코드스테이츠

Team Project 1
AI 소프트웨어 업그레이드

2023-07-19 ~ 2023-07-31

18기 - 11팀 아자아자 김우영 김보미 최재용 이강우

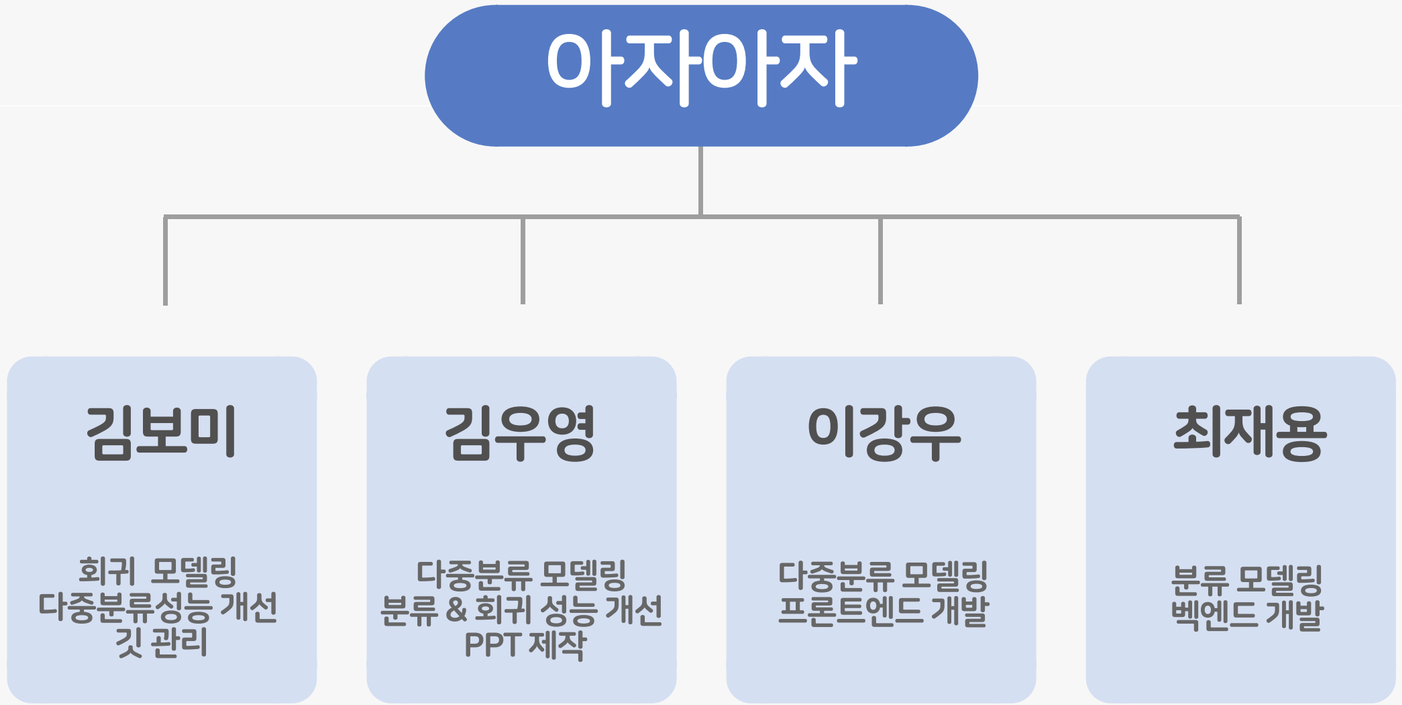
코드스테이츠

목차

Chapter 01
프로젝트 개요 및 구성

프로젝트 수립 배경 및 구성

사전에 구축된 AI Model은 성능이 부족하여 정확한 예측을 제공하지 못하고, 인터페이스를 가지고 있지 않습니다. AI Model의 성능 개선과 사용자 친화적인 웹 인터페이스 구축을 통해, 정확한 예측을 제공하는 높은 성능의 AI 서비스를 제공하며, 사용자들이 쉽게 접근하고 활용할 수 있도록 하여 AI 기술의 보다 넓은 영향력을 발휘합니다.



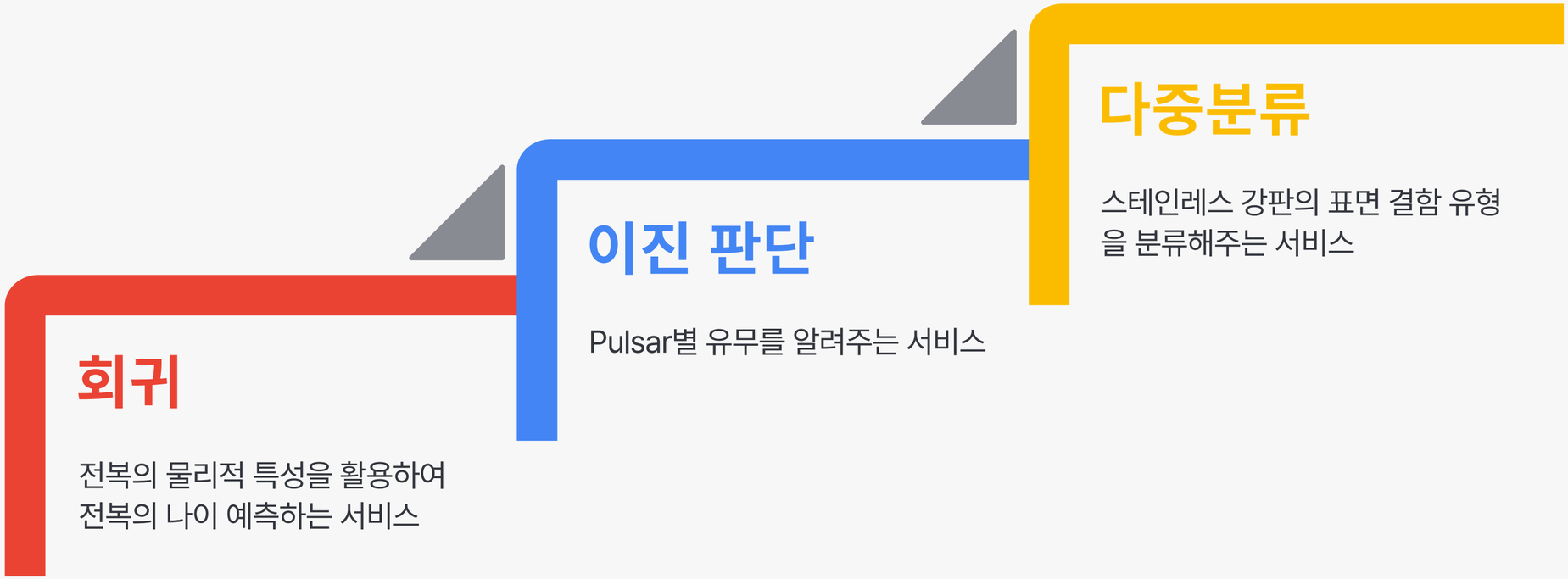
Chapter 02
프로젝트 수행 절차 및 방법

코드스테이츠

프로젝트 수행 절차 및 방법

Chapter 03
프로젝트 수행 결과

코드스테이츠


회귀
전복의 물리적 특성을 활용하여
전복의 나이 예측하는 서비스
이진 판단
Pulsar별 유무를 알려주는 서비스
다중분류
스테인레스 강판의 표면 결함 유형
을 분류해주는 서비스

Chapter 03-1
회귀모델

회귀 프로젝트 목적: 전복의 나이 예측해서 전복 연령을 알려주는 자동화 웹사이트 배포하기

전복 나이를 알면 뭐가 좋을까?
전복은 전 세계의 차가운 연안 해역에서 발견되며 멸종 위기에 처해있습니다.
전복의 나이는 가격과 비례적인 관계가 있어, 전복의 가격을 측정하는 데 중요한 역할을 합니다.
그러나 전복의 나이를 결정하는 것은 매우 복잡한 작업입니다. 우리의 서비스는 기계 학습 모델을 사용하여 전복의 나이를 예측함으로써 이러한 수동적인 프로세스를 효율적으로 가속화하고자 합니다.
이를 통해 보다 정확하고 신속한 전복 연령 예측 서비스를 제공할 수 있습니다.

코드스테이츠

데이터 소개 및 전처리

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7
5	I	0.425	0.300	0.095	0.3515	0.1410	0.0775	0.120	8
6	F	0.530	0.415	0.150	0.7775	0.2370	0.1415	0.330	20
7	F	0.545	0.425	0.125	0.7680	0.2940	0.1495	0.260	16
8	M	0.475	0.370	0.125	0.5095	0.2165	0.1125	0.165	9
9	F	0.550	0.440	0.150	0.8945	0.3145	0.1510	0.320	19

데이터 소개

구성: 4177개의 데이터, 9개의 column

주요 정보: 전복의 성별, 길이 등 물리적인 특성, 전복의 나이

데이터 전처리

Point 1

이상치 제거 : z-score의 절대값이 3보다 크면 이상치로 판단하고 제거

Point 2

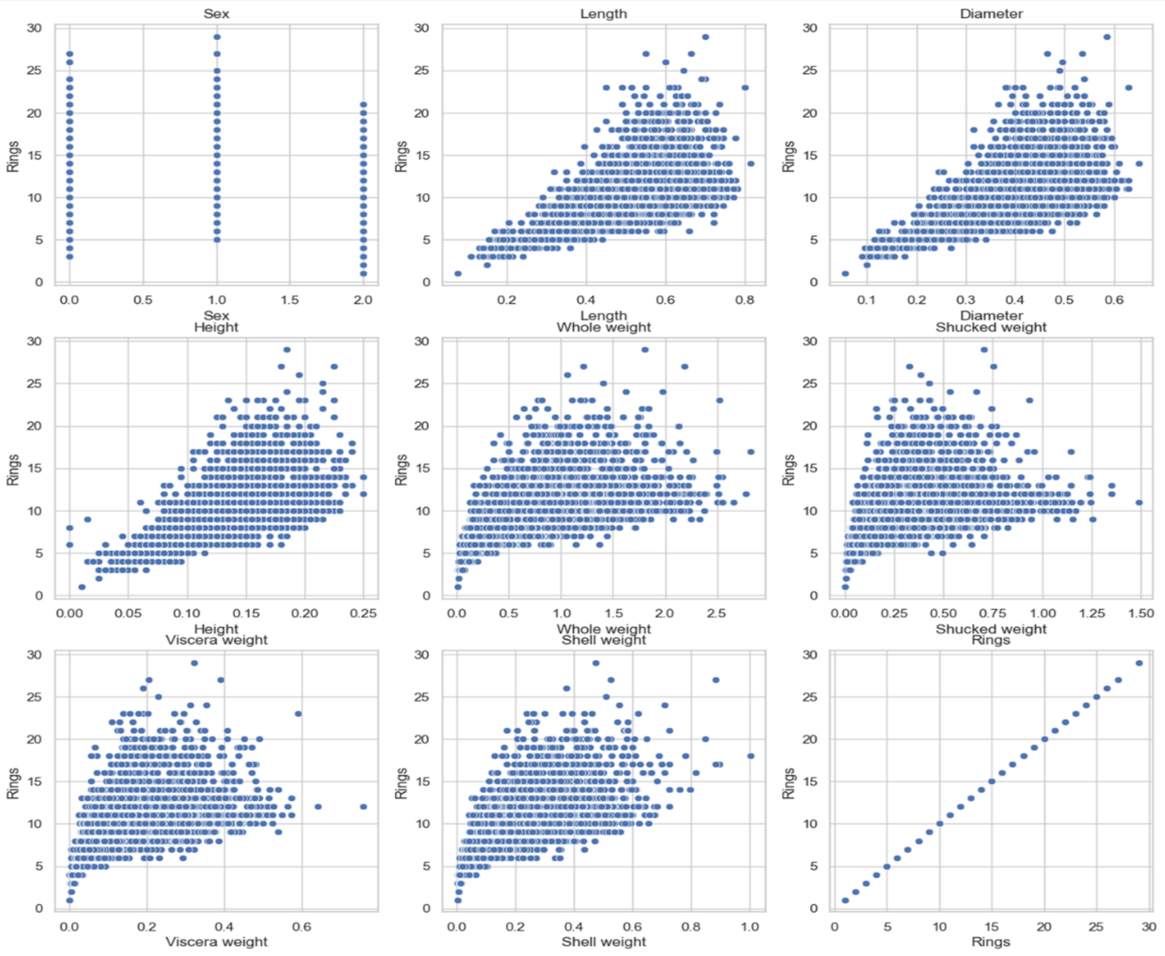
이상치 제거 : "전체 무게 < 조개껍질 벗긴 무게 + 내장 무게 + 껍질 무게" 인 열들 제거

Point 3

범주형인 성별 컬럼 onehotencoding 적용

코드스테이츠

데이터 EDA



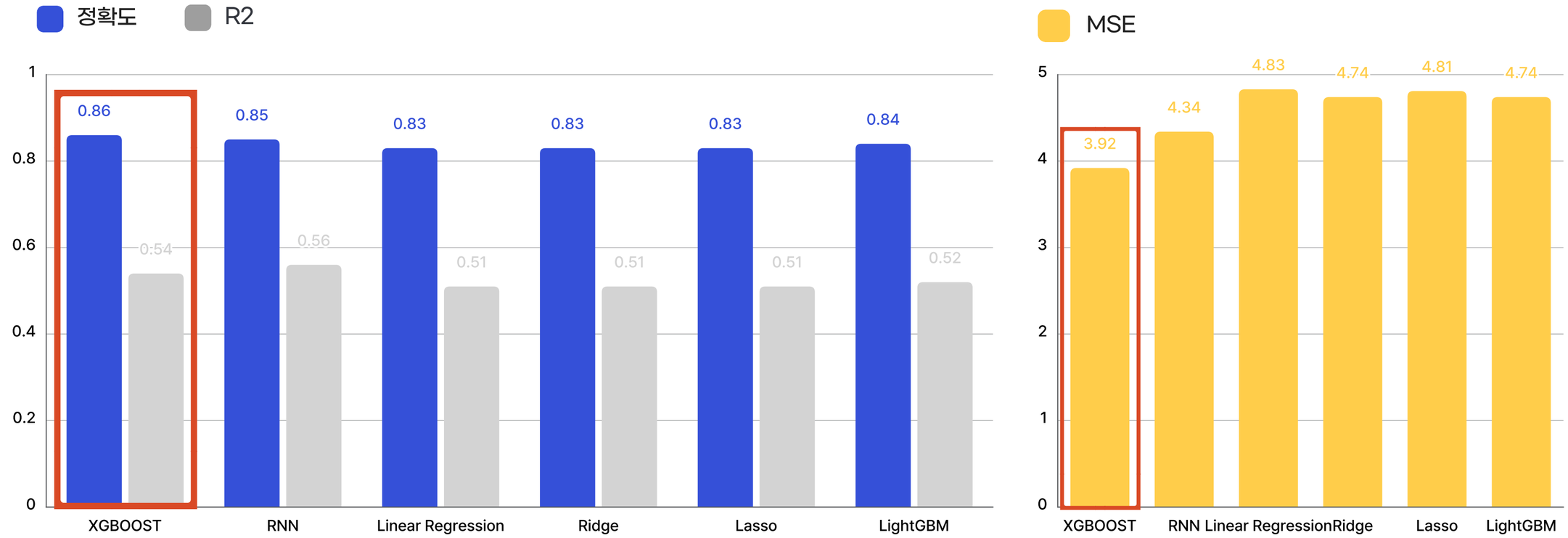
컬럼 간 상관관계

비례 관계: 높이, 껍질 무게, 전체 무게, 지름, 길이가 클수록, 조개 나이가 많아집니다.
반비례 관계: 내장 무게, 껍질 벗긴 무게가 작을수록, 조개 나이가 많아집니다.

⇒ 대부분의 나이가 든 조개일수록 몸통은 커지나, 내장과 같은 안쪽 부분이 작아질 가능성이 있습니다.

코드스테이츠

모델링 및 성능평가



XGBOOST

XGBOOST 는 MSE : 3.92 Accuracy : 86%로 가장 높은 성능을 보이고 Baseline의 Accuracy:82% 보다 높은 성능을 보입니다.

모델 결과 분석

WHY ? XGBOOST

비선형 관계 모델링
1. XGBoost는 비선형 패턴을 학습하는 능력이 뛰어나, 전복의 물리적 특성들과 나이 간에 복잡한 비선형 관계를 고려합니다.

2. Gradient Boosting 알고리즘 기반
* 각 모델이 이전 모델의 오차를 보완하는 방향으로 학습하여 예측 모델의 정확도 향상

3. 앙상블 학습: n_estimators = 400
n_estimators=400
400개의 개별 모델을 결합해 성능 향상

모델 결과 분석

아쉬웠던 점

1. 상관관계 큰 열 제거 효과 없음.
 상관관계가 큰 열들이 많아 그 중 높은 것을 제거해봤지만, 오히려 성능이 더 떨어졌습니다. 상관관계가 큰 열은 중요한 정보를 포함하고 있을수있어, 정보 손실이 발생할 수 있습니다. 그러나, XGBoost는 특성중요도를 계산해 높은 상관관계가 있더라도 적절히 활용할 수 있어, 상관관계 큰 열 제거는 성능 향상에 도움이 되지 않았던 것으로 보입니다.

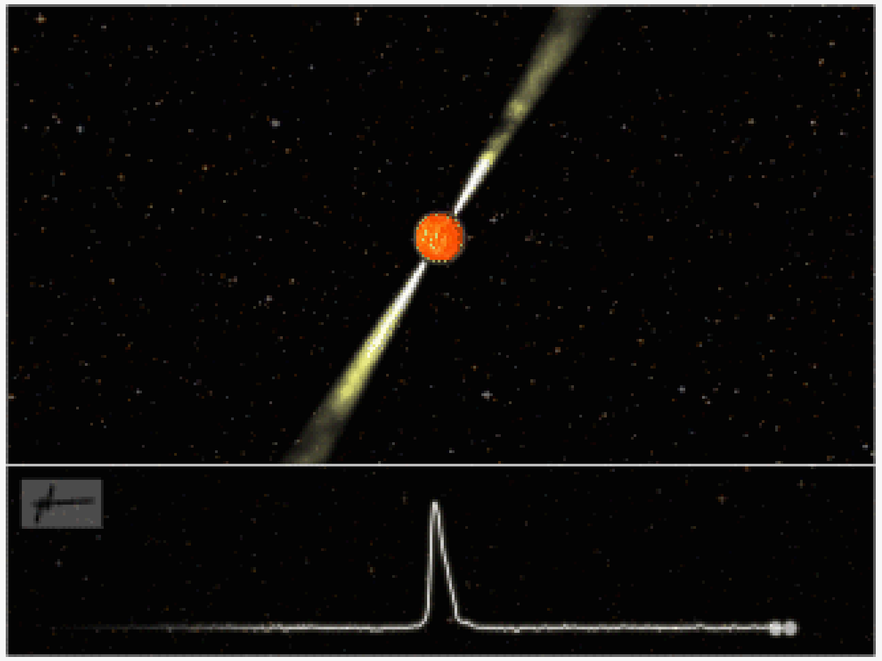
2. PCA효과 없음
 PCA는 주로 선형적인 차원축소 기법이므로, 전복 데이터셋처럼 비선형 데이터에는 적합하지 않을 것으로 보입니다.

3. 모델 성능 비교
 XGboost와 RNN모델이 그 중 성능이 좋았는데, 다른 모델들의 성능도 비교적 유사한 결과를 보였지만 아쉬웠습니다. 다른 모델들에 비해, XGBoost와 RNN이 비교적 더 좋은 성능을 보인 이유는, 전복 데이터셋의 특성과 비선형 관계를 더 잘 처리할 수 있는 모델들이기 때문으로 생각됩니다.

Chapter 03-2
이진 판단

이진 판단 프로젝트 목적: 중성자 별 중에서 Pulsar별 혹은 Not Pulsar별을 분류해주고 Pulsar별일 확률을 나타내주는 웹사이트 배포하기

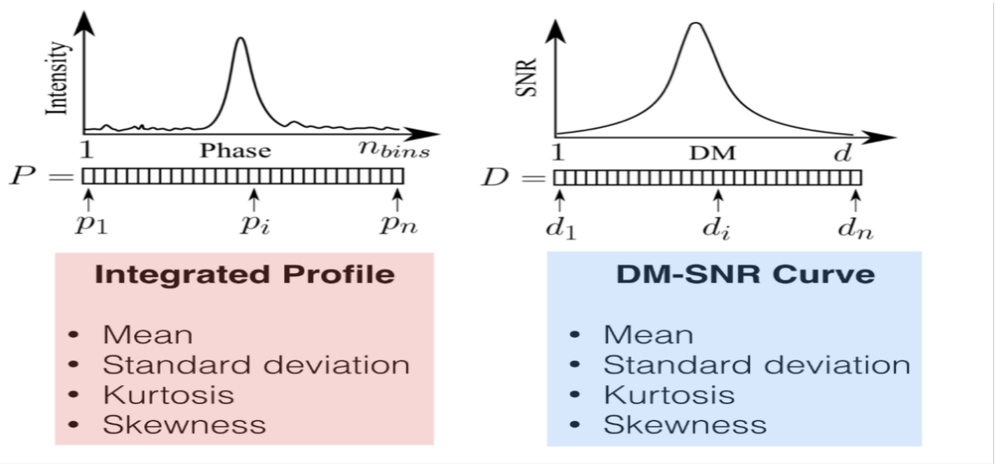
Pulsar Stars란 -지구에서 감지 가능한 라디오 신호를 만드는 특별한 종류의 별입니다.



이 별이 중성자별이 확률은 72%입니다

코드스테이츠

데이터 소개 및 전처리



데이터 소개

구성: 17898개의 데이터, 9개의 column

주요 정보: 통합 펄스 프로필(접힌 프로필),DM-SNR 곡선에서 얻은 간단한 통계, 펄서 여부

https://allofyourbases.com/2019/03/09/classifying-dead-stars/

데이터 전처리

Point 1

수치형 데이터 StandardScaler적용하여 데이터 분포를 조정해 주고 target 컬럼을 레이블 인코딩해주었습니다.

Point 2

이상치 제거: 하위 5%와 상위 95% 값을 이상치로 지정해 주고 삭제해주었다.

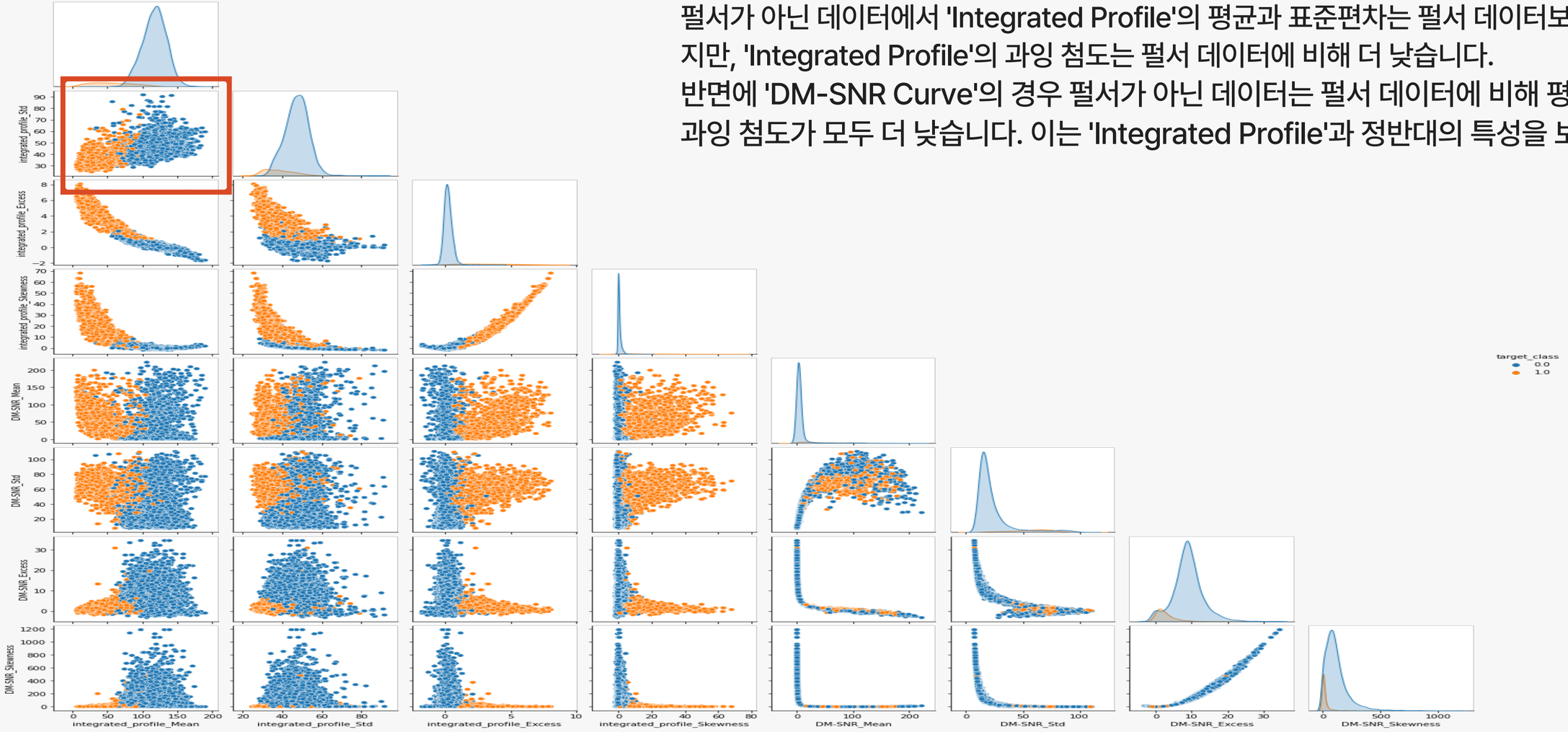
Point 3

SMOTE 로 target 데이터의 불균형 해결

코드스테이츠

데이터 EDA

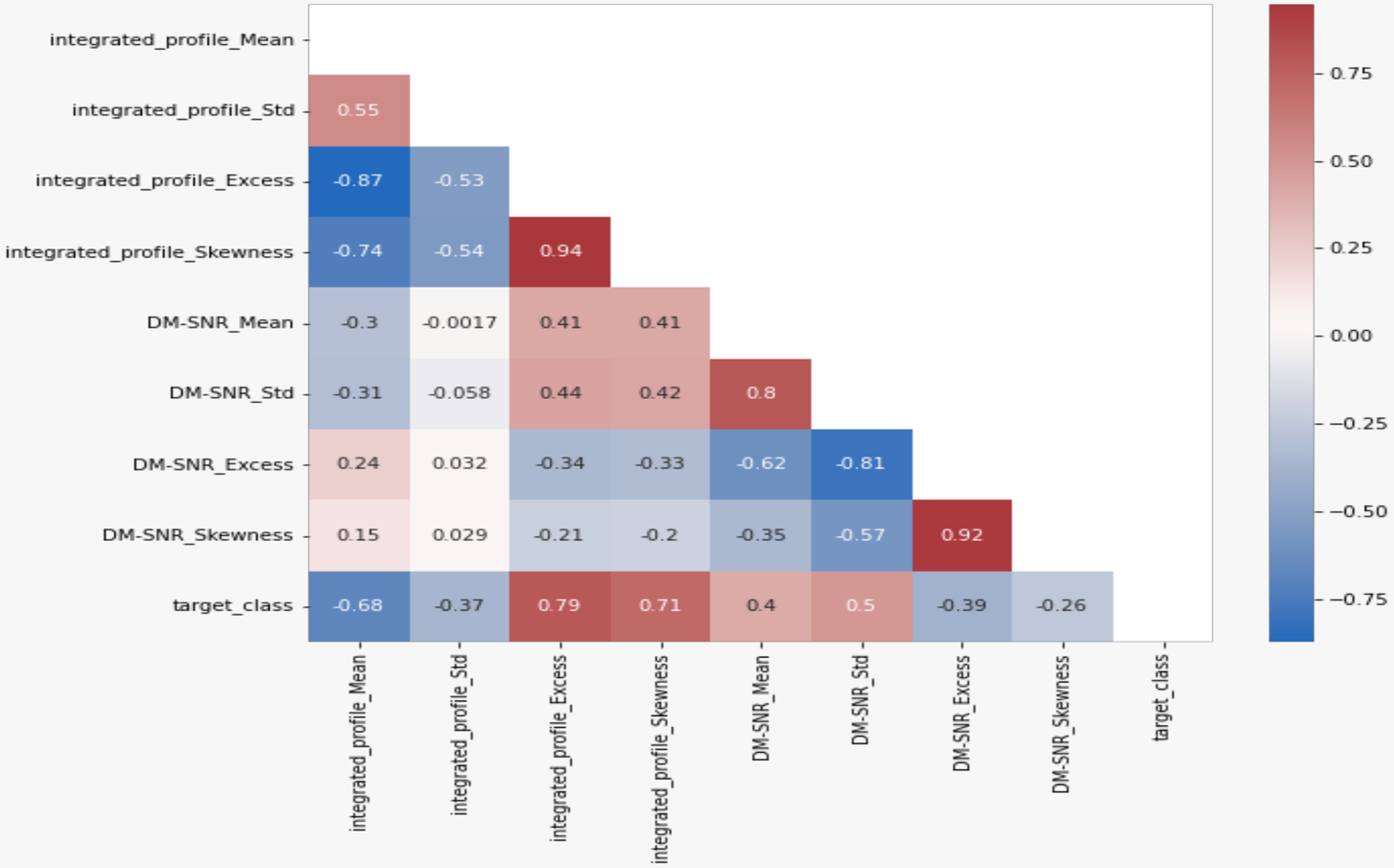
펄서가 아닌 데이터와 펄서 데이터의 차이
펄서가 아닌 데이터에서 'Integrated Profile'의 평균과 표준편차는 펄서 데이터보다 높습니다. 하지만, 'Integrated Profile'의 과잉 첨도는 펄서 데이터에 비해 더 낮습니다.
반면에 'DM-SNR Curve'의 경우 펄서가 아닌 데이터는 펄서 데이터에 비해 평균, 표준편차, 과잉 첨도가 모두 더 낮습니다. 이는 'Integrated Profile'과 정반대의 특성을 보여줍니다.



통합 프로파일 기능의 페어 플롯

코드스테이츠

데이터 EDA



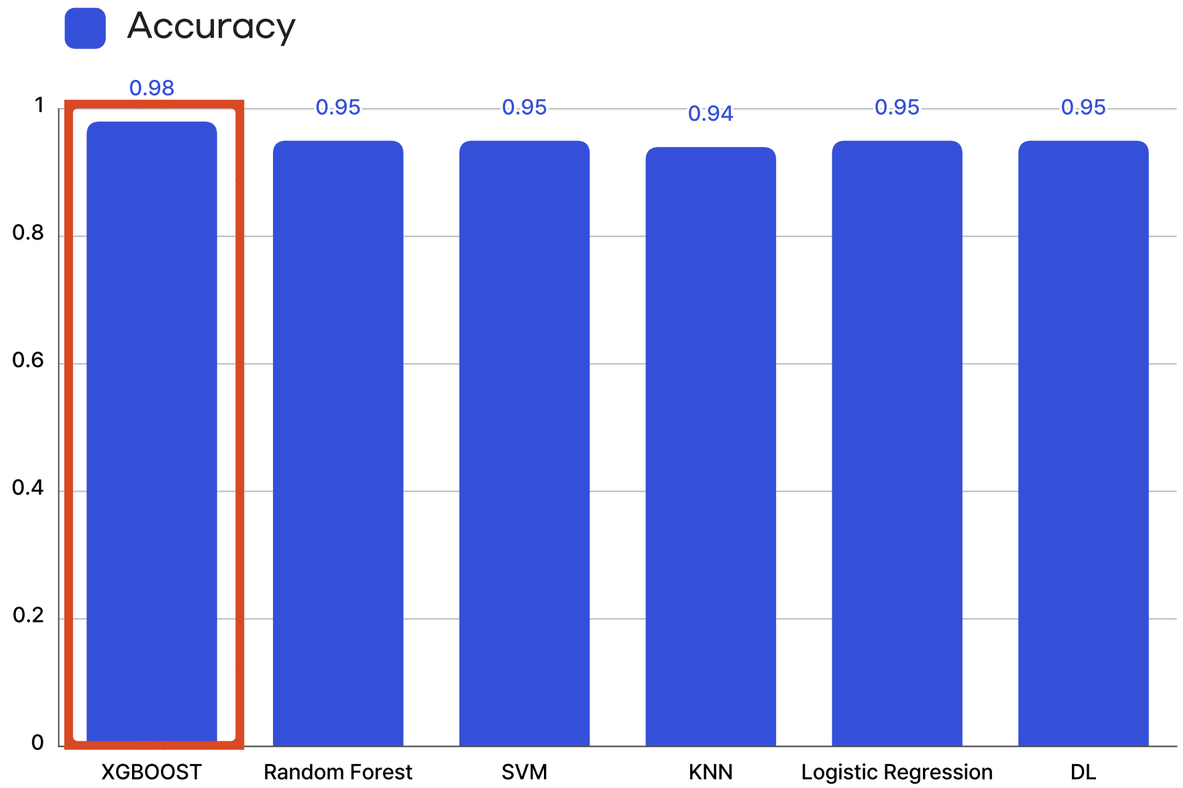
컬럼 간 상관관계

양의 상관관계
-과잉 첨도(EK)와 왜도(Skewness)
-시간-주파수 대기곡선의 표준 편차(SD_DMSNR)와 시간-주파수 대기곡선의 과잉 첨도(EK_DMSNR)

음의 상관관계
-평균 통합 프로파일(Mean_Integrated)와 DM-SNR 곡선의 과잉 첨도(EK_DMSNR_Curve)
-평균 통합 프로파일'과 과잉 첨도(EK)
-평균 통합 프로파일과 왜도(Skewness)

코드스테이츠

모델링 및 성능평가



Metric	precision	recall	f1-score	support
Class 0	0.98	0.99	0.99	3256
Class 1	**0.91**	**0.82**	0.86	324
Accuracy			0.98	3580
Macro Avg	0.95	0.91	0.93	3580
Weighted Avg	0.98	0.98	0.98	3580

XGBoost

Baseline의 Accuracy:85% 보다 모든 모델들이 높은 성능을 보이고 있기 때문에 모델이 향상되었습니다.
XGBOOST가 그 중에서 가장 높은 성능을 가지고 있지만 모든 모델의 성능이 비슷한 것을 알 수 있습니다.

모델 결과 분석

WHY 모델들 성능이 다 높을까?

1. 문제 복잡성: 데이터의 패턴이 비교적 단순하고 직관적이기 때문에, 복잡한 모델 없이도 높은 성능을 얻을 수 있었던 것 같습니다.

2. 데이터 불균형: 데이터의 분포가 9대 1 정도로 심하게 불균형합니다.

3. 적절한 전처리: 데이터 전처리 단계에서 스케일링, 이상치 제거, 특성 선택 등 전처리 과정에서 데이터를 잘 정제했기 때문에 모든 모델들의 성능 향상에 도움을 준 것 같습니다.

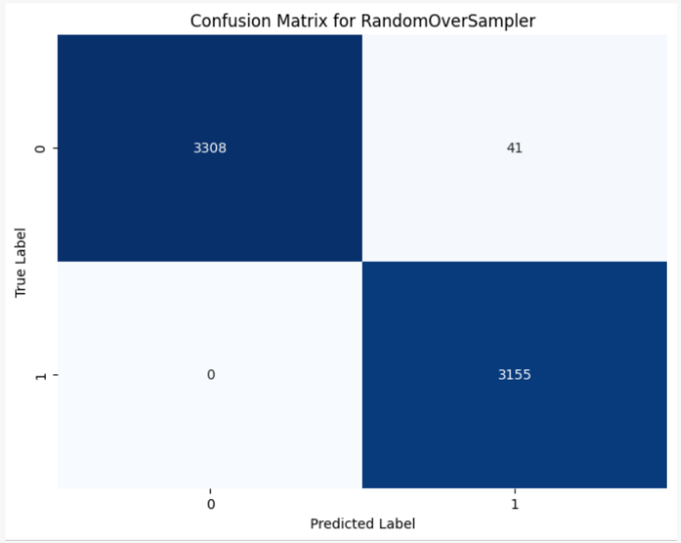
모델 결과 분석

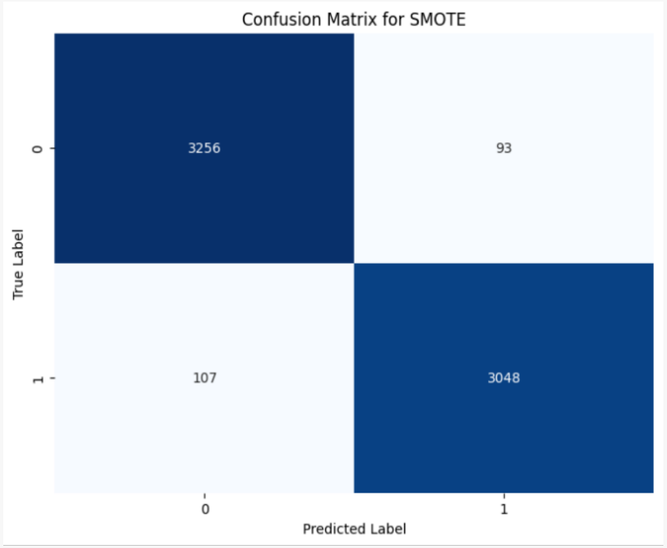
WHY ? XGBOOST

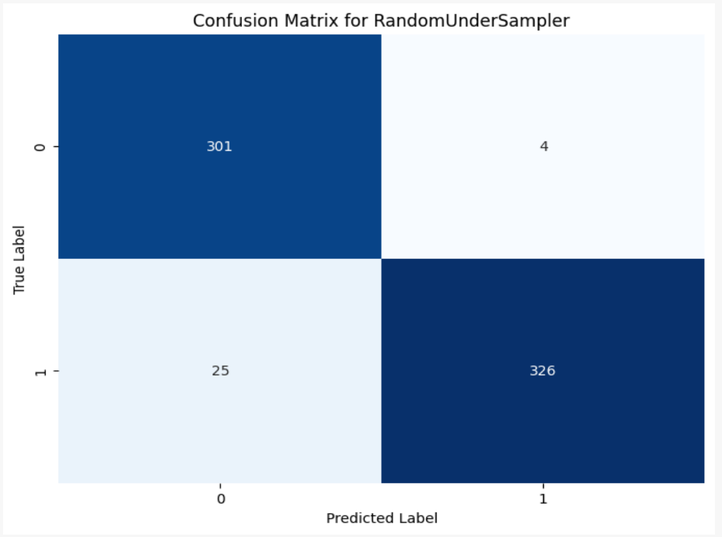
1. Gradient Boosting 기법
이 데이터셋의 특성(통합 프로파일의 평균, 표준편차, 첨도 등)은 서로 상관관계가 있으며, 각 특성이 결정 경계를 형성하는데 기여하므로, 데이터 패턴을 학습할 수 있는 XGBoost가 뛰어난 성능을 보일 수 있습니다.

2. 규제화(Regularization) 특성
XGBoost는 모델의 복잡도를 제어하는 규제화 매개변수를 제공합니다. 이 데이터셋은 여러 특성이 있고, 일부 특성은 노이즈일 수 있으므로, 이러한 노이즈를 제어하고 과적합을 방지하기 위해 규제화가 필요할 수 있습니다.

RamdomOverSample VS SMOT VS RamdomOverSample







Confusion Matrix

RamdomOverSample의 정확도는 0.99, SMOT의 정확도는 0.98, RamdomOverSample 0.95로 RamdomOverSample가 가장 높게 나왔지만 RamdomOverSample는 위 방법은 무작위로 데이터를 복제하기 때문에 데이터 과적합이 발생할 수 있고, 새로운 정보가 부족하고, 노이즈가 증폭 될 수 있는 단점이 존재하기 때문에 SMOT 방식을 선택했습니다.

모델 결과 분석

아쉬웠던 점

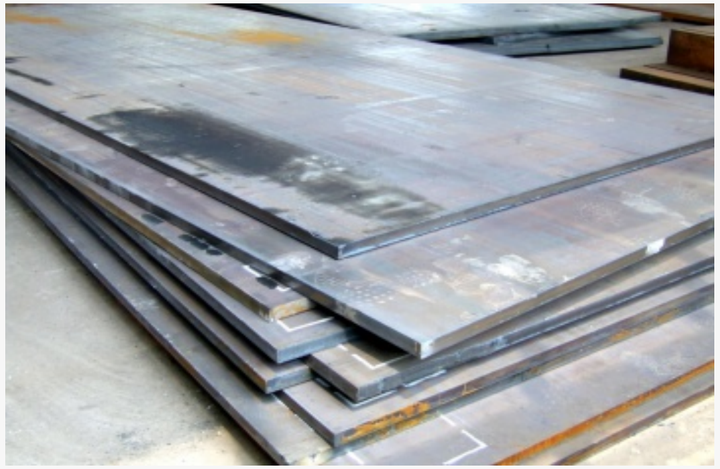
1. 특성 중요도
일반적으로, 중요하지 않은 특성을 제거하면 모델의 성능이 향상될 수 있습니다. 그러나 이 경우, 중요도가 낮은 특성을 제거했을 때보다 그대로 두었을 때 성능이 더 높았습니다. 이는 모든 특성이 분류 문제를 해결하는 데 기여하고 있다고 해석 할 수 있습니다.

2. 전처리의 영향
데이터 전처리는 일반적으로 모델의 성능 향상에 도움이 됩니다. 그러나 이 경우, 이상치 제거나 스케일링 전처리 외에는 크게 성능 향상에 도움이 되지 않습니다.

3. 모델 성능 비교와 데이터 불균형
다양한 모델을 테스트해 보았고, XGBoost가 가장 높은 성능을 보였습니다. 그러나 다른 모델들 또한 높은 성능을 보였고, 심지어 기본 모델도 높은 성능을 보였습니다. 이는 데이터셋의 클래스 불균형이 심해도 모델들이 높은 성능을 나타내는 것으로, 이러한 높은 성능은 클래스 불균형의 영향으로 나타난 것일 수 있다고 생각합니다.

Chapter 03-3
다중분류모델

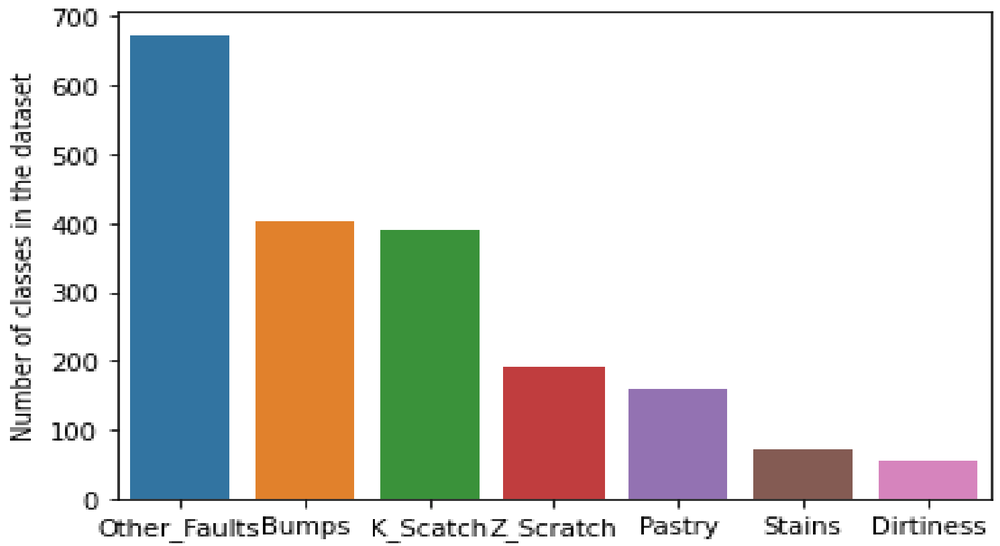
다중분류 데이터 프로젝트 목표: 스테인레스 강판 생산 과정에서 결함문제를 분류 해주고 해당 문제 설명 및 해결방법 해당 문제를 제시해주는 웹



이 강판의 문제점은 Dirtiness로 더러운 자국이 있는 결함입니다
이 문제를 해결하기 위해서 청소 및 유지 관리, 표면 보호가 중요합니다

코드스테이츠

데이터 소개 및 전처리



결함 종류 그래프

데이터 소개

구성: 1941개의 데이터, 34개의 column

주요 정보:결함의 기하학적 모양과 윤곽설명 하는 27개의 지표, 결함들의 종류

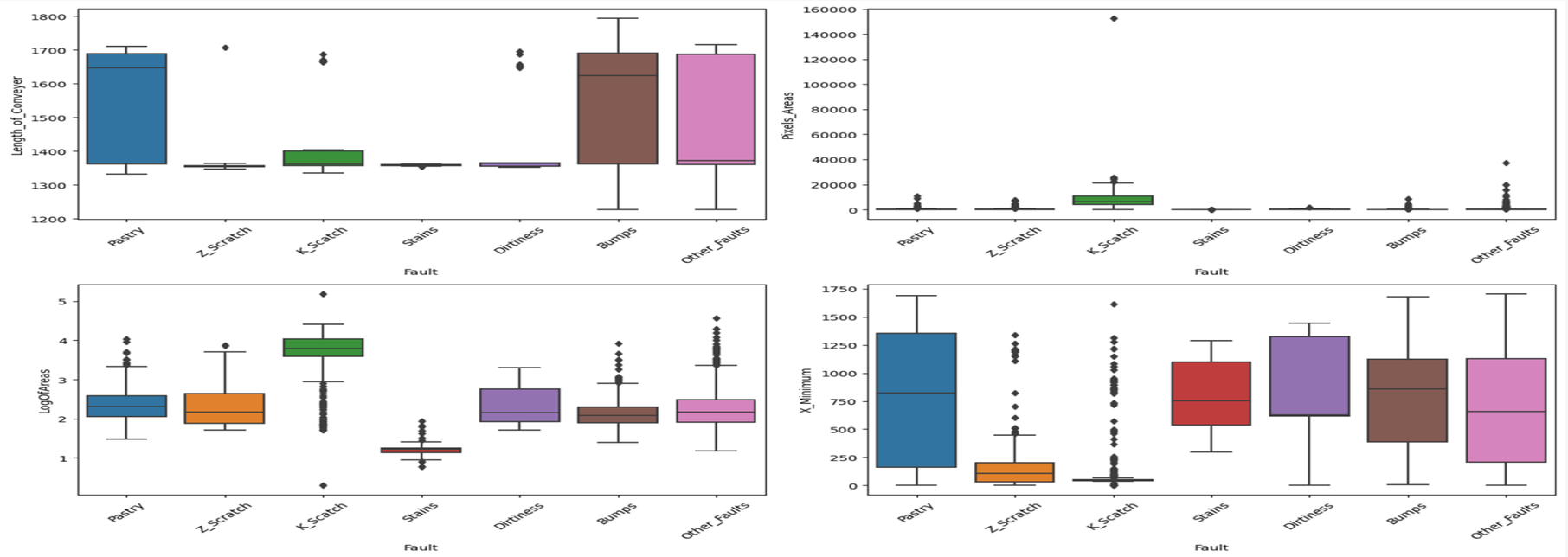
데이터 전처리

Point 1	수치형 ⇒ category 타입 변환 : TypeOfSteel_A300, TypeOfSteel_A400, Outside_Global_Index
Point 2	합칠 수 있는 컬럼 합치기 x_perimeter + y_perimeter = Total_perimeter min, max of Luminosity : mean 값으로 묶기
Point 3	수치형 데이터를 StandardScaler적용하여 데이터 분포의 조정해주고 Tukey's fences를 사용하여 이상치를 제거하였다.
Point 4	SMOTE 로 target 데이터들의 불균형 해결
Point 5	다중공산성 문제를 위해 상관계수 0.95 이상인 열 제거

코드스테이츠

데이터 EDA

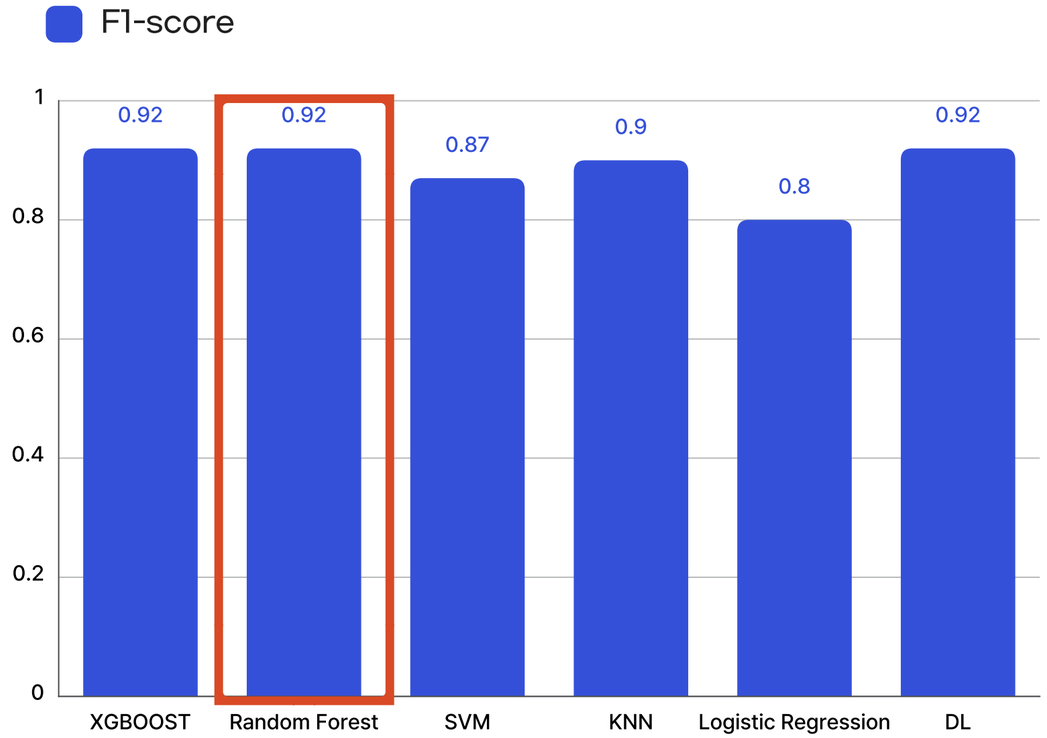
X_Maximun 결함 종류 BOXPLOT



Bumps는 X_MIN이 높은 반면, K_Scatch는 X_MIN이 작은 것을 알 수 있다. 이외에도 많은 속성들이 결함 종류에 따라 다른 특성을 가지고 있음을 알 수 있습니다.

코드스테이츠

모델링 및 성능평가



RandomForestClassifier:

	precision	recall	f1-score	support
0	0.79	0.86	0.83	124
1	0.99	0.99	0.99	144
2	0.99	0.98	0.98	126
3	0.84	0.69	0.76	134
4	0.88	0.94	0.91	139
5	0.99	1.00	0.99	134
6	0.97	0.99	0.98	142
accuracy			0.92	943
macro avg	0.92	0.92	0.92	943
weighted avg	0.92	0.92	0.92	943

Accuracy: 92.4%

Random Forest

Random Forest는 Accuracy 92%로 가장 높은 성능을 보인다. 이는 Baseline의 Accuracy:41% 보다 훨씬 향상된 성능을 보인다.
이 외에도 다른 모델들 또한 Baseline보다 훨씬 높은 성능을 보이는 것으로 보아 전처리가 중요한 데이터이라고 해석할 수 있을 것 같습니다.

모델 결과 분석

WHY ? Random Forest

1.특징 간 상호작용 처리
특징 간의 상호작용을 자연스럽게 처리할 수 있습니다.

2. 다중 분류에 강함
각 트리를 독립적으로 투표해 가장 많이 투표되는 클래스를 선택하는 메커니즘을 사용하기 때문에 다중 분류에 좋은 성능을 낼 수 있었습니다.

3. 비선형 및 비정규 데이터를 다룰 수 있음
Random Forest는 비선형 및 비정규 데이터를 다룰 수 있는 강력한 모델입니다.

모델 결과 분석

WHY Regression is Low?

1. 이진 분류에 최적화
이진 분류 문제를 다루도록 설계되어 있습니다. 다만, 다중 클래스를 분류하도록 원대모든 방식과 같은 방법을 사용할 수 있지만 그럴 필요가 있을 정도의 성능을 보장하지 않습니다.

2. 특징 간 상호작용 무시
변수 간 상호작용을 고려하지 않습니다.

3. 선형성 가정
독립 변수와 종속 변수 간 선형 관계가 있다고 가정합니다. 이 가정은 데이터가 복잡하고 비선형 관계를 가지고 있다면 잘 학습할 수 없고 성능 또한 낮을 수 있습니다.

모델 결과 분석

아쉬웠던 점

1. 모델 다양성 확보
Gradient Boosting, Neural Networks, Naive Bayes, Decision Trees 등 더 다양한 모델을 시도해 보고 문제를 더 깊게 이해할 수 있는 시간이 부족했습니다.

2. 데이터 전처리
여러 모델을 학습하고 비교하는 시간과 전처리를 함께 진행하다 보니 놓친 부분도 있는 것 같고, 전처리 전과 후의 차이를 여러 방법으로 비교해 볼 수 있는 시간을 가지지 못했습니다.

Chapter 04
웹 시현

Chapter 05
자체 평가 의견

아자아자

"AI모델 성능 1위 달성"

김보미

팀원들과 역할을 분담하여 협력하여 작업해서 일정 관리와 작업 흐름에 있어 효율적으로 잘 해낸 것 같습니다.또한 모든 팀원들과의 의사소통이 적극적으로 되어 프로젝트의 목표와 방향성에 대한 공통 이해를 도모하고, 개선을 위한 피드백을 자유롭게 주고받을 수 있었습니다. 그래서 프로젝트의 완성도 또한 높았습니다.
부트캠프에서의 첫 팀프로젝트가 성공적으로 끝나서, 향후 다른 프로젝트에서도 팀원들과의 성장과 발전에 좋은 영향을 미칠 것 같습니다.

김우영

프로젝트를 팀으로 진행한적이 처음이라 처음에는 걱정이 많았었는데 역할을 확실히 정하고 서로 피드백을 거듭하면서 서로 몰랐던 부분을 알게 되었고 배울 수 있어서 유익한 시간이었습니다.
팀에게 민폐가 되지 않기 위해 앞으로 더 많이 공부해야한다는 것을 깨달았고 혼자 했으면 절대 이정도 퀄리티가 되지 못했을텐데 함께 하니 훨씬 좋은 퀄리티의 프로젝트를 완료할 수 있었던 것 같아 뿌듯합니다.

이강우

팀 프로젝트를 처음 시작하기 전에 개인적으로 세운 목표인 "프로젝트 완성"을 이룬 것 같아서 기분이 좋습니다. 첫 팀 프로젝트에서 업무 분담을 하면서 해결된 문제들에서 어떠한 개선점을 개인적으로 가져야 할지 취업 전 알게 된 것이 좋았습니다. 팀원들의 데이터 전처리와, 모델 튜닝, 예측 결과를 분석하는 과정을 밀접하게 소통하면서 더 배우고 익혔습니다. 프로젝트의 모든 과정에서 내가 부족했던 부분을 팀원이 채워 주고, 프로젝트 곳곳에 팀원들의 노력이 들어갔으나 나 스스로 채워 넣은 것이 부족한 것 같아 좀더 노력해야 하는 부분이 어느 것인지 알게 되었습니다.

최재용

처음에는 머신러닝과 딥러닝 flask html등 모두가 기억이 잘 나지 않았는데 다시 하면서 복습할 수 있었던 중요한 시간이였던 것 같습니다. 그리고 매번 혼자 했던 프로젝트와는 달리 다같이 서로의 부족한 점을 보완해주며 호흡을 맞추고 또 서로에게 배우면서 더욱 더 질 높은 프로젝트를 만드는 것 또한 중요한 경험이 되었습니다.

감사합니다