐색

CGPA vs Placed or Not

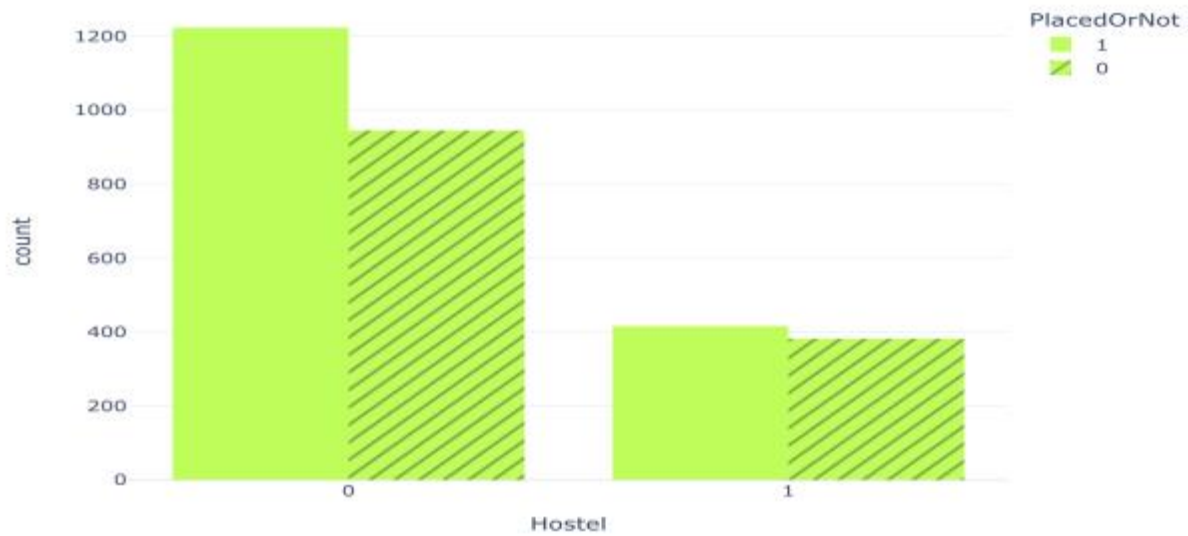


CGPA가 8점인 학생들이 가장 많이 취업이 되었으며,

CGPA가 5점인 학생들은 취업에 어려움을 겪었습니다.

04 EDA 탐색

Hostel vs Placed or Not

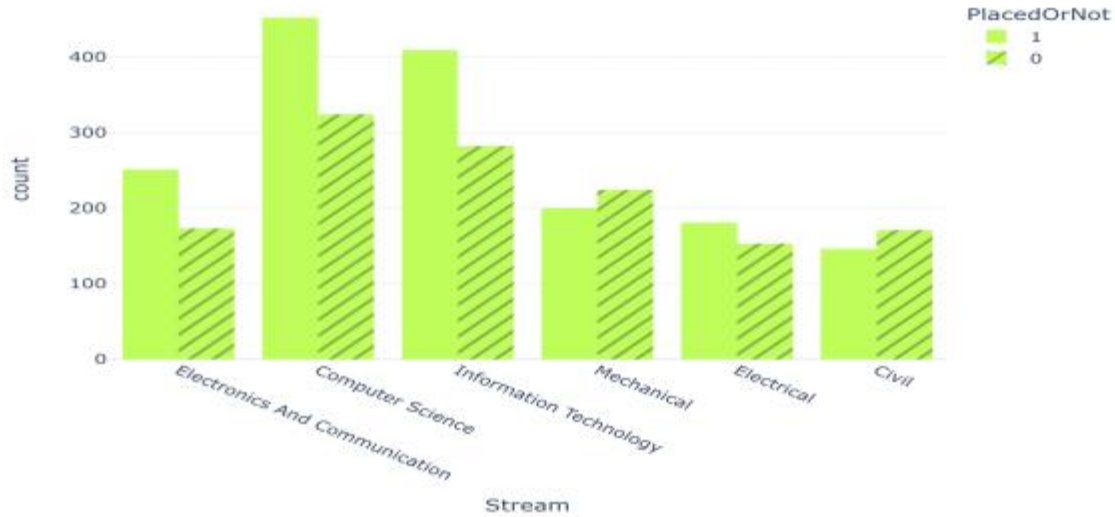


취업에 성공한 많은 학생들이 기숙사가 아닌

자택에서 거주하는 것을 알 수 있습니다.

04 EDA 탐색

Stream vs Placed or Not



취업률이 가장 높은 학과는 컴퓨터 공학과 이고 취업률이 가장 낮은 학과는 전기 공학과 토목 공학과입니다.

04 EDA 탐색

HistoryOfBacklog vs Placed or Not

back log가 있는 학생들 보다 back log가 없는 학생이 취직에 성공적이었습니다.

05

연관 관계

05 연관 관계

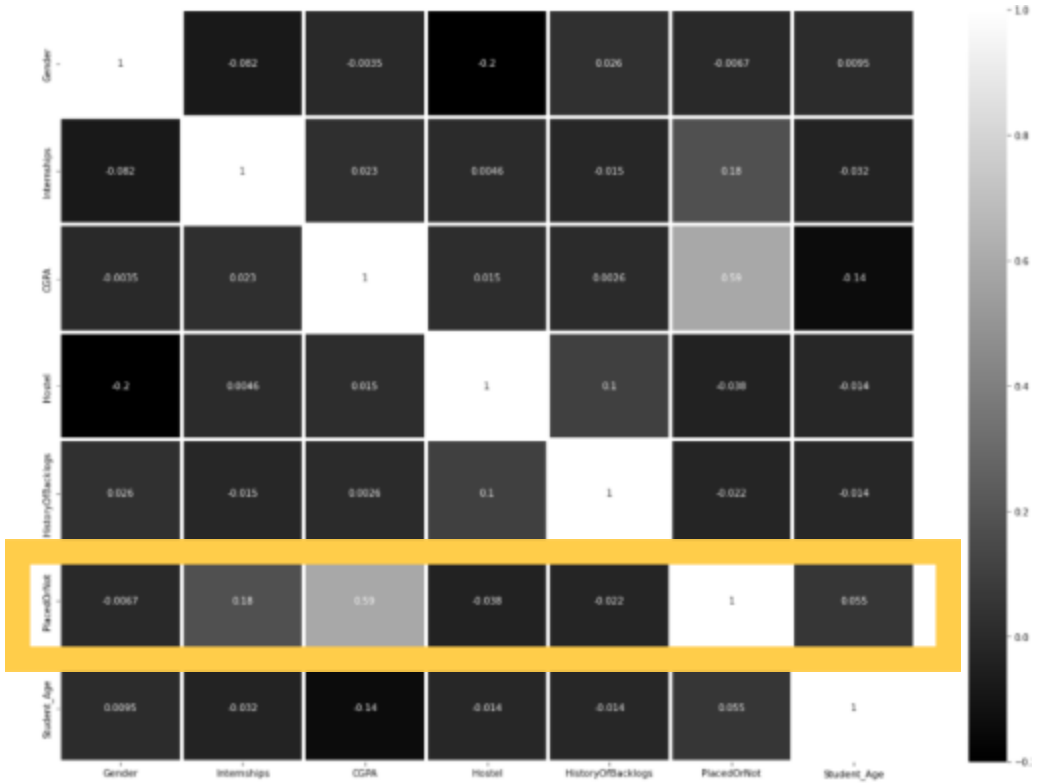
연관 관계

저희의 주된 관심사는 채용 여부입니다.

따라서, placed or not 부분을 보면

CGPA가 0.59 그리고 Intership이 0.18로,

채용 여부와 가장 연관성이 있음을 알 수 있습니다.



05 연관 관계

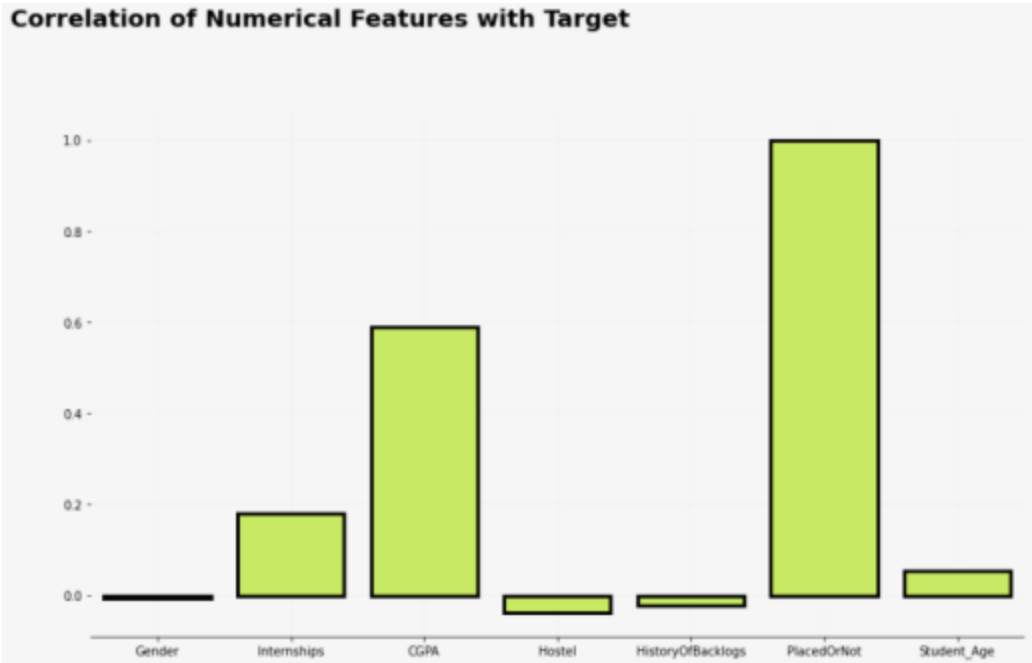
연관 관계

저희의 주된 관심사는 채용 여부입니다.

따라서, placed or not 부분을 보면

CGPA가 0.59 그리고 Intership이 0.18로,

채용 여부와 가장 연관성이 있음을 알 수 있습니다.



06

데이터 스케일링

06 데이터 스케일링

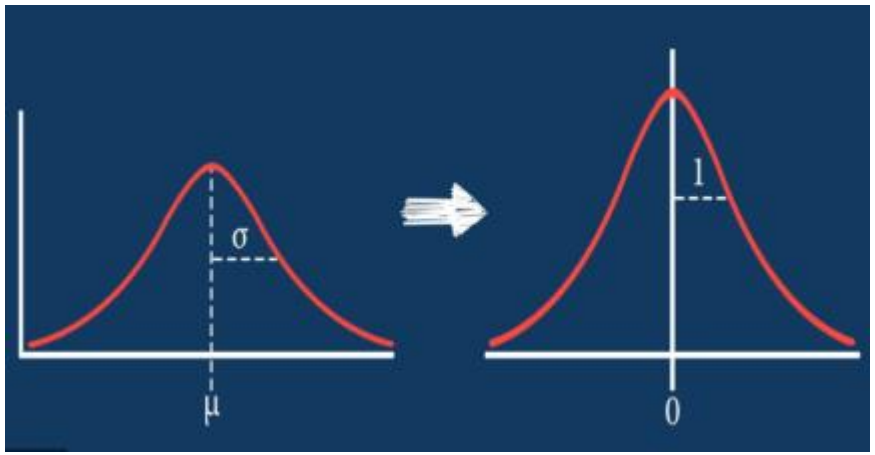
스케일링 이란?

스케일링 이란?

A와 B를 학습시키기 위해 데이터 셋을 정제할 때,

서로 데이터의 스케일이 다르면 예측하는데 큰 영향을 주지 못합니다.

따라서, **스케일링 작업을 통해 모든 특성의 범위를 같게 만들어 주는 작업**을 해야 합니다.



06 데이터 스케일링

데이터 셋 스케일링

```
scaler = StandardScaler()

scaler.fit(df.drop('PlacedOrNot',axis=1))

scaled_features = scaler.transform(df.drop('PlacedOrNot',axis=1))

scaled_features = pd.DataFrame(scaled_features, columns = df.columns[:-1])
scaled_features.head().style.set_properties(**{"background-color": "black","color": "white", "border-
color": "black","font-size":"11.5pt",'width': 200})
```

저는 특성들의 **평균을 0, 분산을 1로 스케일링하여 정규분포로 만드는 StandardScaler**를 사용했습니다.

06 데이터 스케일링

01 스케일링이란?

02 데이터 셋 스케일링

03. 결과

결과

	Student_Age	Gender	Civil	Computer Science	Electrical	Electronics And Communication	Information Technology	Mechanical	Internships	CGPA	Hostel	HistoryOfBacklogs
0	-0.397804	0.445403	-0.345930	-0.595263	-0.356230	2.448527	-0.551123	-0.408409	0.400445	0.957191	1.648269	2.050246
1	-0.397804	-2.245158	-0.345930	1.679930	-0.356230	-0.408409	-0.551123	-0.408409	-0.950773	-0.076310	1.648269	2.050246
2	-0.397804	-2.245158	-0.345930	-0.595263	-0.356230	-0.408409	1.814478	-0.408409	0.400445	-1.109812	-0.606697	-0.487746
3	-0.397804	0.445403	-0.345930	-0.595263	-0.356230	-0.408409	1.814478	-0.408409	-0.950773	0.957191	-0.606697	2.050246
4	-0.397804	0.445403	-0.345930	-0.595263	-0.356230	-0.408409	-0.551123	2.448527	-0.950773	0.957191	1.648269	-0.487746

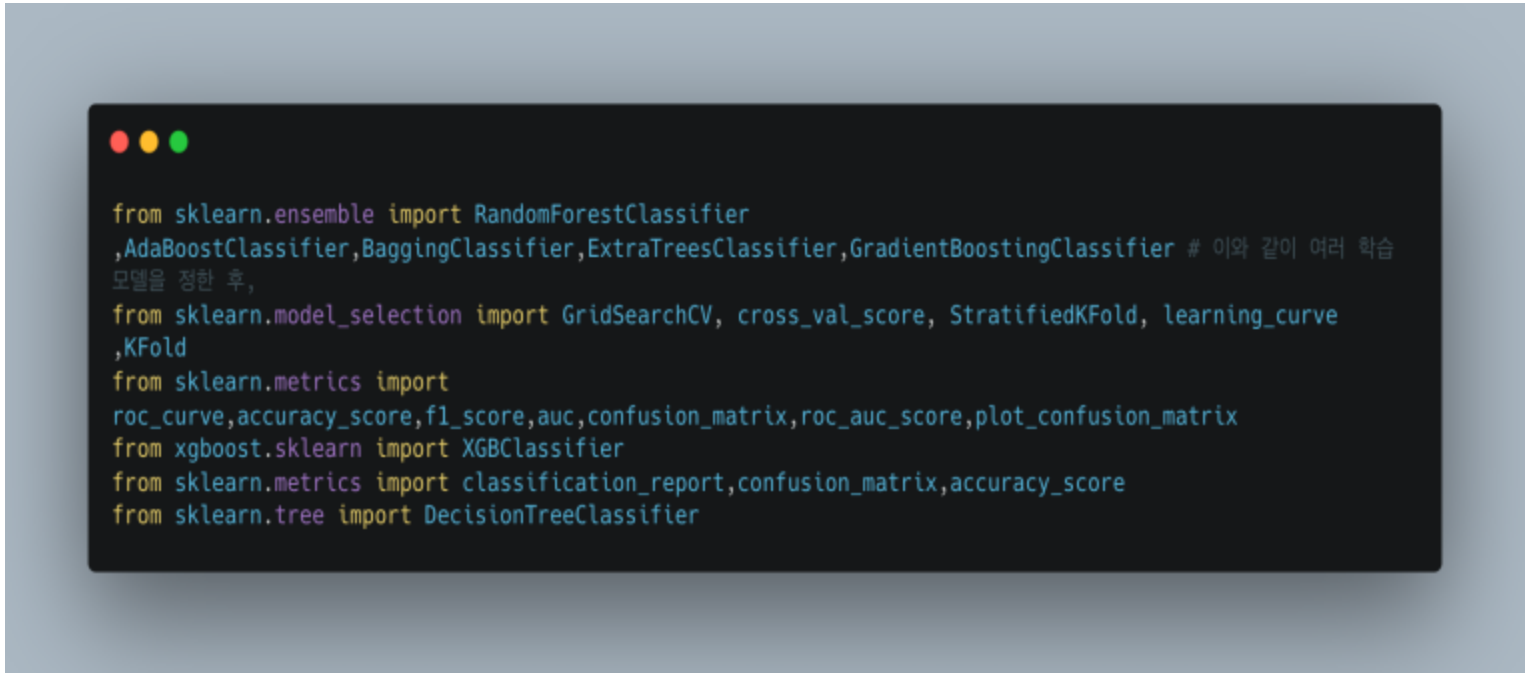
데이터 셋 스케일링의 결과입니다.

07

모델

07 모델

학습 모델 선정



```
from sklearn.ensemble import RandomForestClassifier
,AdaBoostClassifier,BaggingClassifier,ExtraTreesClassifier,GradientBoostingClassifier # 이와 같이 여러 학습
모델을 정한 후,
from sklearn.model_selection import GridSearchCV, cross_val_score, StratifiedKFold, learning_curve
,KFold
from sklearn.metrics import
roc_curve,accuracy_score,f1_score,auc,confusion_matrix,roc_auc_score,plot_confusion_matrix
from xgboost.sklearn import XGBClassifier
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
from sklearn.tree import DecisionTreeClassifier
```

다음과 같이 학습할 모델을 정했습니다.

- RandomForestClassifier
- AdaBoostClassifier
- BaggingClassifier
- ExtraTreesClassifier
- GradientBoostingClassifier

07 모델

01 학습 모델 선정

02. 최적 모델 선정

03 AdaBoostClassifier

최적 모델 선정

```
kfold = StratifiedKFold(n_splits=8,shuffle=True, random_state=42)

rs = 15
clrs = []

clrs.append(AdaBoostClassifier(random_state=rs))
clrs.append(GradientBoostingClassifier(random_state=rs))
clrs.append(RandomForestClassifier(random_state=rs))
clrs.append(ExtraTreesClassifier(random_state = rs))
clrs.append(DecisionTreeClassifier(random_state = rs))

cv_results = []
for clr in clrs :
    cv_results.append(cross_val_score(clr, X_train, y_train , scoring = 'accuracy', cv = kfold,
n_jobs=-1))

cv_means = []
cv_std = []
for cv_result in cv_results:
    cv_means.append(cv_result.mean())
    cv_std.append(cv_result.std())

cv_df = pd.DataFrame({"CrossVal_Score_Means":cv_means,"CrossValerrors": cv_std,"Algo":
["RandomForestClassifier","AdaBoostClassifier","Gradient
Boosting",'ExtraTreesClassifier','DecisionTreeClassifier']})


g = sns.barplot("CrossVal_Score_Means","Algo",data = cv_df,orient = "h",**{'xerr':cv_std},color =
'#d2ff4d')
g.set_xlabel("Mean Accuracy",fontsize = 18)
g = g.set_title("Cross validation scores",fontsize = 24)
plt.figure(figsize = (12,8))
print(cv_df)
```

위 코드로 어떤 모델이 가장 좋은 확률로 학습이 가능한지 알아보겠습니다.

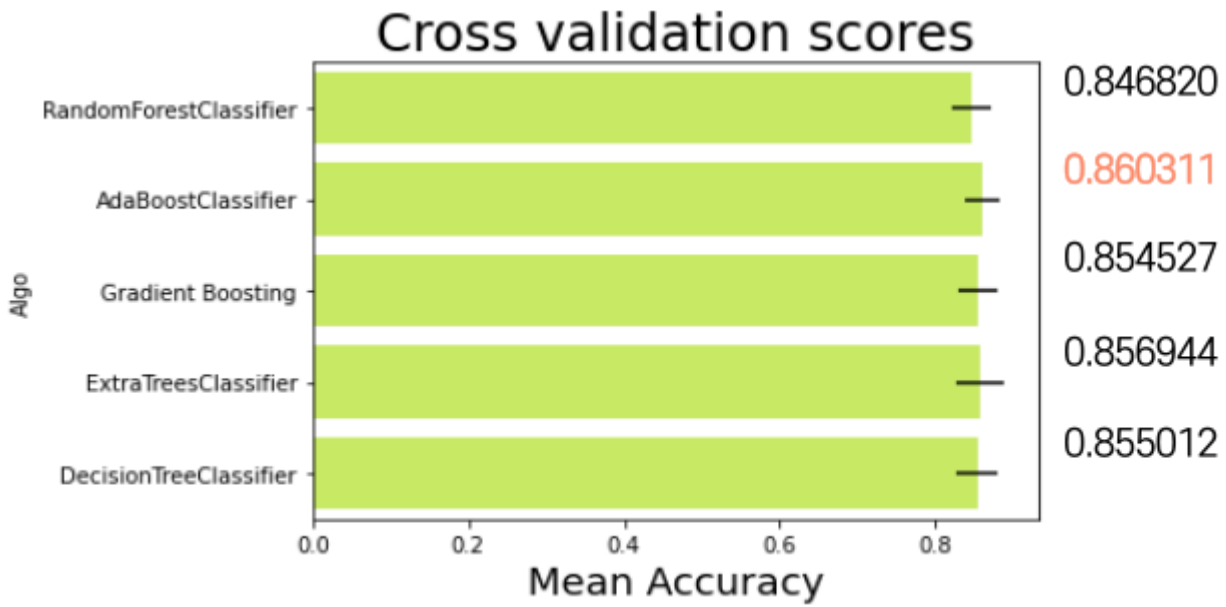
07 모델

01 학습 모델 선정

02. 최적 모델 선정

03 AdaBoostClassifier

최적 모델 선정



결과를 보면 다들 비슷하지만, AdaBoostClassifier가 근소한 차이로 가장 좋은 예측을 나타냄을 볼 수 있습니다.

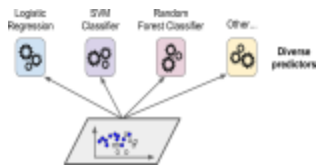
07 모델

01 학습 모델 선정

02. 최적 모델 선정

03 AdaBoostClassifier

AdaBoostClassifier



앙상블

앙상블은 조화를 의미합니다. 이는 단어의 의미처럼 **여러 개의 모델을 이용해서 학습시켜 그 결과를 토대로 더 정확한 예측값을 도출하는 모델**입니다.

〈부스팅 알고리즘〉

부스팅 (Boosting)

부스팅은 여러 개의 약한 학습기를 순차적으로 학습 및 예측하면서 **이전의 예측값 데이터에 가중치 부여를 통해 오류를 개선해 나가면서 학습하는 기계학습 방법**입니다.

부스팅의 대표적인 종류는 AdaBoost와 GradientBoost 등이 있습니다.

AdaBoost

Adaboost 알고리즘은 분류 기반 기계학습 모형으로, **예측 성능이 조금 낮은 약한 학습기를 다량 구축 및 조합하여 가중치 수정을 통해 좀 더 나은 성능을 발휘하는 하나의 강한 분류기를 합성하는 방법의 알고리즘** 입니다.

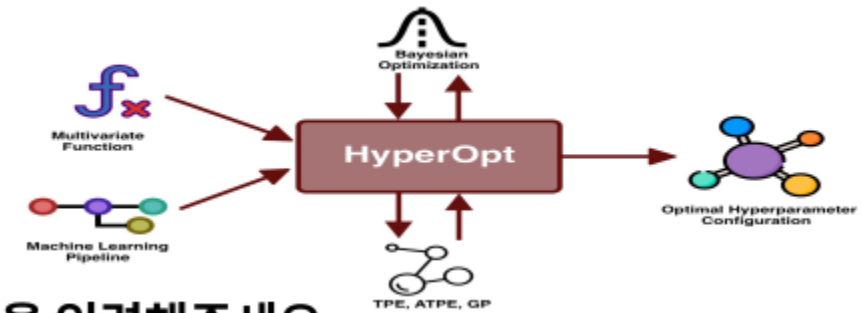
이는 약한 분류기의 실수를 통해 가**중치를 반복적으로 수정 및 결합하여 정확도를 높일 수 있으며 과적합 현상이 적게 발생**하여 예측 성능을 저해하지 않는 장점이 있습니다.

08

학습

08 학습

하이퍼 파람 튜닝이란?



본문 내용을 입력해주세요.

하이퍼 파라미터 튜닝

하이퍼 파라미터 튜닝은 **모델 학습 프로세스를 제어 할 수 있게 하는 조정 가능한 매개 변수** 입니다.

예를 들어 신경망을 사용하여 숨겨진 레이어 수와 각 레이어의 노드 수를 결정합니다.

이에 대한 튜닝은 **최상의 성능을 발휘하는 하이퍼 매개 변수 구성을 찾는 프로세스** 입니다.

따라서 이를 통해 최상의 파라미터 조합을 찾을 수 있습니다.

08 학습

하이퍼 파람 튜닝

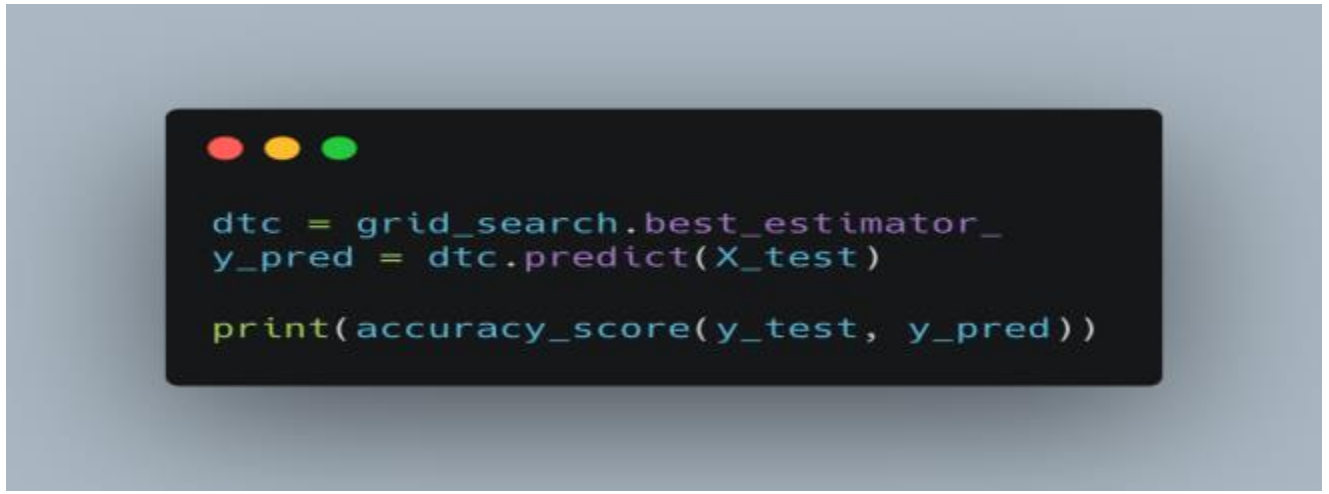
```python
from sklearn.model_selection import GridSearchCV
grid_params = {
    'criterion' : ['gini', 'entropy'], # DecisionTreeClassifier의 분류 기준
    'max_depth' : [3, 5, 7, 10], # 레이어의 깊이
    'min_samples_split' : range(2, 10, 1), # 분류시 최소 포함 샘플 갯수
    'min_samples_leaf' : range(2, 10, 1) # 최대 리프 노드의 개수
}

grid_search = GridSearchCV(dtc, grid_params, cv = 5, n_jobs = -1, verbose = 1)
grid_search.fit(X_train, y_train)
```

하이퍼 파라미터 튜닝으로 더 좋은 모델을 만들어 보겠습니다. 이는 모델을 만들때 개발자가 직접 제어하는 코드로, **이 값에 따라 모델 성능 차이**가 납니다.

또, **GridSearch는 내가 지정한 범위 내의 모든 파라미터 조합을 계산해 가장 좋은 예측 결과를 나타내는 하이퍼 파람 조합**을 알 수 있습니다.

08 학습

최적 하이퍼 파람



```
dtc = grid_search.best_estimator_
y_pred = dtc.predict(X_test)

print(accuracy_score(y_test, y_pred))
```

```
{'criterion': 'entropy', 'max_depth': 5, 'min_samples_leaf': 2, 'min_samples_sp
lit': 2}
0.8622381835032437
```

GridSearch를 통해 알아낸 최적의 예측 결과와 최적의 하이퍼 파라미터 입니다.

08 학습

01 하이퍼 파람 튜닝 이란?

02. 하이퍼 파라미터 튜닝

03. 최적 하이퍼 파람

04. 모델 학습

모델 학습

```
ada = AdaBoostClassifier(base_estimator = dtc, algorithm = 'SAMME.R', learning_rate = 0.001,
n_estimators = 50)
ada.fit(X_train, y_train)

print(accuracy_score(y_test, y_pred))
```

```
0.8696629213483146
```

GridSearch를 통해 알아낸 최적의 하이퍼 파라미터로 AdaBoostClassifier 모델로 학습시키는 코드와 결과 입니다.

기존의 정확도 0.860311에서 0.8696292134831460으로 올라간 것을 볼 수 있습니다.

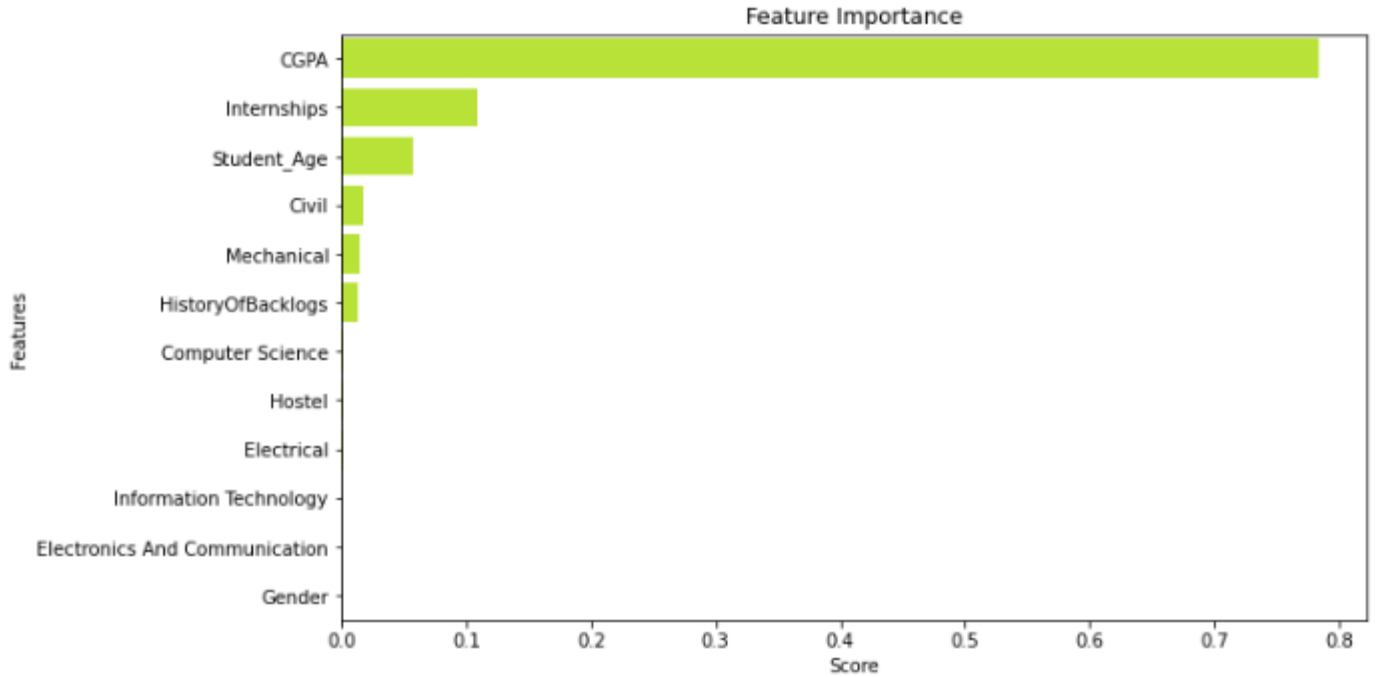
따라서, 하이퍼 파라미터를 알아내고 학습시킨 결과 **확실한 성능 향상**이 있었음을 알 수 있었습니다.

09

결론

09 결론

결론



학생의 캠퍼스 채용 여부와 가장 연관성 있는 데이터는 CGPA이며,
그 다음으로 Internships임을 알 수 있습니다.

감사합니다.

컴퓨터공학전공 2017106119 염상권