Open& Close_hospital

202121656 인공지능학과 장서연

Contents

01 Data

병원의 개업과 폐업 여부를 나누는 데이터로 변수의 개수 총 58개로 구성

inst_id - 각 파일에서의 병원 고유 번호
OC – 영업/폐업 분류, 2018년 폐업은 2017년 폐업으로 간주함
sido – 병원의 광역 지역 정보
sgg – 병원의 시군구 자료
openDate – 병원 설립일
- 병원이 갖추고 있는 병상의 수
instkind – 병원, 의원, 요양병원, 한의원, 종합병원 등 병원의 종류
· 종합병원 : 입원환자 100명 이상 수용 가능
· 병원 : 입원 환자 30명 이상 100명 미만 수용 가능
· 의원 : 입원 환자 30명 이하 수용 가능
· 한방 병원(한의원) : 침술과 한약으로 치료하는 의료 기관.

revenue1 – 매출액, 2017(회계년도)년 데이터를 의미함
salescost1 – 매출원가, 2017(회계년도)년 데이터를 의미함
sga1 - 판매비와 관리비, 2017(회계년도)년 데이터를 의미함
salary1 – 급여, 2017(회계년도)년 데이터를 의미함
noi1 – 영업외수익, 2017(회계년도)년 데이터를 의미함
noe1 – 영업외비용, 2017(회계년도)년 데이터를 의미함
Interest1 – 이자비용, 2017(회계년도)년 데이터를 의미함
ctax1 – 법인세비용, 2017(회계년도)년 데이터를 의미함
Profit1 – 당기순이익, 2017(회계년도)년 데이터를 의미함
liquidAsset1 – 유동자산, 2017(회계년도)년 데이터를 의미함
quickAsset1 – 당좌자산, 2017(회계년도)년 데이터를 의미함
receivableS1 - 미수금(단기), 2017(회계년도)년 데이터를 의미함
inventoryAsset1 – 재고자산, 2017(회계년도)년 데이터를 의미함

nonCAsset1 – 비유동자산, 2017(회계년도)년 데이터를 의미함
tanAsset1 – 유형자산, 2017(회계년도)년 데이터를 의미함
OnonCAsset1 - 기타 비유동자산, 2017(회계년도)년 데이터를 의미함
receivableL1 – 장기미수금, 2017(회계년도)년 데이터를 의미함
debt1 – 부채총계, 2017(회계년도)년 데이터를 의미함
liquidLiabilities1 – 유동부채, 2017(회계년도)년 데이터를 의미함
shortLoan1 – 단기차입금, 2017(회계년도)년 데이터를 의미함
NCLiabilities1 – 비유동부채, 2017(회계년도)년 데이터를 의미함
longLoan1 – 장기차입금, 2017(회계년도)년 데이터를 의미함
netAsset1 – 순자산총계, 2017(회계년도)년 데이터를 의미함
surplus1 – 이익잉여금, 2017(회계년도)년 데이터를 의미함

01 Data

Data 10개 (독립변수 9개 + 종속변수 1개)

X = hospital[['openDate','bedCount','revenue1','salescost1','sga1','salary1','liquidAsset1','nonCAsset1','netAsset1','surplus1']]

OC - open/ close
openDate - 병원 설립일
bedCoount- 병원이 가진 병상의 수
revenue1 - 매출액 2017
salescost1 - 매출 원가
sga1- 관리비
salary- 급여
liquidAsset1 - 유동자산
nonCAsset1 - 비유동자산
netAsset1 - 순자산총계
surplus1- 이익잉여금

```
Data columns (total 11 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   OC             301 non-null    object
 1   openDate       301 non-null    int64
 2   bedCount       296 non-null    float64
 3   revenue1       293 non-null    float64
 4   salescost1     293 non-null    float64
 5   sga1           293 non-null    float64
 6   salary1        293 non-null    float64
 7   liquidAsset1   293 non-null    float64
 8   nonCAsset1     293 non-null    float64
 9   netAsset1      293 non-null    float64
 10  surplus1       293 non-null    float64
```

02 목적

인공지능이 폐업을 예측하고 설명한다면?

(이것만 피하면 된다!)

수 많은 병원이 개업하고 폐업하는 시대.

이때 인공지능이 미래의 개폐업의 여부를 알려준다면 어떨까?

폐업에 가장 결정적인 원인을 알고, 예방한다면 피해를 줄일 수 있지 않을까?

이를 확장해서 모든 소상공인에게 적용시킬 수 있지 않을까?

01. 폐업을 예측할 수 있을까?

- Logistic Regression
- Naive Bayse
- Support Vector Machine
- K-Nearest Neighbor
- Decision Tree

02. 가장 영향력이 큰 변수는?

- 규모? 급여? 고정지출?
- 의사결정 나무
- 변수 중요도
- 설명가능성- SHAP

03 Preprocessing

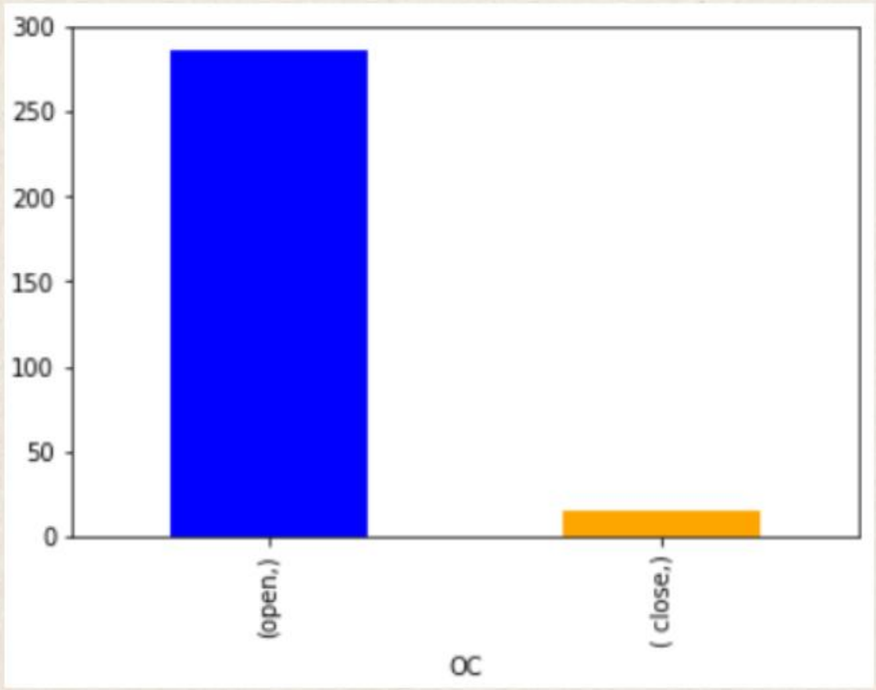
효과적인 학습을 위한 준비 과정

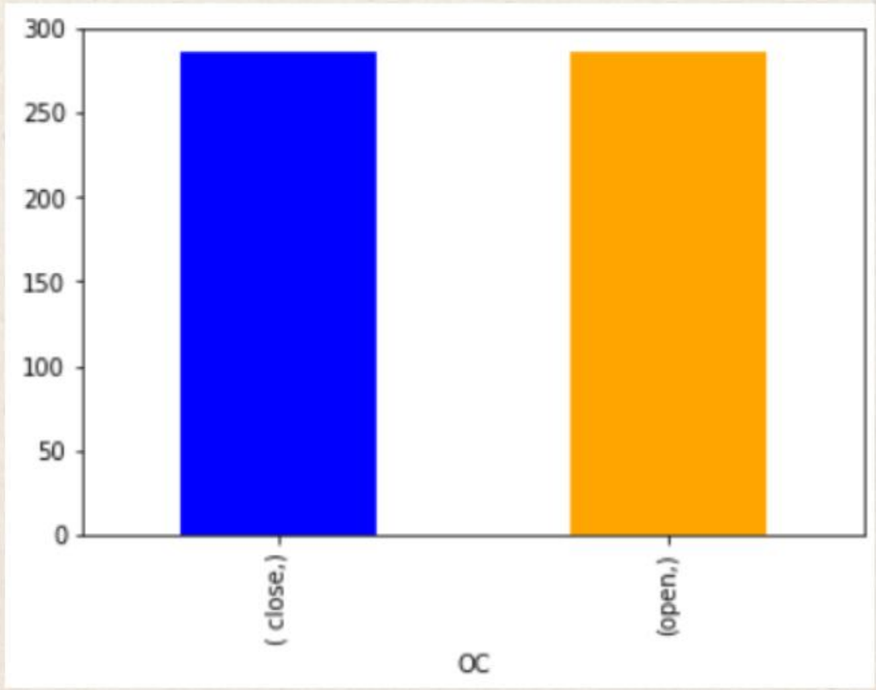
Over Sampling	Nan 대체	상관계수	표준화
Open Data 쏠림 현상	결측치 → 각 중앙값	변수 선정 및 상관성	평균및 분산 조정
ros = RandomOverSampler() X,y = ros.fit_resample(X,y)	hospital_train['bedCount'] = hospit al_train['bedCount'].fillna(med)	hospital_train.corr (method='pearson')	scaler = StandardScaler() X_s = pd.DataFrame(scaler.fit_tran sform(hospital_train))

03 Preprocessing

1) Over Sampling

```
ros = RandomOverSampler()
X,y = ros.fit_resample(X,y)
count = y.value_counts()
count.plot(kind = 'bar',color=['blue','orange'])
```

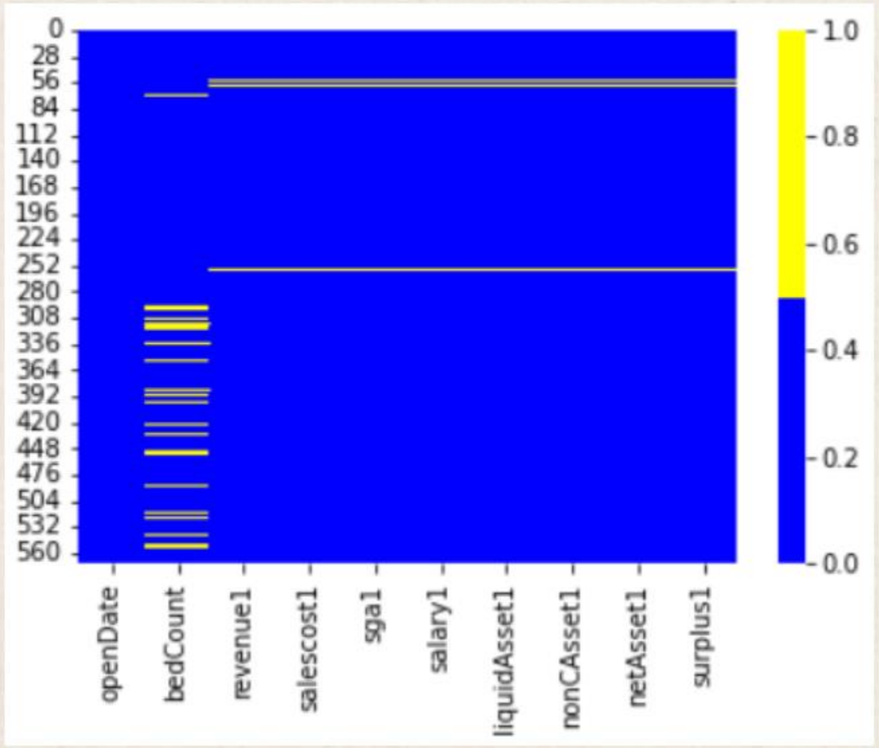




03 Preprocessing

2) NaN

```
# fillna() 로 누락 데이터 대체- bedCount
hospital_train = X
med = hospital_train['bedCount'].median()
print(med)
hospital_train['bedCount'] = hospital_train['bedCount'].fillna(med)
```



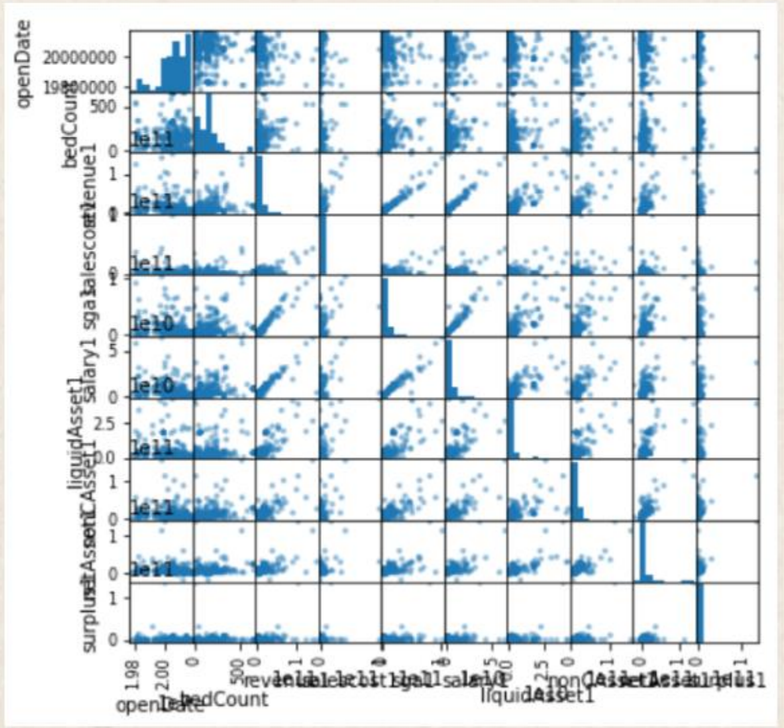

```
openDate - 0%
bedCount - 0%
revenue1 - 0%
salescost1 - 0%
sga1 - 0%
salary1 - 0%
liquidAsset1 - 0%
nonCAsset1 - 0%
netAsset1 - 0%
surplus1 - 0%
```

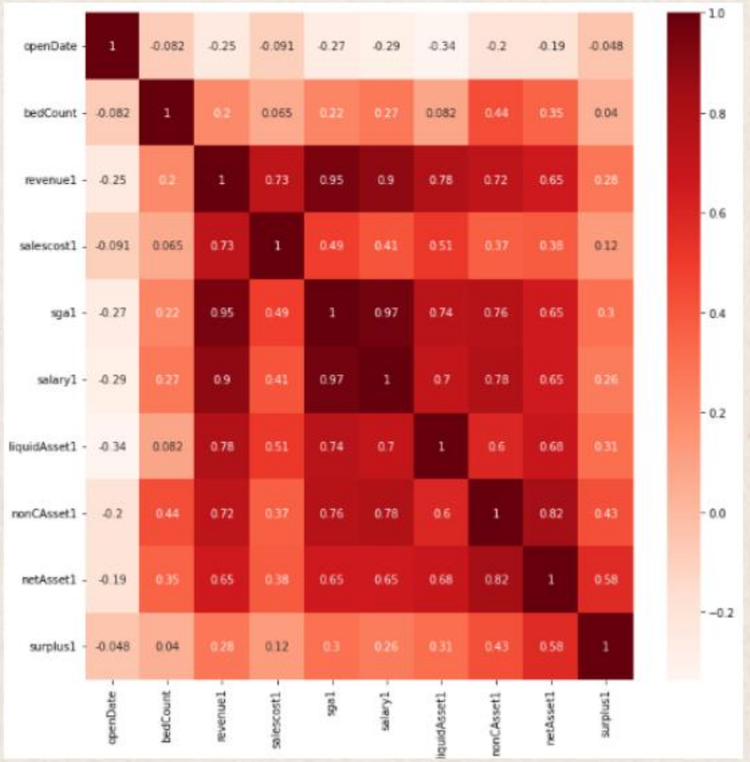
03 Preprocessing

3)상관계수

상관계수가 너무 큰 'sga1' 제거

hospital_train = hospital_train.drop(['sga1'],axis=1)





03 Preprocessing

4) Standard Scaler

```
scaler = StandardScaler()
X_s = pd.DataFrame(scaler.fit_transform(hospital_train))
X_s.head()
```

	0	1	2	3	4	5	6	7	8
0	0.115740	0.093249	-0.331562	-0.271418	-0.354525	-0.397898	-0.557025	-0.258209	0.047851
1	-1.041152	1.760734	-0.305090	-0.222888	-0.304292	-0.284323	-0.291737	-0.224236	-0.194325
2	1.148402	2.172284	-0.537418	-0.178134	-0.628398	-0.534819	-0.733844	-0.510216	-0.194325
3	-0.692193	1.356280	4.043729	-0.271418	4.335347	1.827013	2.446312	0.832720	1.204562
4	-0.117493	0.242258	2.540416	-0.271418	3.180658	0.583226	2.482352	-0.531435	1.266367

04 Model

Classification

- Logistic Regression

```
테스트 집합 정확도(accuracy): 0.687
테스트 집합 정밀도(precision): 0.725
테스트 집합 재현율(recall): 0.537
테스트 집합 f1 점수: 0.617
```

- SVM - Poly

```
테스트 집합 정확도(accuracy): 0.861
테스트 집합 정밀도(precision): 0.839
테스트 집합 재현율(recall): 0.870
테스트 집합 f1 점수: 0.855
```

- Decision Tree

```
DecisionTreeClassifier(criterion='gini')
학습 집합 정확도: 1.000
테스트 집합 정확도(accuracy): 0.957
테스트 집합 정밀도(precision): 1.000
테스트 집합 재현율(recall): 0.907
테스트 집합 f1 점수: 0.951
```

- KNN

```
학습 집합 정확도: 0.965
테스트 집합 정확도(accuracy): 0.948
테스트 집합 정밀도(precision): 1.000
테스트 집합 재현율(recall): 0.889
테스트 집합 f1 점수: 0.941
```

04 Model

Accuracy + F1 score

KNN

최적의 n_neighbors 찾기

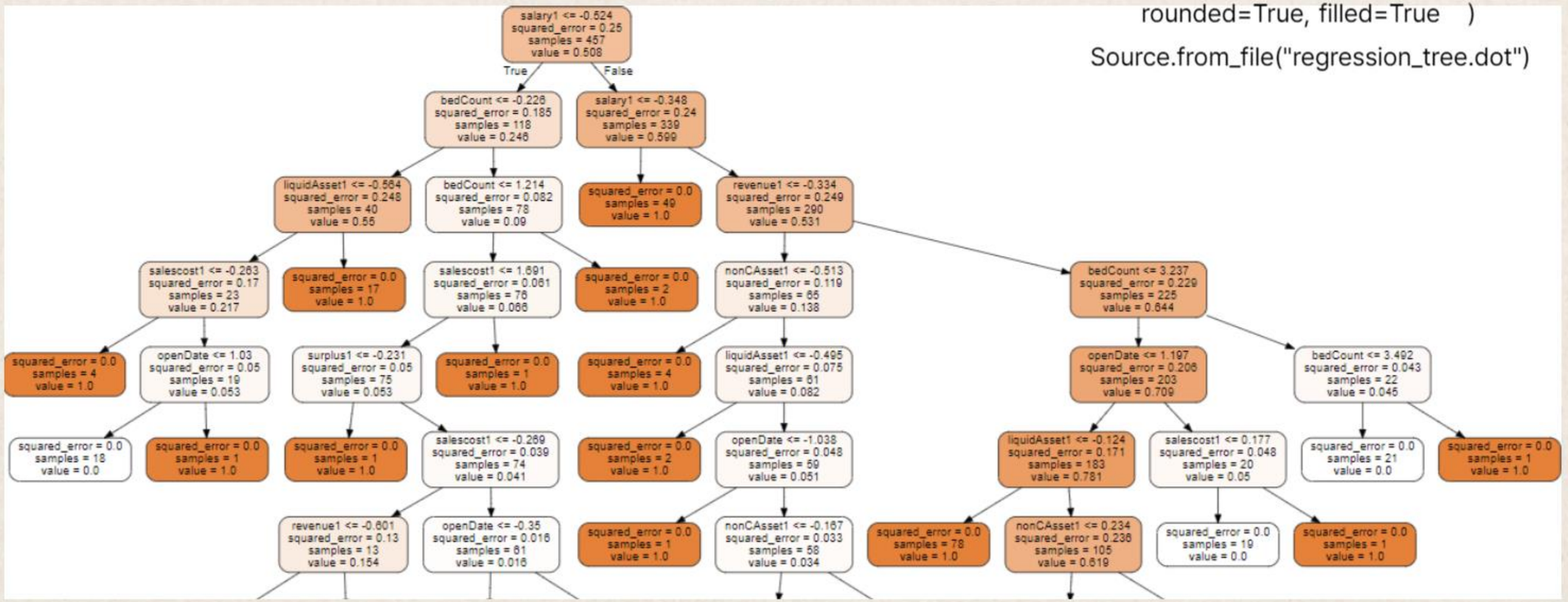
k: 1, accuracy: 95.65
k: 2, accuracy: 95.65
k: 3, accuracy: 95.65
k: 4, accuracy: 94.78
k: 5, accuracy: 94.78
k: 6, accuracy: 92.17
k: 7, accuracy: 92.17
k: 8, accuracy: 91.30
k: 9, accuracy: 91.30
k: 10, accuracy: 89.57

Model	Accuracy	F1 Score
LogisticRegression(multi_class="multinomial", max_iter=1000)	0.687	0.617
GaussianNB()	0.522	0.225
LinearSVC(C=1, max_iter = 10000)	0.696	0.632
SVC(kernel="poly", degree=3, coef0=1, C=5)	0.861	0.855
SVC(kernel="rbf", gamma=5, C=0.001)	0.470	0.639
DecisionTreeClassifier(criterion='gini')	0.957	0.951
DecisionTreeClassifier(criterion='gini', max_depth=4, random_state=42)	0.800	0.807
DecisionTreeClassifier(criterion='gini', max_depth=10, random_state=42)	0.939	0.931
KNeighborsClassifier(n_neighbors=5, n_jobs=-1)	0.965	0.941

04 Model

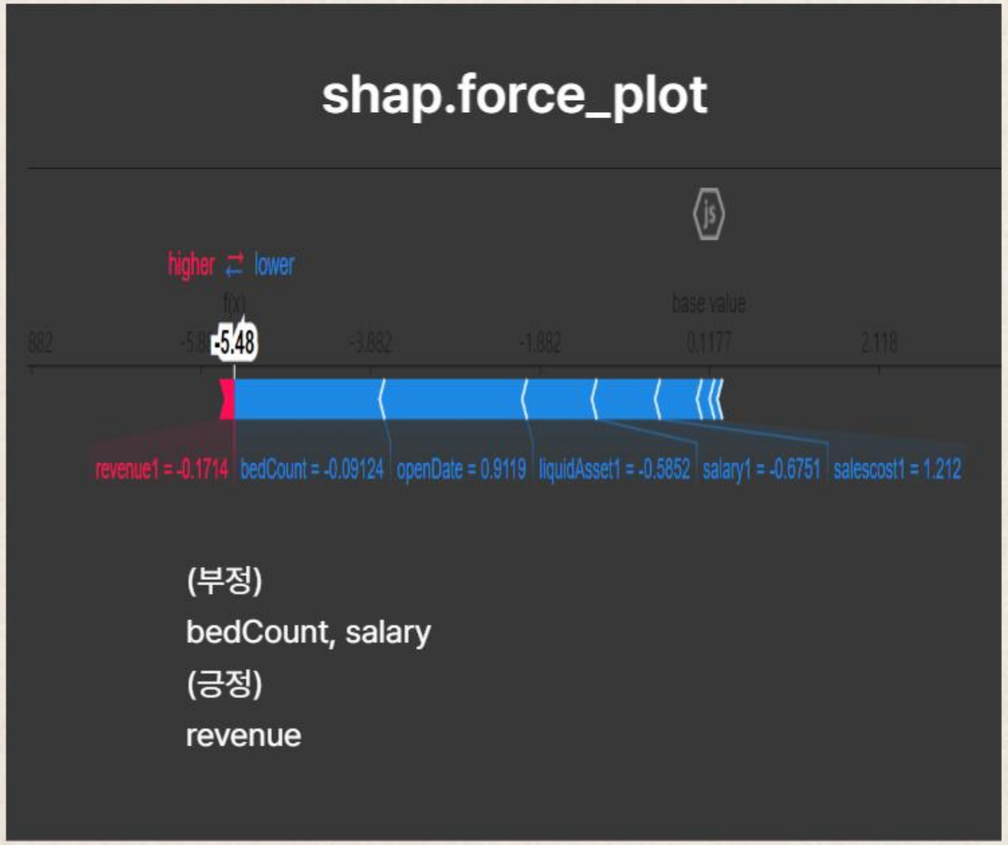
Decision Tree 시각화

```
tree_reg = DecisionTreeRegressor()
tree_reg.fit(X, y)
export_graphviz( tree_reg,
        out_file="regression_tree.dot",
        feature_names=X.columns,
        rounded=True, filled=True    )
Source.from_file("regression_tree.dot")
```



05 변수중요도

새플리

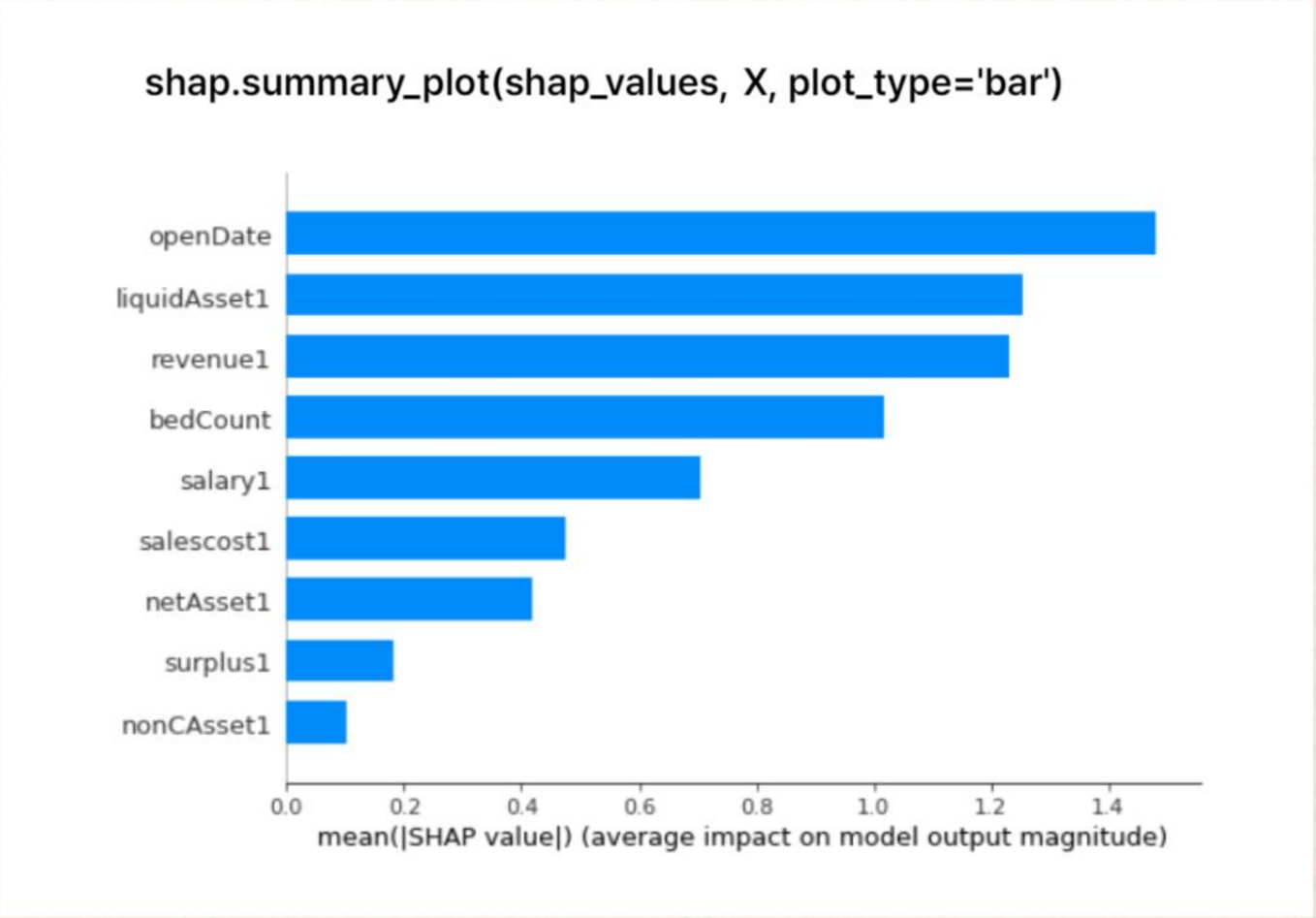


(부정)
bedCount, salary
(긍정)
revenue

05 변수중요도

Decision Tree및 새플리

openDate - 병원 설립일
bedCoount- 병원이 가진 병상의 수
revenue1 - 매출액 2017
salescost1 - 매출 원가
salary- 급여
liquidAsset1 - 유동자산
nonCAsset1 - 비유동자산
netAsset1 - 순자산총계
surplus1- 이익잉여금



06 중요성및 활용방안

의의와 효과

의의
AI가 폐업을 예측할 수 있다

기존에 구성되어 있던 58개의 변수가 아닌
선정한 8개의 변수만으로
병원의 운영 여부를 예측할 수 있다

ㅂ병원 설립일, 병상의 수, 매출액, 매출 원가
급여, 유동자산, 비유동자산, 순자산총계
이익잉여금

효과
폐업에 중요한 항목을 알고 예방할 수 있다

폐업 여부에 중요한 역할을 하는 5가지의 변수
병원 설립일, 매출액, 유동자산, 병상의 수, 급여를 바탕으로
오래된 병원에 경고, 개업시 유동자산을 확인하는등
폐업을 예측하고 미리 준비할 수 있다.

활용
병원, 가게등 소상공인 창업에 활용할 수 있다

설립일, 매출액, 유동자산, 규모, 급여등은
대부분의 가게에도 해당됨으로
병원의 Data를 통해 일반 소상공인에 경우에도 확장하여
고려할 수 있다.

Thank you

2022-2 머신러닝

인공지능학과 202121656

장서연
