공공조달 빅데이터 경진대회 기획서

1. 과제 목표

“공사 입찰공고별 투찰업체 내역”을 토대로 조달청에서 입찰이 이루어질 때 ‘업종제한내용’, ‘낙찰자결정방법’, ‘입찰률’에 따라 약소한 차이로 낙찰에 실패하는 경우를 파악했다. 대학입시합격예측 프로그램과 같이 현재 희망 금액으로 신청서를 제출했을 때 결과를 미리 예측할 수 있으면 적절한 금액으로 입찰 확률을 높일 수 있을 것이라고 생각했다. 따라서, 기존 데이터들을 활용하여 낙찰여부를 예측하는 머신러닝 모델을 만드는 것이 이번 프로젝트의 목표이다.

2. 핵심내용

공공조달 홈페이지에 있는 공사 입찰 관련 데이터를 가공한 후, 머신러닝 알고리즘인 랜덤포레스트를 이용해 모델을 만들었다. API서비스를 통해 사용자가 어떤 프로젝트에 대해 자신이 제시할 가격을 입력하면 모델이 학습한 데이터를 바탕으로 입찰 성공 여부를 나타내준다.

3. 세부내용

ㅇ 활용데이터 URL/목록명/활용내용

http://data.g2b.go.kr:8275/pt/pubdata/moveCntrwkBidPblancAcctoBddprEntrpsPop.do

목록명: 조달정보개방포털 파일데이터 공사 입찰공고별 투찰업체 내역

활용내용: 낙찰자결정방법, 입찰률, 업종제한내용, 낙찰여부

ㅇ 개발과정(데이터가공->모델학습->모델평가)

1. 데이터가공

<원본 데이터>

index	등록유형	조달구분	공고시스템명	입찰공고번호	입찰공고차수	공고게시일자	공고명	공사현장	입찰방식	낙찰자결정적용법규	낙찰자결정방법	입찰계약방법	긴급공고여부	업종제한여부	업종제한내용	지역의무공동도급여부	지역제한내용	공동도급구성방식명	기초금액
0	나라장터(G2B)	자체조달	NaN	20220105921	0	20220109	2022년 처인구 도로시설물(철물) 유지보수 연간단가공사(남부)	경기도 용인시 처인구	전자입찰	지방계약법	수의(견적제출)	수의(소액)-견적입찰(2인 이상 견적 제출)	N	Y	[금속창호,지붕건축물조립공사업(4991)]업종 또는[토목공사업(0001)]	N	경기도 용인시[41460]	공동수급불허	200000000.0
1	나라장터(G2B)	자체조달	NaN	20220105921	0	20220109	2022년 처인구 도로시설물(철물) 유지보수 연간단가공사(남부)	경기도 용인시 처인구	전자입찰	지방계약법	수의(견적제출)	수의(소액)-견적입찰(2인 이상 견적 제출)	N	Y	[금속창호,지붕건축물조립공사업(4991)]업종 또는[토목공사업(0001)]	N	경기도 용인시[41460]	공동수급불허	200000000.0
2	나라장터(G2B)	자체조달	NaN	20220105921	0	20220109	2022년 처인구 도로시설물(철물) 유지보수 연간단가공사(남부)	경기도 용인시 처인구	전자입찰	지방계약법	수의(견적제출)	수의(소액)-견적입찰(2인 이상 견적 제출)	N	Y	[금속창호,지붕건축물조립공사업(4991)]업종 또는[토목공사업(0001)]	N	경기도 용인시[41460]	공동수급불허	200000000.0

↓ 원본 데이터에서 모델에 학습시킬 데이터를 추출

	낙찰자결정방법	입찰률	업종제한내용	낙찰여부
0	수의(견적제출)	87.771	[금속창호,지붕건축물조립공사업(4991)]업종 또는[토목공사업(0001)]	Y
1	수의(견적제출)	87.835	[금속창호,지붕건축물조립공사업(4991)]업종 또는[토목공사업(0001)]	N
2	수의(견적제출)	87.843	[금속창호,지붕건축물조립공사업(4991)]업종 또는[토목공사업(0001)]	N
3	수의(견적제출)	88.014	[금속창호,지붕건축물조립공사업(4991)]업종 또는[토목공사업(0001)]	N
4	수의(견적제출)	88.062	[금속창호,지붕건축물조립공사업(4991)]업종 또는[토목공사업(0001)]	N

<데이터 피처 : 낙찰여부에 영향을 주는 열 > <데이터 레이블>

↓

	낙찰자결정방법	입찰률	업종명	낙찰여부
0	수의(견적제출)	87.771	[건설업]	Y
1	수의(견적제출)	87.835	[건설업]	N
2	수의(견적제출)	87.843	[건설업]	N
3	수의(견적제출)	88.014	[건설업]	N
4	수의(견적제출)	88.062	[건설업]	N

<포괄적인 데이터 분류를 위해 업종제한내용을 업종명으로 변경>

↓

```
ads = ADASYN(random_state = 42, n_neighbors =5)
X_ads, y_ads = ads.fit_resample(X_data, y_data)

print(Counter(y_data))
print(Counter(y_ads))
```

```
Counter({'N': 243395, 'Y': 1505})
Counter({'Y': 243445, 'N': 243395})
```

<데이터 레이블 오버샘플링 : 'Y'와 'N' 비율 맞춤>

2. 모델학습

```
X_train, X_test, y_train, y_test = train_test_split(X_ads.to_numpy(), y_ads.to_numpy(),random_state=156,test_size = 0.2)


rf_clf = RandomForestClassifier()
rf_clf.fit(X_train, y_train)
```

<머신러닝 분류알고리즘 랜덤포레스트 모델 학습>

3. 모델평가

① 정확도 = 예측 결과가 동일한 데이터 건수(TN) / 전체 예측 데이터 건수(TP)

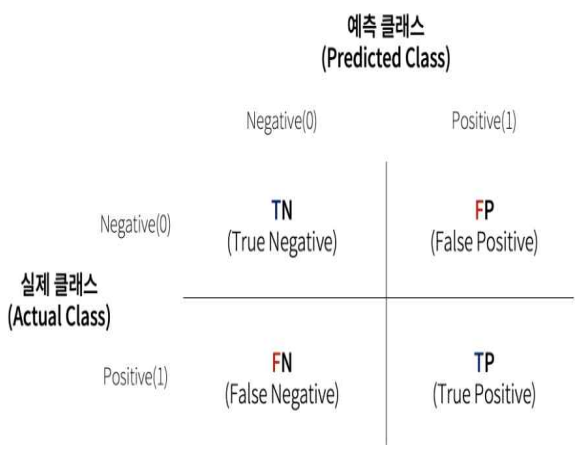
```
print(accuracy_score(y_test, random_forest_pred))
```

```
0.9714998767562238
```

② 오차행렬 : 학습된 분류 모델이 예측을 수행하면서 얼마나 헷갈리고 있는지도 함께 보여주는 지표

```
# 2.오차행렬

from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, random_forest_pred)
print(cm)
```

```
[[47359  1366]
 [ 1409 47234]]
```



③ 정밀도 : positive로 예측한 값들 중에 실제로 positive한 값의 비율

```
# 3.정밀도

from sklearn.metrics import precision_score

print(precision_score(y_test,random_forest_pred, pos_label="Y"))
```

$$Precision = \frac{TP}{FP + TP}$$

```
0.9718930041152264
```

④ 재현율 : 실제 값이 positive인 값들 중에 예측을 positive로 한 값의 비율

```
# 4.재현율

from sklearn.metrics import recall_score

print(recall_score(y_test,random_forest_pred, pos_label="Y"))
```

$$Recall = \frac{TP}{FN + TP}$$

```
0.9710338589313982
```

⑤ F값 : 정밀도와 재현율을 결합한 지표

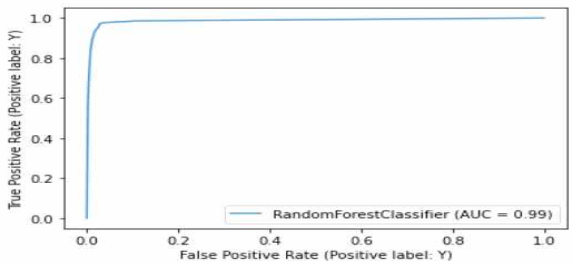
```
# 5.F1스코어

from sklearn.metrics import f1_score

f1 = f1_score(y_test, random_forest_pred, pos_label="Y")
print(f1)
```

$$F1 = \frac{2}{\frac{1}{recall} + \frac{1}{precision}} = 2 \times \frac{precision \cdot recall}{precision + recall}$$

```
0.9714632415700873
```

⑥ ROC 곡선 : FPR(X축)과 TPR(Y축)의 관계를 그린 곡선

```
ax = plt.gca()
rfc_disp = RocCurveDisplay.from_estimator(rf_clf_from_joblib, X_test, y_test, ax=ax, alpha=0.8)
plt.show()
```



ㅇ 분석기법

머신러닝 지도학습의 대표적인 유형인 랜덤포레스트 분류 기법 사용.

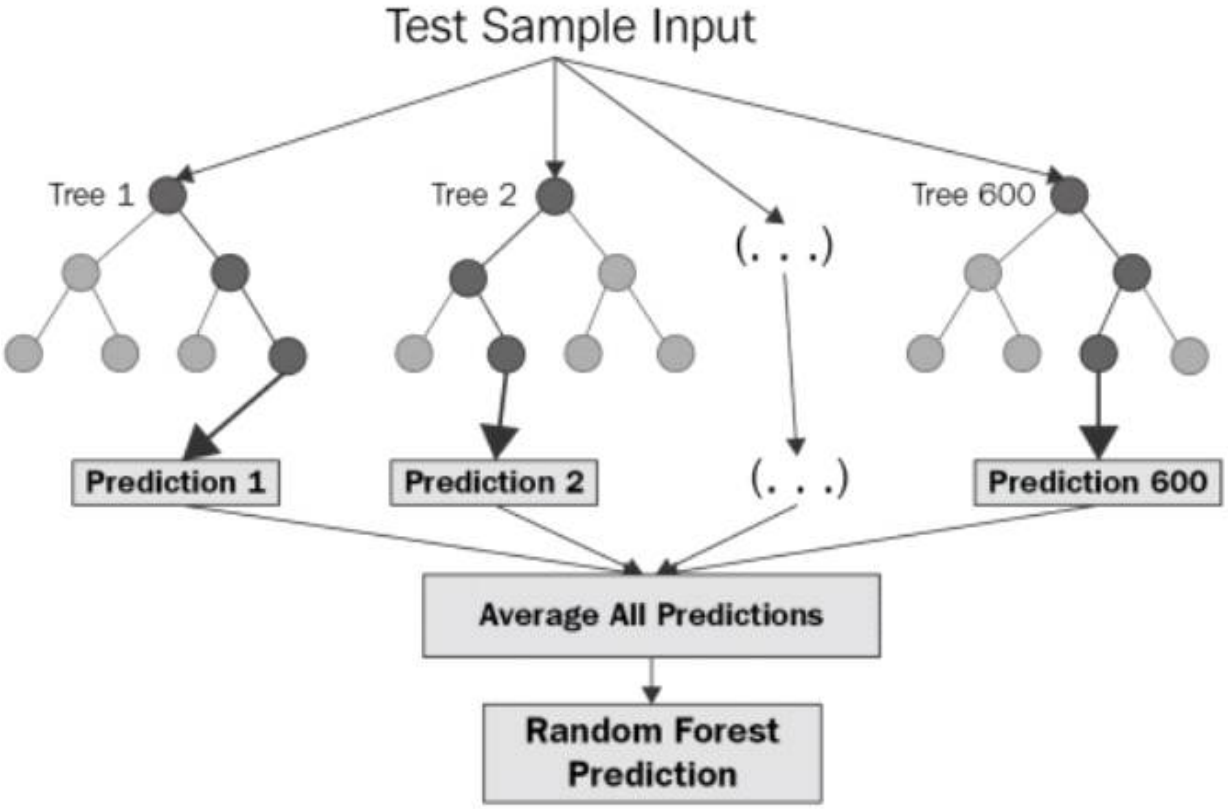


그림1. 랜덤 포레스트는 대표적인 배깅 방식 앙상블 알고리즘으로 여러 개의 트리를 만들어 보팅으로 최종 클래스 값을 결정하는 알고리즘이다. 부스팅 기반의 다양한 앙상블 알고리즘과 마찬가지로 랜덤 포레스트 역시 트리 기반의 알고리즘으로 Decision Tree의 직관적이라는 장점이 있다. 배깅 방식 앙상블 모델인 랜덤 포레스트는 부스트래핑 분할 방식으로 각각의 트리가 각자의 데이터를 샘플링하여 개별적으로 학습을 진행한 후 모든 분류기가 보팅을 통해 예측을 결정한다.

랜덤포레스트 알고리즘으로 모델을 학습하고, 학습된 모델에 새로운 데이터 값이 주어졌을 때 미지의 레이블 값을 예측하는 시스템.
즉, 주어진 입찰가에 대한 낙찰여부를 예측하는 것이다.

데이터 피처: 낙찰자 결정방법, 입찰률, 업종명
데이터 레이블: 낙찰 여부

데이터 레이블인 낙찰 여부에서 낙찰이 된 데이터보다 되지 않은 데이터가 훨씬 많아 데이터 불균형이 심하다. 이런 경우, 모델은 적은 수의 데이터의 분포를 제대로 학습하지 못해 모델 성능이 떨어진다. 따라서, 문제해결을 위해 오버샘플링 기법을 이용했다.

오버샘플링은 임의의 소수 레이블 중심으로 인근 레이블 사이에 새로운 데이터를 생성해 소수 레이블, 다수 레이블 비율을 1:1로 만든다.

오버샘플링된 데이터를 바탕으로 랜덤포레스트 알고리즘으로 학습시켜 모델을 생성한다. 이 모델에 새로운 데이터 값을 주면 낙찰여부를 확인할 수 있다.

- 개발언어 : python 3.10.0
- 사용 라이브러리 목록 : pandas 1.4.1, numpy 1.22.2, joblib 1.1.0, scikit-learn 1.0.1, scipy 1.8.0, django4.0.2, djangorestframework 3.13.1, drf-yasg 1.20.0, threadpoolctl 2.0.0, imblearn, matplotlib 3.2.2
- 확인 가능한 소스코드 파일 : 공사입찰예측모델링.ipynb, result.json

4. 결과 및 기대효과

데이터를 가지고 온 공공조달 홈페이지에는 학습시킨 데이터보다 훨씬 방대한 양의 데이터가 있었으나 정해진 시간 내에는 이 데이터를 모두 가지고 와 학습시키기에는 무리가 있었다.
따라서, 최근 3개월을 기준으로 랜덤한 날짜를 뽑아 학습시켰다. 추후 홈페이지에 주어진 데이터를 모두 학습시킨다면 훨씬 더 정확한 모델을 완성시킬 수 있을 것으로 기대된다.

프로젝트 초기에는 홈페이지를 제작한 후, 콤보 박스를 통해 입찰자가 원하는 항목을 선택하고 그것에 대해 입찰 성공 여부를 알려주는 서비스를 진행하려고 했다. 그러나 시간 문제 상 계획 수정이 불가피해 졌다. 추후 종목마다 데이터를 학습시킨 후 홈페이지를 제작해 API에서 각 항목마다 입찰 가능 여부를 알려준다면 더욱 넓은 범위의 예측 시스템을 구축할 것으로 기대된다.

이 프로젝트에서 만든 모델은 입찰 참여의 진입 장벽을 낮출 수 있다는 기대효과를 가지고 있다. 신생 기업 같은 경우 입찰에 익숙하지 않아 좋은 기술과 충분한 가격 경쟁력을 갖추고 있음에도 불구하고 입찰에 실패할 가능성이 높다.

이번 서비스를 통해 신생 기업들도 입찰에 성공적으로 참여해 각 항목의 질이 상향평준화될 것으로 기대된다. 또한, 기업들이 합리적인 가격을 제시함에 따라 정부에서는 예산을 효율적으로 운용할 수 있고 모든 항목들에서 건강한 입찰 문화가 형성될 수 있다.

이 서비스를 통해 가격의 기준점을 제시해주게 된다면 기업들이 이것에 맞춰 합리적인 가격을 제시하게 될 것이다.

합리적인 가격을 제시받은 정부는 예산을 효율적으로 사용해 더 많은 곳에 남은 예산을 적극적으로 사용할 수 있게 된다.

궁극적으로 건강한 입찰 문화 증진, 필요한 곳에 예산이 더 쓰일 수 있는 기회로 작용할 수 있다.