선박 닻 끌림 예측

권병근[1], 신호섭[2], 박민서[3], 이다인[2], 성다솜[2,*]

[1]부산대학교 수학과, 산업 수학 소프트웨어 연계전공, 산업수학센터 학부 연구생
[2]부산대학교 수학과, 빅데이터 연계전공, 산업수학센터 학부 연구생
[3]부산대학교 수학과, 빅데이터 연계전공
이메일 : *som0608@pusan.ac.kr

1. Abstract

이 프로젝트는 기상청과 환경부에서 주최하는 2023 날씨 빅데이터 콘테스트에 참석하기 위해 진행하였다. 최종적으로 본선 진출 후 "입선"하였다.

선박의 닻이 끌렸을 때의 문제점에 대해서 파악하고, 이를 해결하기 위해 문제를 정의하고 목표를 설정하여 프로젝트를 진행하였다. 주요 문제를 해양 사고의 원인이 되는 닻 끌림 발생을 사전에 예측하는 모델을 개발하는 것으로 정의하였고, 이를 해결하기 위해 데이터 전처리 이후에 주요 모델을 선정하고, 파생 변수들을 생성한 뒤, 데이터 불균형 문제를 해소해서 효율적인 이진 분류 모델을 개발하는 것을 목적으로 설정하였다.

2. Introduction

닻 끌림이란 해저의 닻이 끌리면서 선박의 위치가 고정되지 않고 이동하는 현상으로 배와 닻 사이의 파주력이 최대가 된 시점에서 더 큰 외력이 작용하면 닻이 끌린다. 닻 끌림이 일어나 선박사고가 발생하면 인명 피해 또는 기름 유출이나 파괴된 선박의 잔해 등으로 인한 환경 오염 등이 유발되므로 조기 탐지와 신속한 초동 조치가 필요하다.

해양 사고의 원인이 되는 닻 끌림 발생을 사전에 예측하는 모델을 개발하는 것이 목적이다. 이를 해결하기 위해서 닻 끌림 예측에 필요한 유의미한 변수를 도출하고 데이터 불균형 문제를 해소해서, 닻 끌림 발생 여부에 따라 발생은 1, 미발생은 0으로 판단하는 효율적인 이진 분류 모델을 개발했다.

3. Main

(1) 사용한 데이터

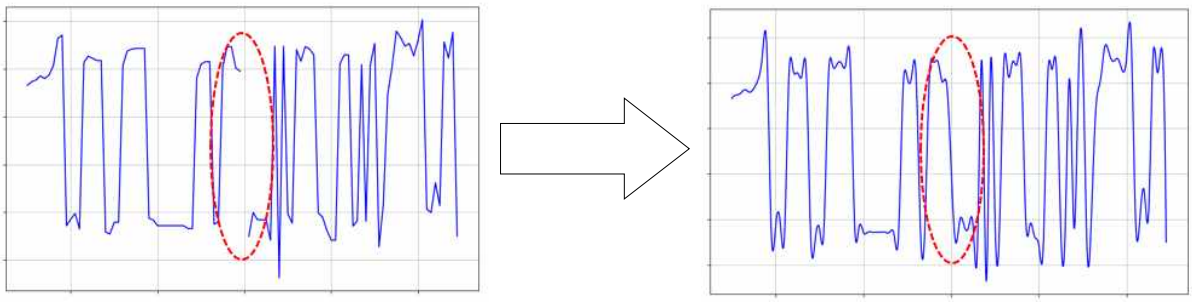
울산과 부산 각각의 정박상태, 닻 끌림 발생 상태, 닻 끌림 발생 시점들의 데이터를 활용하였다.

(2) 전처리 1 – 데이터 통합, 결측치 처리

각각의 데이터들의 column 들이 복잡하고 통일되지 않았던 것들을 간략하고 통일되게 rename 하였다. 실수형 데이터가 문자열 타입으로 되었던 경우에 실수형 타입으로 변경하였다. 이진 분류를 위한 Labeling을 위하여 "anchor_drag" 칼럼을 생성하였고, 닻 끌림 발생 시점 데이터를 활용하여 발생 시점은 0, 발생 시점 이후는 모두 1로 채웠다.

이후에 울산과 부산 각각의 정박 및 닻 끌림 데이터들을 병합하면서 선박들의 이름이 중복되어 혼동되는 문제를 해결하기 위해 울산과 부산 정박상태 데이터에는 각각 1000, 2000을 더하고 닻 끌림 발생 데이터에는 3000, 4000을 더하여 구별하여 주었다.

선박의 상태 중에 결측치가 있어서 처리해주었다. 분석 중인 데이터는 자연현상에서 발생하는 선박들의 움직임을 수치화한 것으로 이는 연속적이며 미분 가능한 것이라고 가정하였다. 따라서 결측치를 2차 보간을 사용하여 처리하였다. 선형 보간과 spine 보간도 사용해 보았다. 결측치가 아니었던 부분에 임의로 결측치를 생성하고 선형 보간, 2차 보간, spine 보간을 사용하였을 때, 2차 보간이 가장 높은 정확도를 보여주어서 2차 보간을 선택하였다.



(3) 모델 선정

전처리 1의 결과로 나온 데이터를 가지고 다양한 머신러닝과 딥러닝을 진행해 보았고 그중에서 앙상블 모델들이 좋은 성능을 보여주었다.

	Accuracy	F1-Score	CSI-Score
Random Forest	0.9941	0.8458	0.7329
XGBoost	0.9944	0.8591	0.7531
LightGBM	0.9908	0.7617	0.1652
CatBoost	0.9914	0.7710	0.6273

Accuracy : $\frac{TP+TN}{TP+TN+FP+FN}$

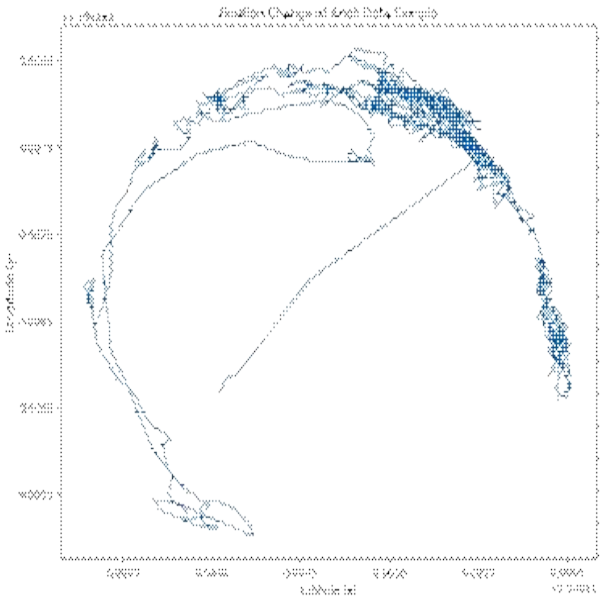
F1-Score : $2 \times \frac{\frac{TP}{TP+FP} \times \frac{TP}{TP+FN}}{\frac{TP}{TP+FP} + \frac{TP}{TP+FN}}$

CSI-Score : $\frac{TP}{TP+FP+FN}$

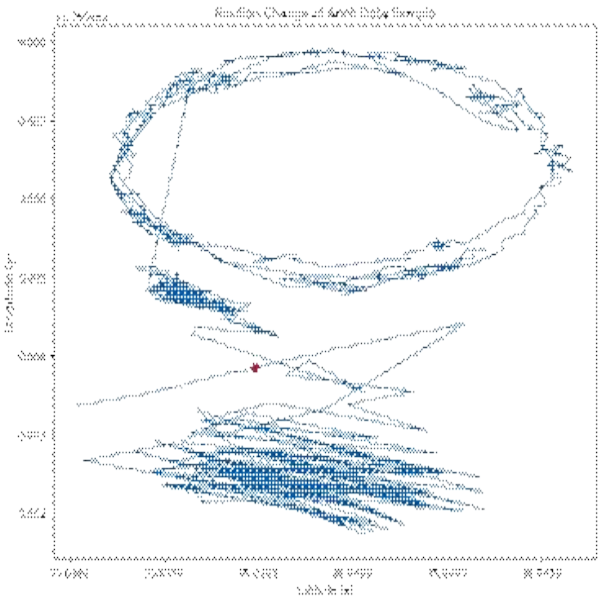
		Prediction	
		발생	미발생
Real	발생	TP	FN
	미발생	FP	TN

각 지표를 비교해본 결과 가장 성능이 좋았던 XGBoost(Extreme Gradient Boosting) 모델을 선정하였다.

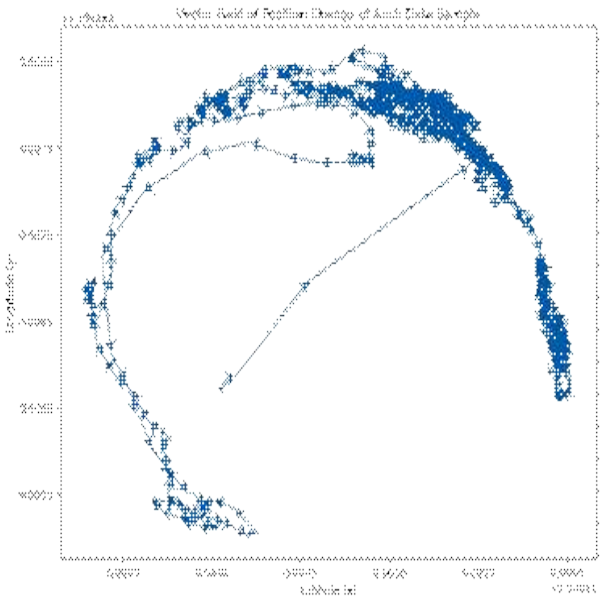
(4) 전처리 2 – 파생 변수 생성



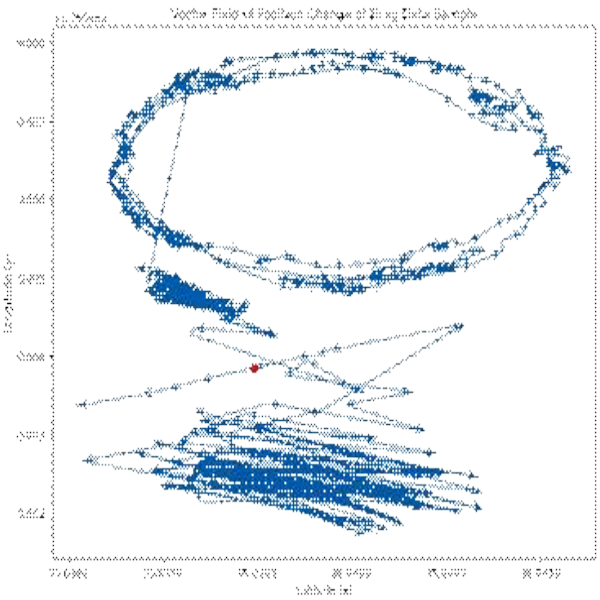
정박상태 데이터 시각화



닻 끌림 발생 데이터 시각화



정박상태 데이터 벡터화



닻 끌림 발생 데이터 벡터화

정박상태와 닻 끌림 발생 데이터를 각각 시각화해보았다. 이는 배의 움직임으로써 움직임을 표현하기 위해서 화살표로 진행 방향이 나타나게 시각화해보았다. 화살표로 진행 방향을 알게 되니, 벡터들이 진행하는 방향들을 알게 되었고, 정박상태와 닻 끌림 발생 데이터 간의 행동 패턴이 다르다는 사실을 확인할 수 있었다. 배가 움직이는 것을 각각의 벡터로 보면 닻이 끌리는 시점에서 벡터의 변화율 관련 변수들이 급격하게 변화가 일어날 것으로 예측하였다. 따라서 다음과 같이 파생 변수들을 계산하여 생성하였다.

d_lati, d_long은 위도, 경도의 변화율로 1차 차분이다. 이는 배가 얼마나 움직이는지 표현하고, 배의 속도를 의미한다.

d_d_lati, d_d_long은 직전에 구했던 변화율의 변화율로 2차 차분이다. 이는 배의 움직임이 얼마나 크게 변화하는지 표현하고, 배의 가속도를 의미한다.

1차 차분과 2차 차분은 Euclidean Coordinate에서 구한 것으로, 실제 지구는 Spherical Coordinate에서 위도와 경도를 설정하므로 실제 거리에서 오차가 발생한다. 따라서 Haversine 함수를 사용하여 실제 거리를 계산한 뒤, 속력과 가속력을 구해 d_km/s, dd_km/s 파생 변수를 생성하였다.

- Haversine Formula

$$\theta = \frac{d}{r}$$

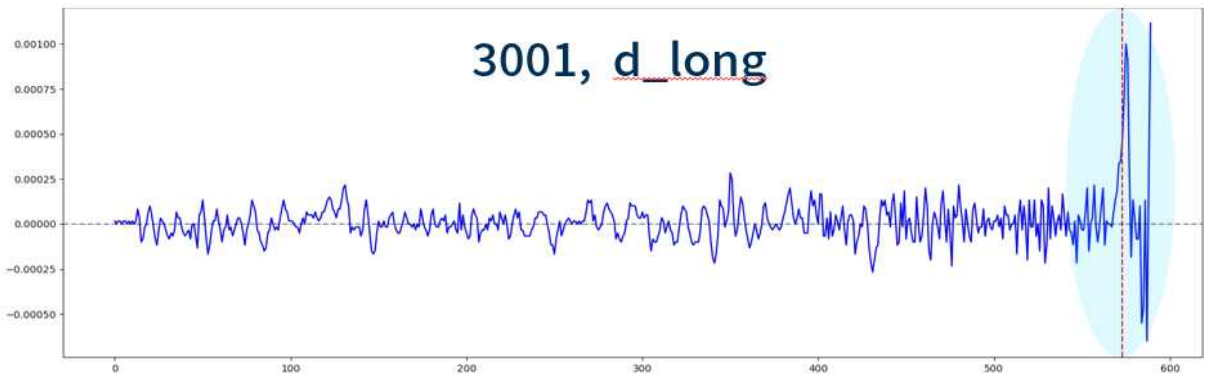
θ : 구면 상의 두 점을 잇는 호의 중심각
d : 두 점 사이의 대원 거리
r : 구의 반지름

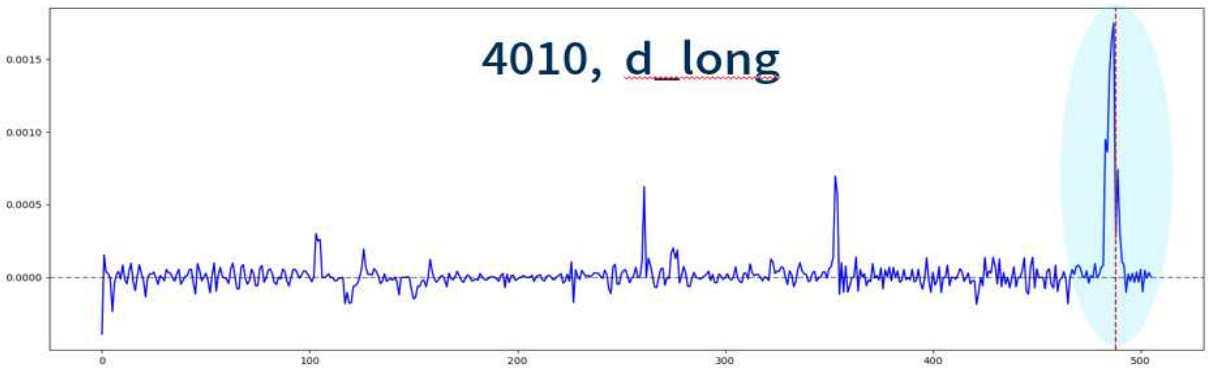
$$hav(\theta) = hav(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)hav(\lambda_2 - \lambda_1)$$

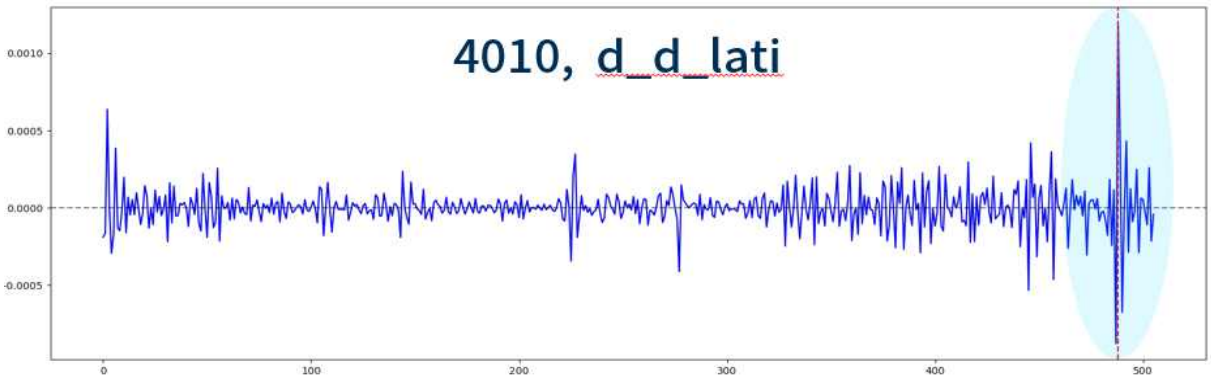
φ_1, φ_2 : 1지점과 2지점의 위도(단위 : $Radian$)
λ_1, λ_2 : 1지점과 2지점의 경도(단위 : $Radian$)

$$hav(\theta) = \sin^2(\frac{\theta}{2}) = \frac{1-\cos(\theta)}{2}$$

$$\begin{aligned} d &= r\,achhav(h) = 2r\arcsin(\sqrt{h}) \\ &= 2r\arcsin(\sqrt{hav(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)hav(\lambda_2 - \lambda_1)}) \\ &= 2r\arcsin(\sqrt{\sin^2(\frac{\varphi_2 - \varphi_1}{2}) + \cos(\varphi_1)\cos(\varphi_2)\sin^2(\frac{\lambda_2 - \lambda_1}{2})}) \end{aligned}$$







위의 plot은 파생 변수를 시각화한 것이다. x축은 시간, y축은 파생 변수들의 값, 파란 선은 시간에 따른 파생 변수의 값, 빨간 점선은 닻 끌림 인지 시점이다. 파생 변수를 생성하기 전에 예측하였던 것처럼 변화율 관련 파생 변수들이 급격하게 변화하는 시점과 닻 끌림 인지 시점이 거의 일치한다.

닻 끌림 발생 이후에 계속해서 닻 끌림이 발생하는 때도 있었지만, 다시 정박상태의 형태를 보여주는 때도 있어 이는 모델의 성능을 저하할 수 있다고 판단하여 닻 끌림 인지 시점 이후의 데이터는 삭제하였다.

- 전처리 이후의 데이터 테이블, 칼럼 설명

	num	latitude	longitude	sog	cog	hdg	time_gap	d_km/s	dd_km/s	d_lati	d_long	d_d_lati	d_d_long	anchor_drag
0	1001	35.461472	129.439878	0.4	9.6	273.0	182.0	0.045969	-0.000152	0.000312	0.000333	0.000342	0.000342	0
1	1001	35.461688	129.439792	0.2	328.0	267.0	175.0	0.025250	-0.000118	0.000216	-0.000086	-0.000096	-0.000096	0
2	1001	35.461708	129.439733	0.2	260.1	265.0	184.0	0.005788	-0.000106	0.000020	-0.000059	-0.000196	-0.000196	0
3	1001	35.461720	129.439708	0.1	262.8	270.0	180.0	0.002628	-0.000018	0.000012	-0.000025	-0.000008	-0.000008	0
4	1001	35.461733	129.439717	0.3	277.6	273.0	181.0	0.001660	-0.000005	0.000013	0.000009	0.000001	0.000001	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
429768	4087	35.055300	129.063917	0.1	272.0	146.0	360.0	0.013654	0.000025	0.000000	-0.000150	0.000033	0.000033	0
429769	4087	35.055250	129.063900	0.0	223.0	142.0	360.0	0.005771	-0.000022	-0.000050	-0.000017	-0.000050	-0.000050	0
429770	4087	35.055183	129.063867	0.1	222.0	138.0	120.0	0.008033	0.000019	-0.000067	-0.000033	-0.000017	-0.000017	0
429761	4087	35.055417	129.064433	0.1	251.0	172.0	540.0	0.021613	0.000008	-0.000033	-0.000234	-0.000066	-0.000066	0
430024	4087	35.054683	129.066017	0.0	139.0	263.0	540.0	0.015091	0.000011	0.000133	-0.000033	0.000050	0.000050	0

1096892 rows × 14 columns

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1096892 entries, 0 to 430024
Data columns (total 14 columns):
 #   Column       Non-Null Count    Dtype
---  ------       --------------    -----
 0   num          1096892 non-null  int64
 1   latitude     1096892 non-null  float64
 2   longitude    1096892 non-null  float64
 3   sog          1096892 non-null  float64
 4   cog          1096892 non-null  float64
 5   hdg          1096892 non-null  float64
 6   time_gap     1096892 non-null  float64
 7   d_km/s       1096892 non-null  float64
 8   dd_km/s      1096892 non-null  float64
 9   d_lati       1096892 non-null  float64
 10  d_long       1096892 non-null  float64
 11  d_d_lati     1096892 non-null  float64
 12  d_d_long     1096892 non-null  float64
 13  anchor_drag  1096892 non-null  int64
dtypes: float64(12), int64(2)
memory usage: 125.5 MB
```

테이블 명	내용	테이블 명	내용
num	배 번호	d_lati	위도의 변화율
latitude	위도	d_long	경도의 변화율
longitude	경도	d_d_lati	d_lati의 변화율
sog	대지속력	d_d_long	d_long의 변화율
cog	실침로	d_km/s	거리의 변화율
hdg	선수미선	dd_km/s	d_km/s의 변화율
time_gap	이전 위치와의 시간 차	anchor_drag	닻끌림 여부(답안)

(5) 데이터 불균형 해소

학습용 데이터가 68,2476행 중, anchor_drag(닻 끌림 시점) 칼럼이 1인 행이 250개로 데이터 불균형 문제가 있다. 이 문제를 해결하기 위하여 Oversampling을 진행하였다. Oversampling 이전의 CSI 점수와 전체 데이터 Oversampling 이후의 CSI 점수를 2023날씨 빅데이터 콘테스트 홈페이지에서 검증하며 CSI 점수를 비교하였다.

Oversampling 유무	기법	CSI-Score
X		0.022
O	Random Oversampling	0.068
	ADASYN Oversampling	0.091
	SMOTE Oversampling	0.083

수치를 확인해보면 Oversampling 이후 CSI 점수가 약 3배에서 4배 정도 좋아진 사실을 알 수 있었다. 하지만 전체 데이터를 Oversampling 하면 Oversampling 비율이 5:5로 모델의 성능을 저하할 가능성이 있다고 판단하였다. 따라서 전체 데이터가 아닌 최초에 anchor_drag가 1이었던 닻 끌림 발생 데이터들만 Oversampling 하여 비율을 6:4로 낮춰서 모델에 학습을 진행해 보았다.

	CSI-Score	
	전체 데이터 Oversampling(5:5)	닻 끌림 발생 데이터 Oversampling(6:4)
Random Oversampling	0.068	0.109
ADA SYN Oversampling	0.091	0.107
SMOTE Oversampling	0.083	0.128

결과적으로 CSI 점수가 가장 높았던 닻 끌림 발생 데이터만 SMOTE 기법으로 Oversampling 한 것을 채택하였다.

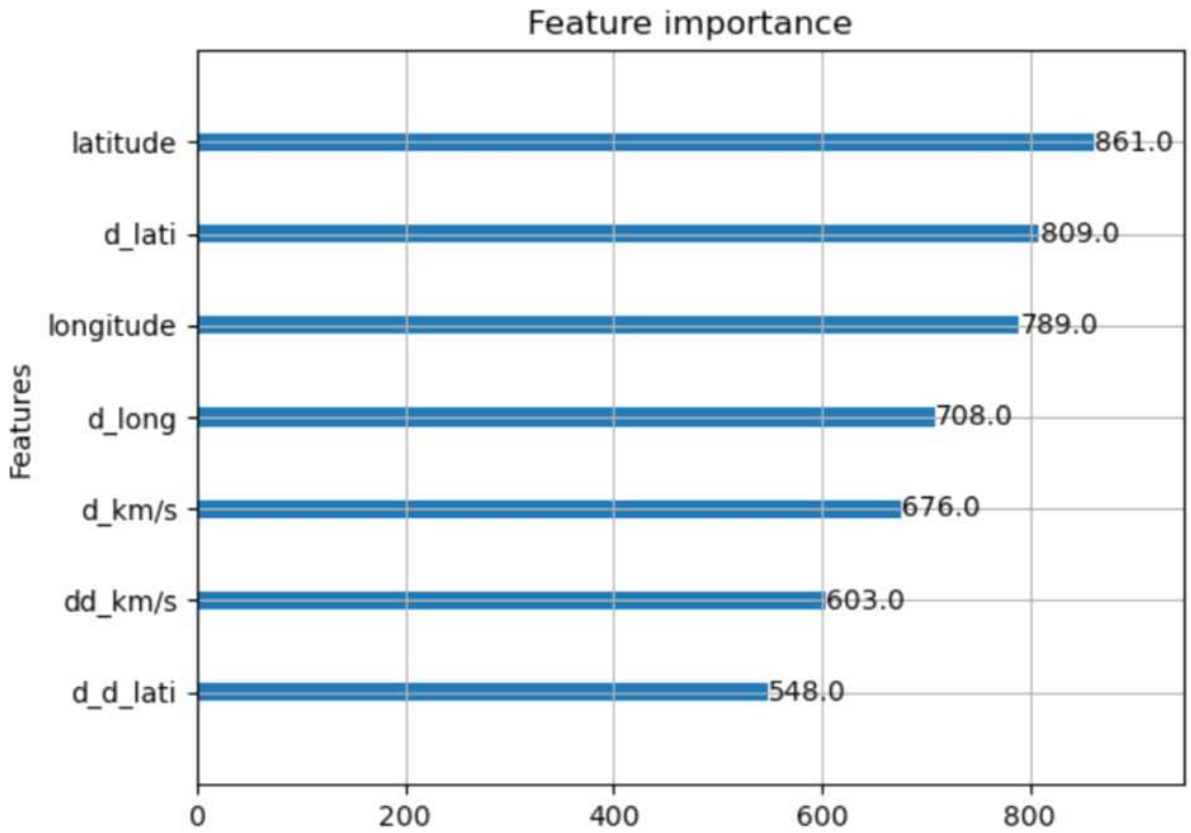
(6) Xgboost 모델링

메모리의 효율과 훈련 속도를 최적화하기 위해 Dmatrix 데이터 객체를 이용하였다. Dmatrix는 데이터를 조밀하고 최적화된 형식으로 저장함으로써 메모리 오버헤드를 줄여 더 큰 데이터셋의 학습이 가능해진다. 이와 더불어 이러한 간소화된 데이터 표현은 학습 단계에서 데이터 접근과 조작을 빠르게 수행할 수 있어 전체적인 학습 과정을 가속화 한다.

모델 성능 개선을 위하여 하이퍼파라미터 튜닝을 실시한 결과, 다음 표와 같이 파라미터를 선정하였다.

파라미터	값
Objective (목적함수)	Binary : logistic
Eval_metric (성능 평가 지표)	Logloss
Num_rounds (총학습 라운드 수)	100

4. Model Analysis



최종적인 모델 학습 이후에 학습된 모델의 Feature Importance를 출력해 보았다. 각 지표는 F-Score로 계산되었다. F-Score는 개별 특성이 모델의 전체적인 예측 능력에 기여하는 정도를 측정하는 척도로, 트리 구축 과정 중 데이터셋 내의 불순도를 얼마나 줄이는데 특성이 기여하는지를 측정한다. 높은 F-Score는 특성이 정확한 예측에 더 큰 영향을 미친다는 것을 나타내며, 모델 성능에 어떤 특성이

영향을 주는지 이해하는데 중요한 도구로 사용된다.

총 12개의 Feature로 학습을 진행하였고 그중에 상위 Feature 들만 출력한 것이다. 앞서 생성하였던 변화율 관련 파생 변수들이 높은 F-Score를 나타내며, 이는 예측에 유의미한 변수였다는 사실을 알 수 있다.

5. Summary & Conclusion

선박의 움직임에 대한 데이터를 분석하고, 배의 정박과 닻 끌림이 발생할 때의 그래프를 그려 항적 패턴의 차이를 확인하였다. 이 패턴을 학습하기 위해 위도와 경도의 시간에 따른 변화율을 계산하였고, 변화율의 변화율을 계산하여 벡터의 변화 추이를 파악하였다.

그러나 앞서 구한 벡터들은 Euclidean Coordinate에서의 값이므로, 지구의 실제 거리와는 차이가 있다. 이를 해결하기 위해 실제 거리를 시간으로 나눈 속력을 구하고 뉴턴의 제2 법칙 (가속도의 법칙)인 F = ma를 이용할 방법을 찾아 보게 되었다. 기상현상에 영향을 받아 배가 움직인다면 배에 외력이 작용해서 움직이는 것으로 닻 끌림이 발생하게 되는 임계치가 있을 거라 가정하였다. 이때 힘의 크기는 가속도의 크기, 즉 가속력에 비례하므로 이전에 구했던 속력을 사용하여 가속력을 구하였다.

학습 이후 Feature Importance를 출력해 보았을 때, 생성했던 변화율 관련 파생 변수들이 높은 점수를 나타내며 이는 모델의 성능에 큰 영향력이 있었다는 사실을 확인할 수 있다.

선박의 움직임을 이해하고 예측하는 데에 중요한 변수들을 도출함으로써 거리와 외력의 실제 영향을 고려한 선박의 움직임 예측 모델을 개발하였으며, 이 모델과 분석 결과를 기반으로 선박의 운항 경로 최적화와 날씨 변화에 따른 예측 모델 개발 등 다양한 응용 분야에서도 이 모델을 활용할 수 있다. 이러한 정확하고 신뢰할 수 있는 선박의 움직임 예측은 선박 운항 관리에 중요한 정보를 제공하여 안정성을 향상시킬 뿐만 아니라 운영 비용을 절감하고 자원 효율성을 향상하는 데에도 도움이 될 것이다.

6. Utilization Plan & Expected Effect

첫 번째 활용방안은 실시간으로 닻 끌림을 인지하는 방안이다. 선박의 실시간

위치 데이터 및 기타 분석에 필요한 데이터를 수집하고, 이를 활용하여 전처리한 뒤, 모델을 활용하여 닻 끌림 발생 여부를 판단한다. 닻 끌림이 발생했다고 판단되면 해당 선박의 선원들이 인지할 수 있게 시각 또는 청각적 수단을 활용하여 경보 알림을 전송한다. 이후에 해당 선박의 선원들이 닻 끌림 발생 가능성이 낮은 위치로 선박을 이동한다.

두 번째 활용방안은 예측을 통한 방안이다. 기상청 및 유관기관의 일기예보 및 기상현상 예측 데이터를 수집한다. 수집한 기상현상 예측 데이터를 활용하여 일정 시간 이후의 파생 변수 및 선박의 위치를 예측한다. 이후 모델을 활용하여 예측된 선박의 닻 끌림 발생 여부를 판단한다. 닻 끌림이 예측되면 해당 선박에 이동 조치 시행 후, 해당 구역을 정박 금지 구역으로 설정한다.

기대효과로는 첫째, 선박 표류로 인한 충돌 또는 좌초의 위험을 방지하여 선박, 선원 및 주변 해양 환경의 안전을 보호할 수 있다. 둘째, 선박의 닻 끌림 발생으로 인해 발생하는 파괴된 선박의 잔해, 유출된 석유등으로 인한 해양오염 및 해양 생태계 파괴를 방지할 수 있고, 이는 많은 금전적 손해를 방지할 수 있다. 셋째, 닻 끌림으로 인해 발생할 수 있는 문제인 광케이블 및 연료 파이프와 같은 중요 수중 구조물의 손상을 방지할 수 있고, 비용과 시간이 많이 드는 수리 및 서비스 중단을 방지할 수 있다. 넷째, 시간이 지남에 따라 지속적인 데이터 수집을 통하여 모델의 정확도와 안정성을 향상할 수 있고, 이후에 기후가 변하더라고 새로 학습하여 사용할 수 있다. 마지막으로 즉각적인 판단 및 조기 탐지를 통하여 불필요한 연료 소비를 방지하여 연비가 좋아져 선박의 수익에 긍정적인 영향을 미칠 수 있다.

7. Future Work

Oversampling을 진행하여 데이터 불균형 문제를 해소한 뒤 학습하는 과정에서 Xgboost 모델만 진행하였고, Oversampling의 비율도 5:5와 6:4만 이용해 보았다. 앞으로 더 나아가 다양한 비율의 Oversampling과 Xgboost를 제외한 다양한 모델들(Random Forest, LightGBM, DNN 등)을 사용하여 학습해보고 결과들을 비교하여 더 나은 모델을 찾아가는 것이 목표이다.

8. GitHub

https://github.com/SDasom/Weather_Bigdata_Contest

9. Reference

* Jerrold E. Marsden(2012), Vector Calculus, Sixth Edition, Freeman ancCompany
* AurélienGéron(2019), Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition,O'ReillyMedia, Inc.
* David Mertz(2021), Cleaning Data for Effective Data Science, Packt
* 민일홍, 김준우, 손윤준(2021), 구조임무를 수행하는 수상함의 주묘현상개선 연구, 국방기술품질원 함정센터, Journal of the Korea Academia-Industrial cooperation Society Vol. 22, No. 12 pp. 193-199, 2021
* 박성호(2006), 부산항 VTS의 효율적인 운영방안에 관한 연구, 한국해양대학교 대학원 운항 시스템공학과
* 이윤석, 정연철, 김세원, 윤종휘, 배석한, 구엔풍(2005), 묘박중인 선박의 주표 한계에 관한 연구(I), 한국해양항만학회 제29권 제1호 춘계학술대회논문집. p165~p171, 2005.4
* 민건태, "남외항묘박지'떠다니는 폭탄' 선박사고 속수무책",「국제신문」, 2014.01.17
* 김기태 , "포항항 잦은 선박사고 원인 캔다.",「경북매일」, 2014.01.07
* 배형욱, "영일만항 침몰 청루-15호 사망자 유족들 시신 6구 찾아가",「경북일보 」, 2013.11.24
* 이승훈, "지난해 부산 최악 해양오염 ' 오션탱크호좌초' 꼽혀",「부산일보 」, 2017.01.11.

청렴하고 신뢰받는 기상청을 만들겠습니다.

기상청

기 상 청

수신 수신자 참조
(경유)
제목 「2023 날씨 빅데이터 콘테스트」 수상 실적 확인

1. 귀하의 무궁한 발전을 기원합니다.

2. 기상청에서 주최한 「2023 날씨 빅데이터 콘테스트」 수상 실적을 다음과 같이 증빙합니다.

가. 대회명: 2023 날씨 빅데이터 콘테스트
나. 과제명: (해양안전) 기상에 따른 **선박 닻 끌림 예측**
다. 수상결과: **입선**
라. 수상자: **Seven 8 Nine**(권병근, 이다인, 성다솜, 신호섭, 박민서). 끝.

기 상 청 장

수신자 권병근 귀하, 이다인 귀하, 성다솜 귀하, 신호섭 귀하, 박민서 귀하

주무관	**김동선**	★기상사무관	**병가**	기상융합서비스과장	전결 2023.8.16. **김영동**
협조자					

시행 기상융합서비스과-2413 (2023. 8. 16.) 접수

우 07062 대전광역시 서구 청사로 189, 정부대전청사 1동 (둔산동) / http://www.kma.go.kr

전화번호 042-481-7488 팩스번호 02-2181-0925 / sayds101@korea.kr / 비공개(6)

위험기상과 기후위기로부터 안전한 국민, 든든한 국가

1 - 1