

데이터분석, AI(머신러닝, 딥러닝)모델 개발

# PORTFOLIO

지원자 : 김서정

souljoin0229@gmail.com

# CONTENTS

---

- 인적사항 및 주요경력

2020.08 서울시립대학교 공간정보공학과 졸업

- 프로젝트 수행 이력

2020.08 ~ 2020.09 서울시 강남구 스쿨존 교통사고 위험지역 예측모델 개발 프로젝트

2020.09 ~ 2020.10 자연어처리 기반의 KOSPI 및 YG엔터주가 예측 프로젝트

2020.11 ~ 2020.11 리그오브레전드 인게임 오브젝트와 승률간의 관계 분석

## 인적사항 및 주요경력



**지원분야** 데이터 분석 및 AI모델(머신러닝, 딥러닝) 개발

**인적사항** 이 름 김서정  
 생년월일 1996년 2월 29일 (만 24세, 여)  
 연 락 처 010-7301-6533  
 이 메 일 [souljoin0229@gmail.com](mailto:souljoin0229@gmail.com)

학 력

2020.08 서울시립대학교 공간정보공학과 졸업  
(학점 3.41/4.50)

2015.02 설월여자고등학교 졸업

## 지원각오

데이터와 함께 성장하는 인재가 되고 싶습니다!

## 프로젝트 수행 이력

프로젝트명	서울시 강남구 스쿨존 교통사고 위험지역 예측모델 개발 프로젝트
목적	머신러닝을 통해 서울시 강남구 스쿨존 내 교통사고 발생 위험 지역을 예측하고, 앱 또는 웹 사이트에서 해당 지역의 사고 위험도를 알림
수행기간	2020.08 ~ 2020.09
팀 구성 및 역할	4인 1개팀 / - 데이터 전처리 - 머신러닝 모델 설계 및 구현 - Map 웹사이트 구현
사용언어 및 도구	Python, Jupyter Notebook, Javascript & css, KAKAO MAP API, Pycharm
Github URL	<a href="https://github.com/kimseojeong6533/SZ-Wannabe/blob/master/%EC%8A%A4%EC%BF%A8%EC%A1%B4%EC%9C%84%ED%97%98%EC%98%88%EC%B8%A1_EDA_Modeling_Mapping.ipynb">https://github.com/kimseojeong6533/SZ-Wannabe/blob/master/%EC%8A%A4%EC%BF%A8%EC%A1%B4%EC%9C%84%ED%97%98%EC%98%88%EC%B8%A1_EDA_Modeling_Mapping.ipynb</a>
비고	서울시와 ICTкомплек스가 함께하는 2020 ICT콕 AI공모전 장려 상 입상

[https://youtu.be/OGY-zS1\\_KOE](https://youtu.be/OGY-zS1_KOE)



## 프로젝트명

서울시 강남구 스쿨존 교통사고  
위험지역 예측모델 개발 프로젝트

## 수행기간

2020.08 ~ 2020.09

## 수행 단계 및 방법

1. 데이터 전처리
2. 도로별 사고발생 횟수를 카운트하여 위험도 등급(Label값) 구분
3. Label값에 대한 각 Feature의 분포를 확인하여 공간적 상관성이 높은 Feature 추출
4. 앙상블트리모델 중 의사결정나무, 랜덤포레스트, 그래디언트 부스팅 회귀트리모델의 각 Train\_Test\_split\_ratio, max\_depth, n\_estimator 등의 파라미터 별 성능 비교
5. 스쿨존 주소 기반의 Testset을 모델에 input하여 위험지역 도출
6. Javascript, kakaomap api를 이용해 위험지역을 시각화한 웹사이트 생성

프로젝트명	서울시 강남구 스쿨존 교통사고 위험지역 예측모델 개발 프로젝트
수행기간	2020.08 ~ 2020.09

## 수행 단계 및 방법

### 1. 데이터 전처리 (서울 열린 데이터광장의 공공데이터 이용)

- 강남구 데이터 외 삭제
- 사고 번호 삭제
- 사고 일시 → 연도/월/일/시로 분류
- 시군구, 도로명 → 위도, 경도변환 후 속성 추가(지오코딩 활용)
- 사고내용 → 사망자수, 중상자수, 경상자수, 부상신고자수 데이터 분류
- 사고 유형 -> 차대차/차대사람으로 분류
- 법규 위반, 노면 상태 삭제
- 기상 상태 → 기타 항목 데이터 많음, 흐림 데이터 양 비율로 랜덤하게 분류
- 도로 형태 → 기타 항목 제거, 교차로 및 단일로 항목만 분류

프로젝트명

서울시 강남구 스쿨존 교통사고  
위험지역 예측모델 개발 프로젝트

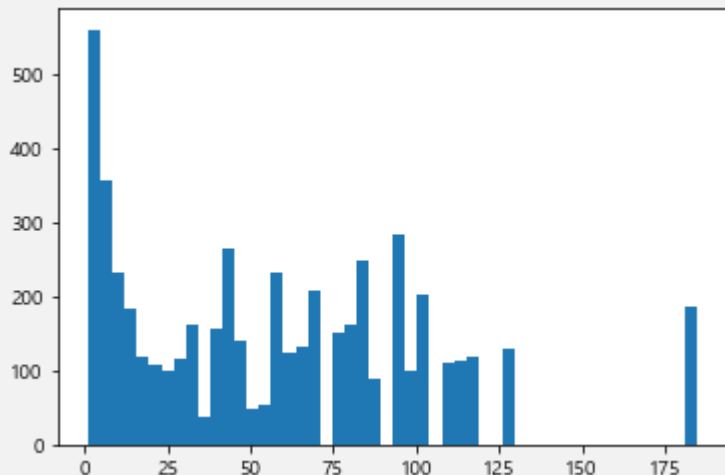
수행기간

2020.08 ~ 2020.09

## 수행 단계 및 방법

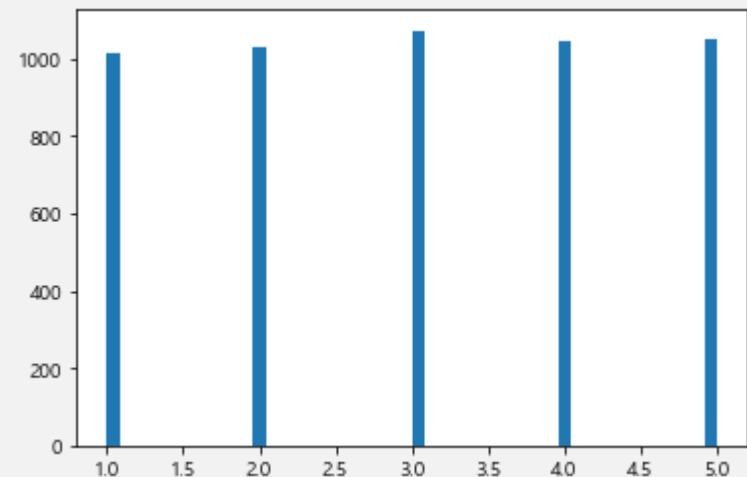
### 2. 도로별 사고발생 횟수를 카운트하여 위험도 등급(Label값) 구분

```
# 사고발생수 분포확인  
x=np.array(df['사고발생수'])  
print(x)  
s = pd.Series(x)  
s.describe()  
  
n, bins, patches=plt.hist(x, bins=50)
```



```
num = pd.Series(x)
```

```
Q1 = num.quantile(.2)  
Q2 = num.quantile(.4)  
Q3 = num.quantile(.6)  
Q4 = num.quantile(.8)  
Q5 = num.quantile(1)
```





프로젝트명

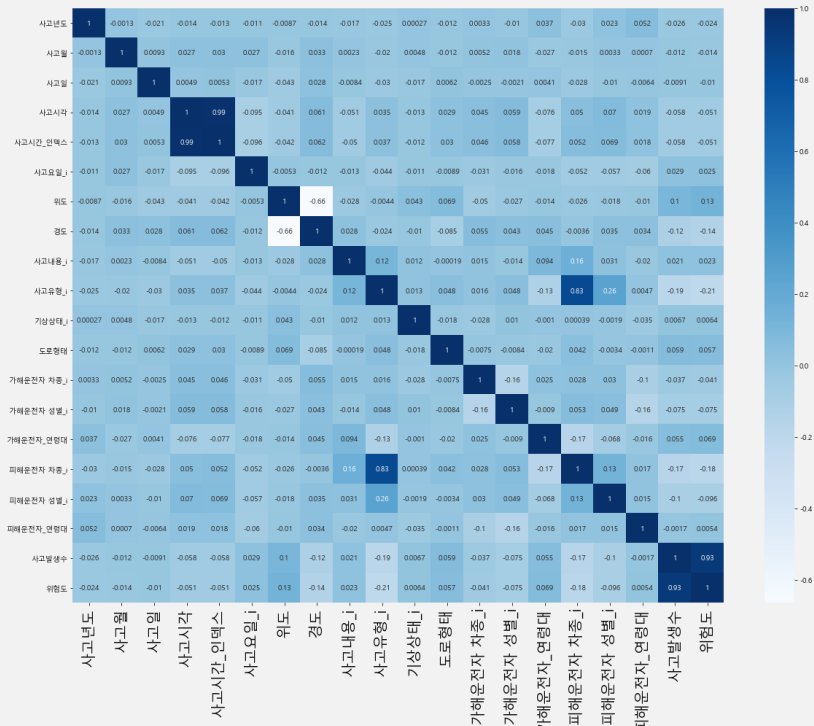
서울시 강남구 스쿨존 교통사고  
위험지역 예측모델 개발 프로젝트

수행기간

2020.08 ~ 2020.09

## 수행 단계 및 방법

### 3-1. Label값에 대한 각 Feature의 EDA 및 Input Feature 도출



프로젝트명

서울시 강남구 스쿨존 교통사고  
위험지역 예측모델 개발 프로젝트

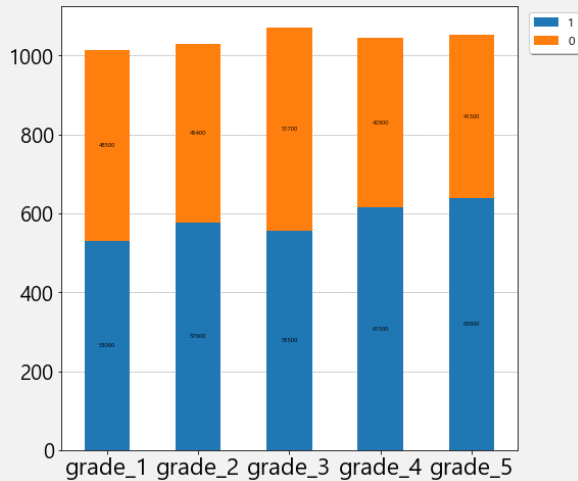
수행기간

2020.08 ~ 2020.09

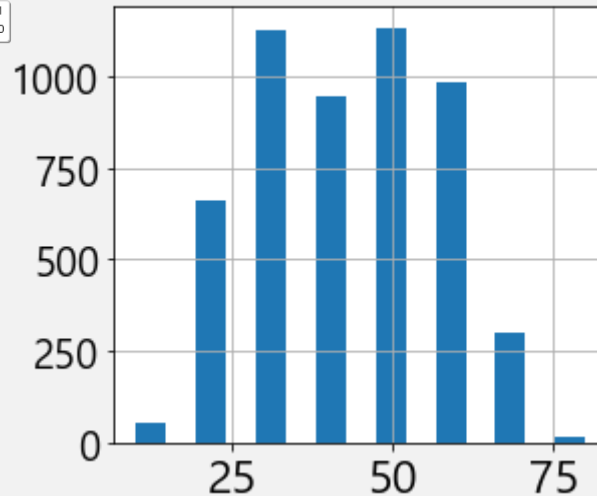
## 수행 단계 및 방법

### 3-2. Label값에 대한 각 Feature의 EDA 및 Input Feature 도출

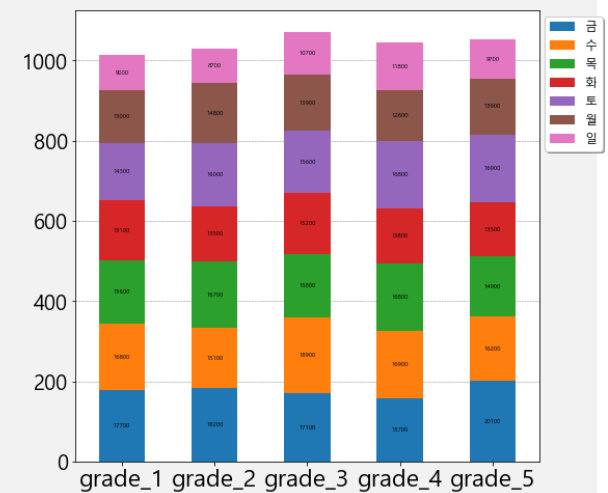
위험도 별 도로형태 (0:단일로, 1:교차로)



피해운전자 연령대별 분포



위험도 별 요일



→ '위도','경도','사고시간\_인덱스','사고요일\_i','기상상태\_i','도로형태','위험도'

## 4. 의사결정나무, 랜덤포레스트, 그래디언트 부스팅 회귀트리모델의 파라미터별 성능비교

```
# Gradient Boosting Regression Tree - 7 : 3
```

```
for i in range(1,6):
    X_train, X_test, y_train, y_test = train_test_split(data.iloc[:, :-1], data['위험도'], test_size=0.3, shuffle=True, random_state=0)
    gbdt = GradientBoostingClassifier(max_depth=i, random_state=111, n_estimators=33)
    gbdt.fit(X_train, y_train)

    print('Max_depth : {}'.format(i))
    print("훈련 세트 정확도: {:.3f}".format(gbdt.score(X_train, y_train)))
    print("테스트 세트 정확도: {:.3f}".format(gbdt.score(X_test, y_test)))
    print('특성 중요도 : \n', gbdt.feature_importances_)
    print()
```

Trial condition	Max_depth	Random Forest (Train data, Test data) (n_estimators=33)	Decision Tree (Train data, Test data)	Gradient Boosting Regression Tree (Train data, Test data) (n_estimators=33)
5:5	1	0.320, 0.311	0.307, 0.336	0.489, 0.502
	2	0.416, 0.428	0.415, 0.430	0.698, 0.689
	3	0.459, 0.463	0.418, 0.435	0.746, 0.736
	4	0.531, 0.532	0.534, 0.544	0.802, 0.772
	5	0.632, 0.606	0.590, 0.602	0.824, 0.783
7:3	1	0.314, 0.339	0.314, 0.339	0.494, 0.504
	2	0.398, 0.409	0.382, 0.386	0.686, 0.689
	3	0.474, 0.474	0.406, 0.404	0.747, 0.737
	4	0.522, 0.519	0.489, 0.499	0.790, 0.776
	5	0.562, 0.555	0.553, 0.573	0.818, 0.788
8:2	1	0.332, 0.345	0.318, 0.334	0.812, 0.779
	2	0.416, 0.434	0.379, 0.397	0.812, 0.779
	3	0.456, 0.461	0.406, 0.405	0.812, 0.779
	4	0.529, 0.539	0.490, 0.497	0.812, 0.779
	5	0.594, 0.574	0.557, 0.579	0.812, 0.779

프로젝트명

서울시 강남구 스쿨존 교통사고  
위험지역 예측모델 개발 프로젝트

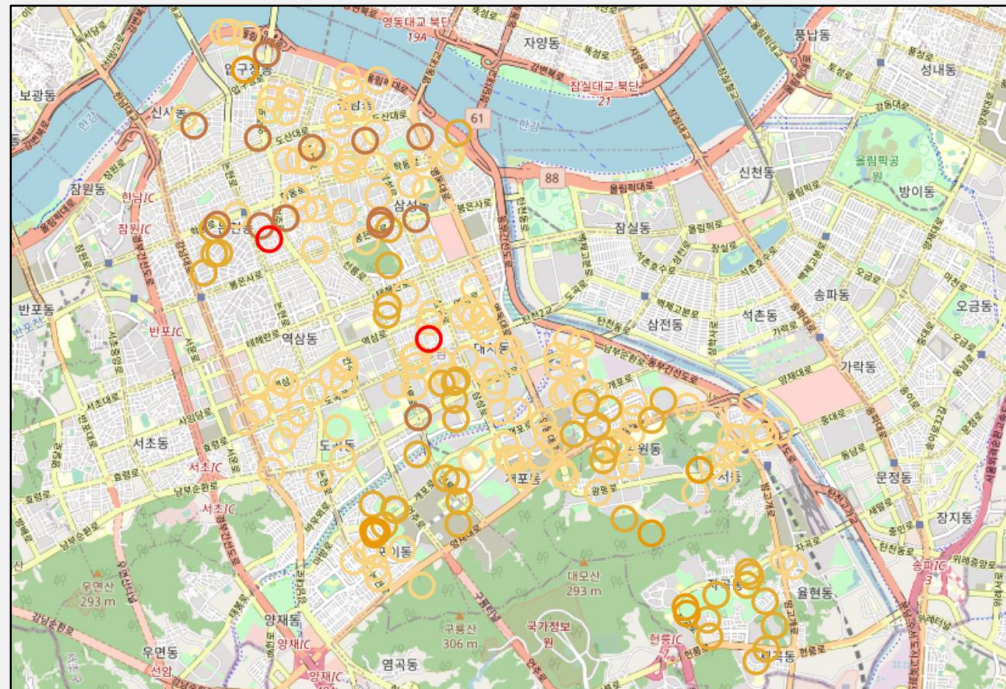
수행기간

2020.08 ~ 2020.09

## 수행 단계 및 방법

### 5. 스쿨존 주소 기반의 Testset을 모델에 input하여 위험지역 도출

- 1단계 위험도
- 2단계 위험도
- 3단계 위험도
- 4단계 위험도
- 5단계 위험도



프로젝트명

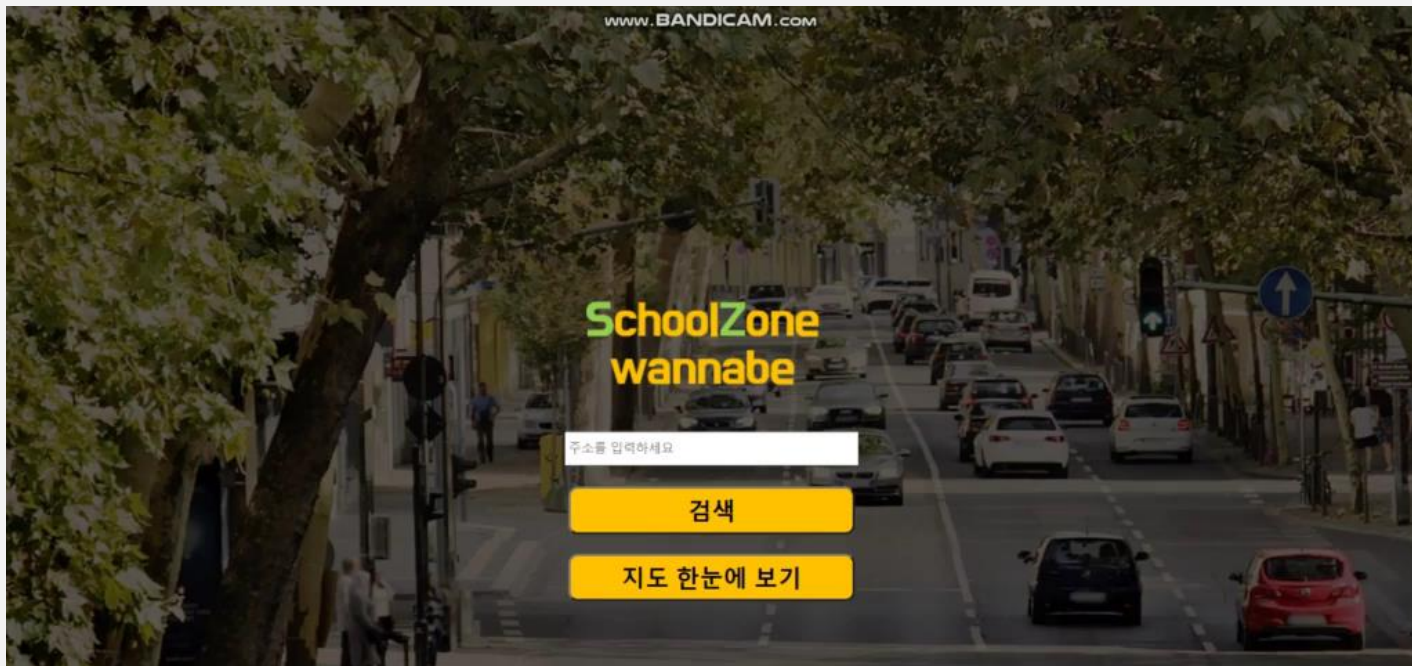
서울시 강남구 스쿨존 교통사고  
위험지역 예측모델 개발 프로젝트

수행기간

2020.08 ~ 2020.09

## 수행 단계 및 방법

### 6. Javascript, kakaomap api, Pycharm를 이용해 위험지역을 시각화한 웹 생성



## 프로젝트 수행 이력

프로젝트명	자연어처리 기반의 KOSPI 및 YG엔터주가 예측 프로젝트
목적	경기에 영향을 받는 코스피와 테마주를 뉴스데이터와 prophet, AutoML, 강화학습 알고리즘을 활용하여 예측하고자 함
수행기간	2020.09 ~ 2020.11
팀 구성 및 역할	5인 1개팀 / - 학습데이터 전처리 - FBProphet을 사용하여 금융 시계열 분석 - Jupyter notebook을 이용한 전체 코드 통합
사용언어 및 도구	Python, Google Colab, Jupyter notebook, Pytorch, Tensorflow, Bert
Github URL	<a href="https://github.com/ejihoon6065/Project_TurnAround">https://github.com/ejihoon6065/Project_TurnAround</a>

<https://youtu.be/aDm0r-bh3I>

app · Streamlit

www.BANDICAM.COM

localhost:8501

예측 방법 결정

Online

프로젝트명 : 자연어 처리 기반의 투자 분석 및 예측시스템 개발

멘토님 : 정좌연 PE

팀명 : 턴어라운드

팀원 : 이지훈, 이문형, 강민재, 구병진, 김서정

15.89 +5.34% 300.000  
45.34 -7.89% 120.000  
17.34 +5.97% 320.000  
34.89 +2.13% 900.000  
18.45 +6.43% 600.000  
23.67 -11.6% 300.000  
34.64 +23.1% 120.000  
43.69 +5.56% 320.000  
12.78 -3.67% 150.000  
4.44 +11.3% 120.000

BANDICAM BETA UNREGISTERED

00:00:00  
0 bytes / 122.0GB

REC

1920x1080 - (0, 0), (1920, 1080) - 디스플레이 1

홈

시작하기 | 비디오 | 이미지

← →

화면 녹화 모드 - 전체 화면

모니터 화면 전체를 녹화할 때 사용하는 방식입니다.

1. 녹화대상 모니터(디스플레이)를 선택합니다.  
2. 녹화시작 단축키나 'REC' 버튼을 눌러 녹화를 시작합니다.

REC 녹화 시작 하기 온라인 도움말 보기

녹화 시작 / 정지

녹화 시작/정지 F12 이미지 캡처 F11

BANDICAM

[삼성 갤럭시 스마트폰]은 밴디캠에서 고화질로 녹화가 가능합니다.

가 예측 모델

	PREC	F1
	0.9246	0.8238

검색하려면 여기에 입력하십시오.

오전 4:57  
2020-11-06

# 프로젝트 수행 이력

프로젝트명	자연어처리 기반의 KOSPI 및 YG엔터 주가 예측 프로젝트
수행기간	2020.09 ~ 2020.11

## 수행 단계 및 방법

### 1. 데이터 크롤링

- 1-1. KRX, Yahoo Finance, Investing.com 등 주가 및 투자보조지표 등의 데이터를 크롤링하여 2018. 01. 01 ~ 2020. 10. 26 주가데이터 확보
- 1-2. 한국경제 신문의 경제,국제부문의 2018. 01. 01 ~ 2020. 10. 26 뉴스기사 타이틀(1일당 50개 기사)을 크롤링

### 2. 자연어처리 (TF계산)

- 2-1. Mecab 형태소분석기를 이용해 텍스트 데이터를 정제
- 2-2. 텍스트의 Unigram,Bigram단어들의 TF를 구하고 Top1000의 단어를 뽑아 복합명사 일체화 및 단일 글자 처리
- 2-3. 한자를 한글로, 고유명사를 각 도메인으로 단어 치환
- 2-4. 코스피 등락률을 확인하여 전일 기사의 라벨링에 반영

### 3. AutoML, Prophet 등을 활용한 등락 예측

- 3-1. 코스피, YG주가에 대한 회귀모델, 분류 모델 생성



# 프로젝트 수행 이력

프로젝트명

자연어처리 기반의 KOSPI 및 YG엔터  
주가 예측 프로젝트

수행기간

2020.09 ~ 2020.11

## 1. 크롤링

```
# 종합지수 (코스피) 차트 데이터
```

```
kospi_ = stock.get_index_ohlcv_by_date(start_date_, end_date_, "1001")
```

```
kospi_.columns = ['Open', 'High', 'Low', 'Close', 'Volume']
```

```
# 코스피 투자자별 공매도 거래량
```

```
kospi_short_sell_volume = stock.get_shorting_investor_volume_by_date(start_date_, end_date_, "KOSPI")
```

```
kospi_short_sell_volume.columns = ['kospi_inst_volume', 'kospi_indi_volume', 'kospi_fore_volume', 'kospi_etc']
```

```
# 코스피 투자자별 공매도 거래대금
```

```
kospi_short_sell_value = stock.get_shorting_investor_price_by_date(start_date_, end_date_, "KOSPI")
```

```
kospi_short_sell_value.columns = ['kospi_inst_value', 'kospi_indi_value', 'kospi_fore_value', 'kospi_etc_val']
```

2018-10-31

허성무  
창원시  
장, 김  
동연  
경제부  
총리에  
"산업  
체질  
개선  
위한  
전략  
산업" 원  
요청

[대  
출금  
제에  
금리  
인상  
압박  
까지  
] 전문  
가  
"빛  
테크  
전략  
재정  
비해  
아"

[대  
출금  
제에  
금리  
인상  
압박  
까지  
] 11월  
대출  
시장  
'한  
파'  
예고

신세  
계그  
를, 1  
유확  
정...  
2023  
년 매  
출 10  
조 '한  
국판  
아마  
존' 키  
운다

북미  
권역  
본부  
격동  
하는  
현대  
차...  
파  
찾  
나

KT&G,  
릴 100만대  
판매 기념  
'보상판매  
이벤트' 실시

롯데  
제과,  
빼빼로  
공식  
캐릭터  
'빼빼로  
일레븐'  
선보여

'공유  
주방'  
시장수  
록 커진  
다...  
주방으로  
들어온  
공유  
경제

'존경  
받는  
기업'  
안  
늘  
다...  
중기부,  
중소  
사  
중심  
문화  
확산  
지원

...  
'유인  
드론'  
시험비  
행쉬워  
진  
다...  
건축물  
점검  
에도  
드론  
허용

[2018  
수입  
차결  
산] 한  
경닷컴  
이  
뽐  
수  
올  
해의  
차에  
티구  
안

KFC,  
올해  
13번째  
신  
매  
장  
신  
역  
오픈

DGB  
금융  
편입  
하이투  
자증권  
출  
범...  
"금융  
투자  
톱10  
도약"

손보사  
빅4,  
모  
등  
편  
바일  
기  
우  
편  
서비스  
도  
입...  
"비  
용절감  
특목"

인민  
은행  
"11월  
러=7  
위안  
막아  
라"...  
홍  
서  
3  
조  
원  
채권  
발행

증  
신  
우  
삼  
재  
보  
고  
심  
의  
들  
입...  
분  
회  
가  
공  
보  
발  
2차  
진

# 프로젝트 수행 이력

프로젝트명

자연어처리 기반의 KOSPI 및 YG엔터  
주가 예측 프로젝트

수행기간

2020.09 ~ 2020.11

## 2. 자연어처리

TF(Text Frequency) :  
Uni-gram

Word	TF
미국	214
금융	202
한국	186
경제	177
기업	152
달러	146
LG	127
일본	123
정보	123
구조	113

TF(Text Frequency) :  
Bi-gram

Word	TF
구조조정	95
정보통신	51
LG전자	38
시스템개발	36
금융기관	34
천만달러	31
벤처기업	29
한국경제	28
금리인하	28
금융위기	26

현대차 'N'에 자극받은 토요타, 고성능 소형 확대

↓  
Mecab

현대차 자극 토요타 성능 소형 확대

# 프로젝트 수행 이력

프로젝트명	자연어처리 기반의 KOSPI 및 YG엔터 주가 예측 프로젝트
수행기간	2020.09 ~ 2020.11

## 3. 사용한 모델

KOSPI

### Regression Model

Prophet

AutoML

(TheilSen, Linear Regression, Ridge)  
→ Ensemble

### Classification Model

AutoML

(Gradient Boosting,  
Linear Discriminant,  
Ridge) → Ensemble

NLP

(Bert, LSTM, AutoML)

### Reinforcement Learning

YG

Prophet

AutoML

(Linear Regression, RANSACR)  
→ Ensemble

AutoML  
(Gradient Boosting,  
Decision Tree,  
LightGBM) → Ensemble

NLP

(Bert, LSTM, AutoML)

A2C  
(value & policy network  
LSTM)

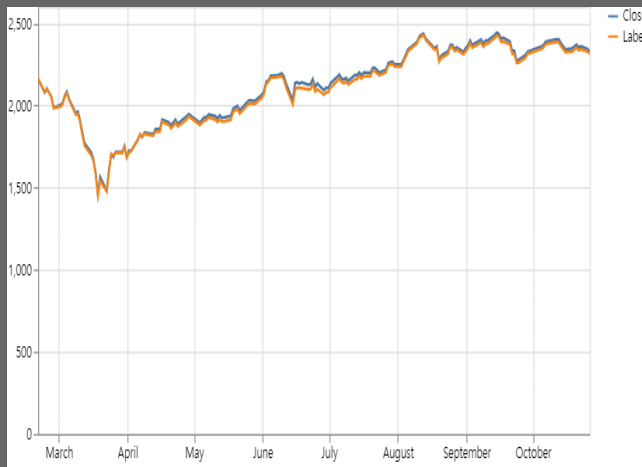
# KOSPI Model

## Regression Model

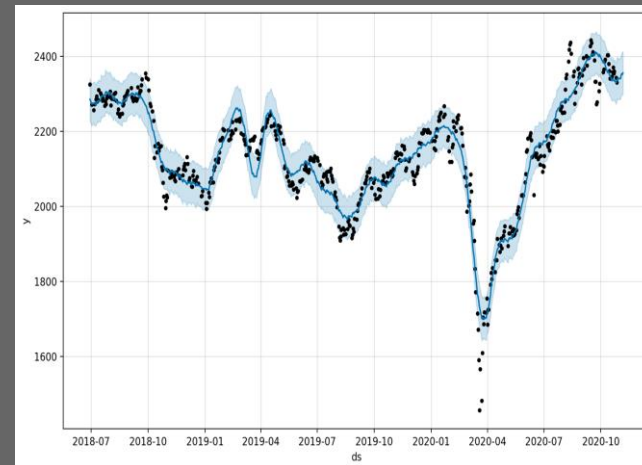
### Test Metric Scores

	MAE	MSE	RMSE	$R^2$
Prophet	85.415	15,005	122.4969	-5.5035
AutoML	16.1176	311.3388	17.6446	0.9942

### Actual & Prediction Graph



AutoML



Prophet

# KOSPI Model

## Classification Model

### Test Metric Scores

	Accuracy	ROC AUC	Recall	Precision	F1
AutoML	0.9152	0.9081	0.8676	0.9219	0.8939
NLP Bert	0.8008	0.8205	0.7429	0.9246	0.8238
NLP LSTM	0.8005	0.8069	0.7645	0.8729	0.8151
NLP AutoML	0.8088	0	0.859	0.8171	0.6068

# YG Model

## Regression Model

### Test Metric Scores

	MAE	MSE	RMSE	R <sup>2</sup>
Prophet	1686.042	4,849,041	2202.053	0.9422
AutoML	1,491.8399	2,835,922.830	1,684.3167	0.9724

### Actual & Prediction Graph



AutoML



Prophet

## YG Model

Classification Model

### Test Metric Scores

	Accuracy	ROC AUC	Recall	Precision	F1
AutoML	0.8844	0.8782	0.8387	0.8814	0.8595

**선행지표, 보조지표 및 뉴스 데이터를 활용**



**예측 정확도 상승가능**



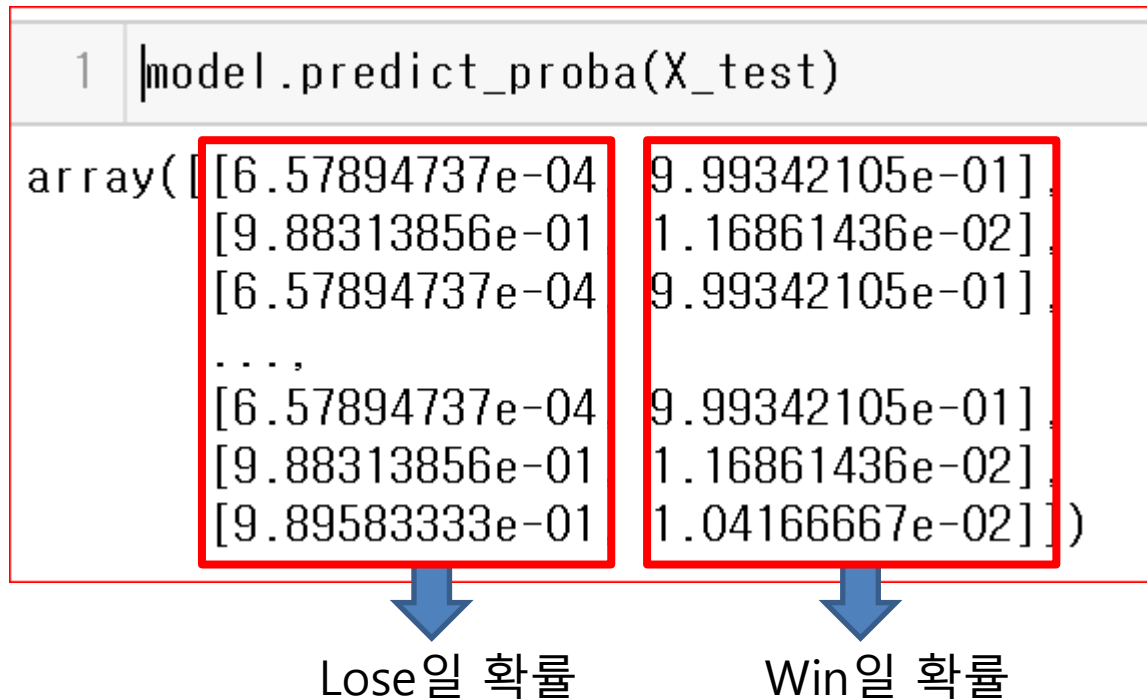
**선행지표, 기술적 분석 및 뉴스 분석을  
주식투자 의사결정에 활용가능**



## 프로젝트 수행 이력

프로젝트명	리그오브레전드 인게임 오브젝트와 승률간 데이터분석
목적	유저의 인게임 오브젝트 획득율 및 오브젝트와 승률간의 분석을 수치로 나타내어 게임 상황에 따라 예측된 승률을 알려주기 위함.
수행기간	2020.11
팀 구성 및 역할	1인 팀 / EDA,데이터 분석 및 머신러닝 모델 설계 및 구현
사용언어 및 도구	Python, Jupyter notebook, AutoML, Tensorflow, Keras
Github URL	<a href="https://github.com/kimseojeong6533/OPGG_DATAANALYSIS_HW/blob/main/Data_Analysis_HW_OPGG.ipynb">https://github.com/kimseojeong6533/OPGG_DATAANALYSIS_HW/blob/main/Data_Analysis_HW_OPGG.ipynb</a>

인게임 오브젝트 상황에 따라 승률을 알려주는 모델



# 프로젝트 수행 이력

프로젝트명	리그오브레전드 인게임 오브젝트와 승률간 데이터분석
수행기간	2020.11

## 수행 단계 및 방법

### 1. 로그 데이터 전처리

데이터 : lol 10.18 버전 KR 지역 솔로랭크 match data (약 17만건중 약 3만건 사용)  
/ KR 지역 솔로랭크 다이아몬드 티어 이상의 소환사가 1명이상 포함된 경기들

### 2. 데이터분석

- 2-1. 전체 데이터 통계를 통한 문제정의
- 2-2. 게임에서 승리하기위해서 최소 몇 개이상의 용을 먹어야할까?
- 2-3. 펑 대신 닭 = 용 대신 전령? 전령의 영향력은 얼마나 될까?

### 3. 각 오브젝트 및 인게임 환경지표를 활용한 승리 예측 모델

- 3-1. 지도학습- 분류 모델 생성 및 블렌딩

# 프로젝트 수행 이력

프로젝트명

리그오브레전드 인게임 오브젝트와  
승률간 데이터분석

수행기간

2020.11

## 1. 로그 데이터 전처리

Json

```
imeline":{"lane":"MIDDLE","participantid":1,"csdiffpermindealtas":{"10-20":-0.8,"0-10":-0.1},"goldpermindealtas":10,"xpermindealtas":{"10-20":400,1,"0-10":434.8},"role":"SOLO","damagetakendiffermindealtas":{"10-20":10,"317.3},"teamid":100,"spellid":12,"spellid":4,"championid":38},"stats":{"physicaldamagedealt":3704,"neutralminionskilledteajungle":0,"magicedamagedealt":5015,"totalplayerscore":0,"edealettochampions":3730,"visionwardsboughtingame":3,"damagedealttoobjectives":3250,"largestkillingspree":2,"se,item2":3859,"item3":0,"item0":2065,"firstbloodassist":false,"visionscore":58,"wardsplaced":29,"item4":0,"magicaldamagetaken":7670,"kills":3,"doublekills":0,"truedamagetaken":1281,"assists":12,"unrealkills":0,"neutns":1152,"goldspent":5975,"truedamagedealt":5345,"truedamagedealttochampions":587,"participantid":2,"pentakiotaldamagedealttochampions":5469,"totalunitshealed":5,"inhibitorkills":0,"totalscorerank":0,"totaldamagetakererk0var3":0,"perk1":8463,"perk1var1":859,"perk1var2":0,"perk1var3":0,"perk2":8444,"perk2var1":775,"perk2var2rk4var3":0,"perk5":8136,"perk5var1":11,"perk5var2":30,"perk5var3":0,"perkprimarystyle":8400,"perkstyle":8{"lane":"BOTTOM","participantid":2,"csdiffpermindealtas":{"10-20":0.2,"0-10":0.45},"goldpermindealtas":{"10-20":10,"xpermindealtas":{"10-20":312.7,"0-10":222.1},"role":"DUO_SUPPORT","damagetakendiffermindealtas":{"10-10":284.1},"teamid":100,"spellid":4,"spellid":14,"championid":497},"stats":{"physicaldamagedealt":73712,"neutralminionskilledteajungle":0,"magicedamagedealt":1359,"totalplayerscore":0,"gedealettochampions":1359,"visionwardsboughtingame":1,"damagedealttoobjectives":8905,"largestkillingspree":4,"alse,item2":3078,"item3":3158,"item0":1055,"firstbloodassist":false,"visionscore":16,"wardsplaced":6,"item4":10692,"magicaldamagetaken":7015,"kills":8,"doublekills":1,"truedamagetaken":3807,"assists":4,"unrealkills":0,"ttochampions":15219,"goldspent":10508,"truedamagedealt":14047,"truedamagedealttochampions":5467,"participant
```



DataFrame

df_wintems											
	bans	baronkills	deaths	dominionvictoryscore	dragonkills	firstbaron	firstblood	firstdragon	firstinhibitor	firsttriftherald	killsplusassists
0	{pickturn: 1, 'championid': 0, 412, {pickturn...	0	40.0	0	4	False	False	True	True	True	84.0
1	{pickturn: 6, 'championid': 2, 238, {pickturn...	2	27.0	0	3	True	True	True	True	True	78.0
2	{pickturn: 1, 'championid': 0, 84, {pickturn...	0	12.0	0	3	False	True	True	True	False	75.0

df\_total.isnull().sum() # 결측치 확인 -> 0

```
baronkills      0
deaths          0
dragonkills     0
firstbaron      0
firstblood      0
firstdragon     0
firstinhibitor  0
firsttriftherald 0
firsttower      0
gameduration    0
inhibitorkills  0
killsplusassists 0
riftheraldkills 0
teamid          0
totaldamagedealttochampions 0
```

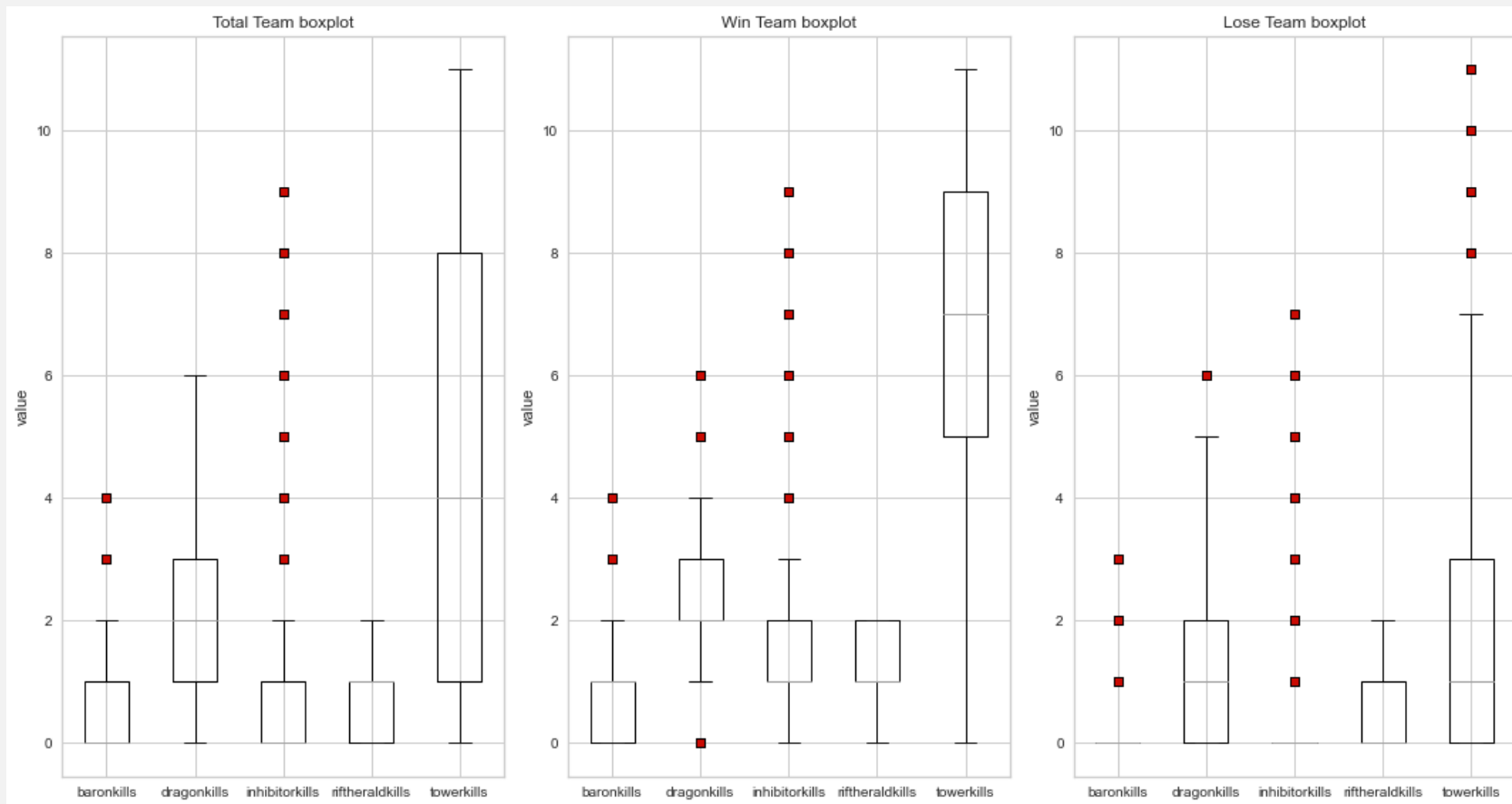
## 프로젝트 수행 이력

프로젝트명	리그오브레전드 인게임 오브젝트와 승률간 데이터분석
수행기간	2020.11

### 2. 사용 데이터 Features

Numeric_type_OBFeatures	Bool_type_OBFeatures	Numeric_type_Champion_Features
Dragonkills (용킬수)	Firstdragon (첫용)	Totaldamagedealttochampions (챔피언에게 가한 피해량)
Inhibitorkills (억제기킬수)	Firstinhibitor (첫억제기)	Totaldamagetaken (받은 총피해량)
Riftheraldkills (전령킬수)	Firstriftherald (첫전령)	totalminionskilled (CS)
Baronkills (바론킬수)	Firstbaron (첫바론)	Killsplusassists (챔피언처치및어시스트)
Towerkills (타워킬수)	Firstblood (퍼블)	Deaths (데스)
Teamid (진영-블루(100)/레드(200))	Firsttower (포블)	visionscore (시야점수)
Gameduration (경기시간)	-	-

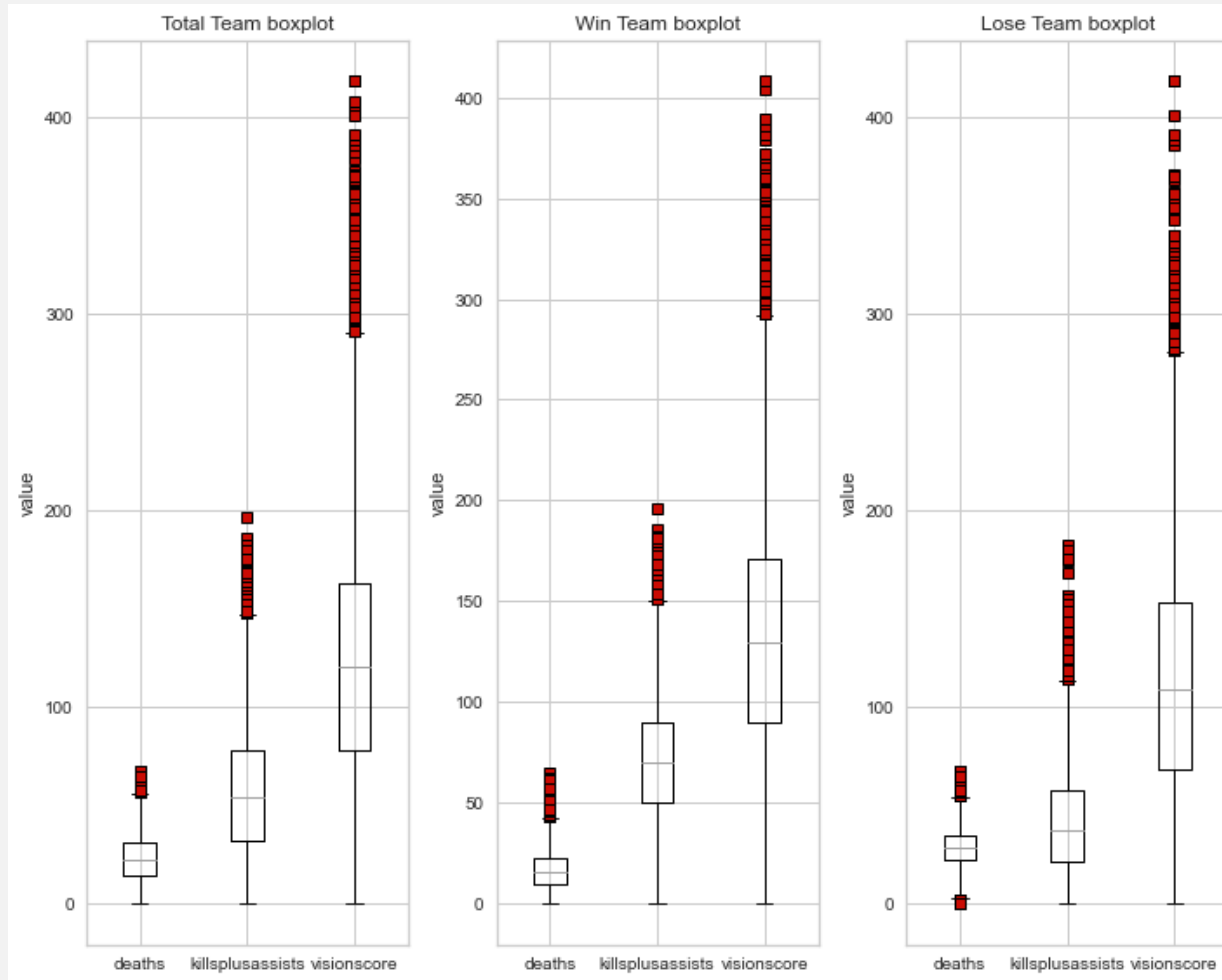
## 1) 전체 데이터 통계 확인 - 전체팀, 승팀, 패팀의 baronkills, dragonkills, inhibitorkills, riftheraldkills, towerkills 확인



패한 팀의 baronkills, dragonkills, inhibitorkills, riftheraldkills, towerkills  
모두 승팀보다 현저하게 낮음

## 1) 전체 데이터 통계 확인

- 전체팀, 승팀, 패팀의 deaths, killsplusassists, visionscore 확인

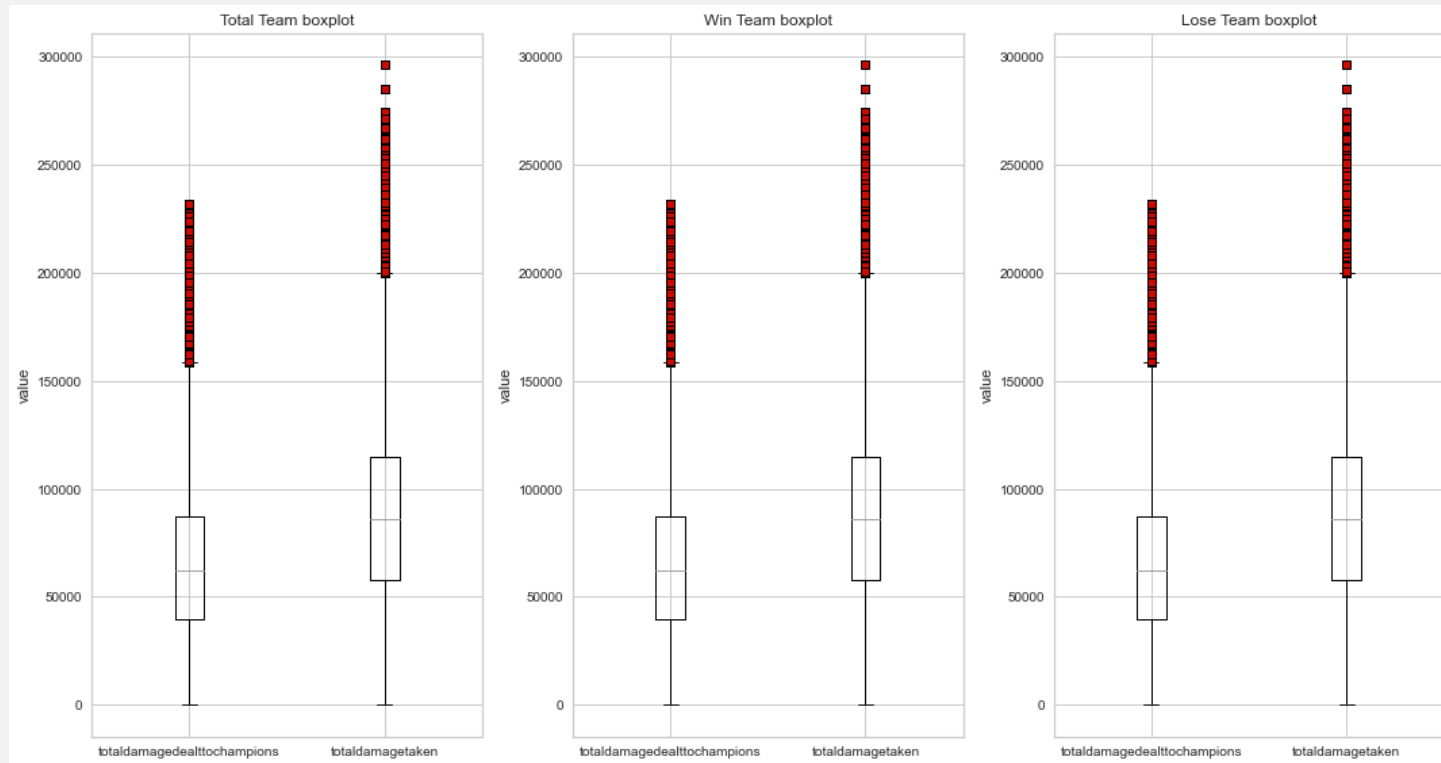


이긴팀의 통계량이  
전체팀의 통계량과  
비슷함.

반면, 진팀은 전체  
팀보다 분포가 넓음

## 1) 전체 데이터 통계 확인

- 전체팀, 승팀, 패팀의 챔피언에게 가한 총피해량, 받은 총 피해량 확인

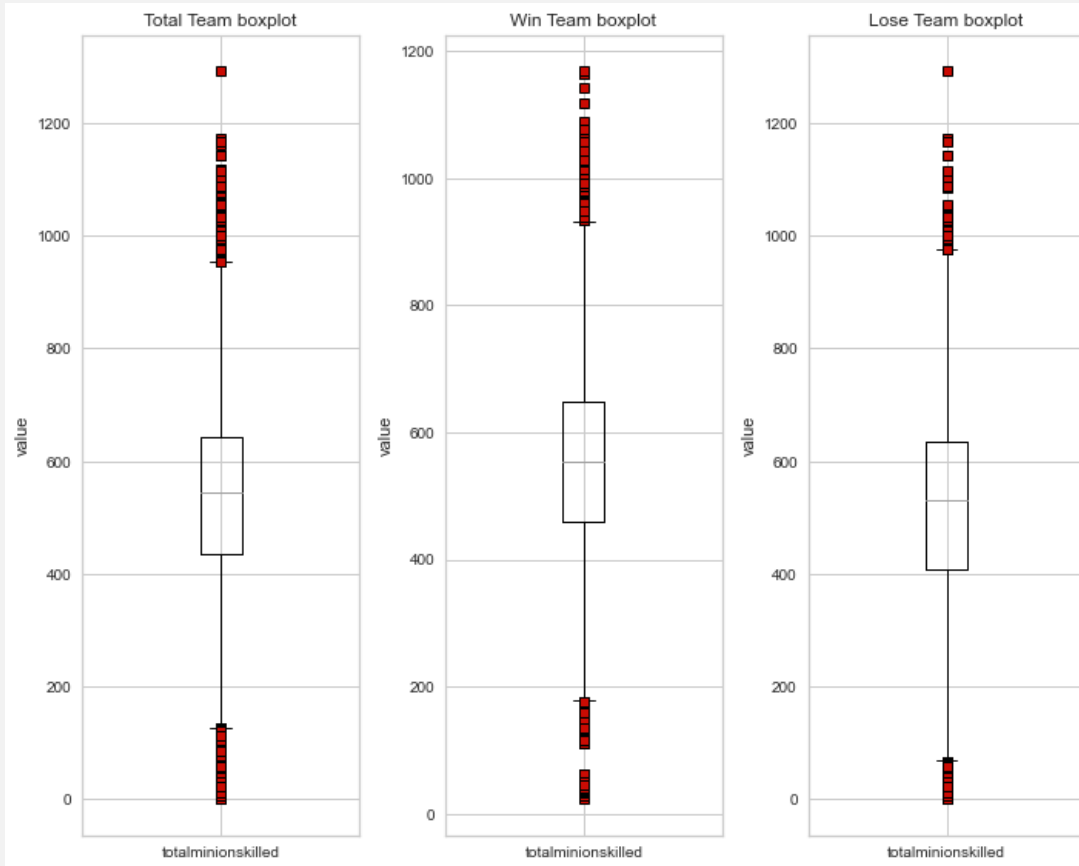


이긴팀, 진팀이 비슷한 평균과 분포  
를 나타냄



## 1) 전체 데이터 통계 확인

- 전체팀, 승팀, 패팀의 CS량 확인



-CS의 경우, 승팀이  
근소하게 평균값이  
높음

하지만, 거의 차이  
없음

## 1) 전체 데이터 통계 확인

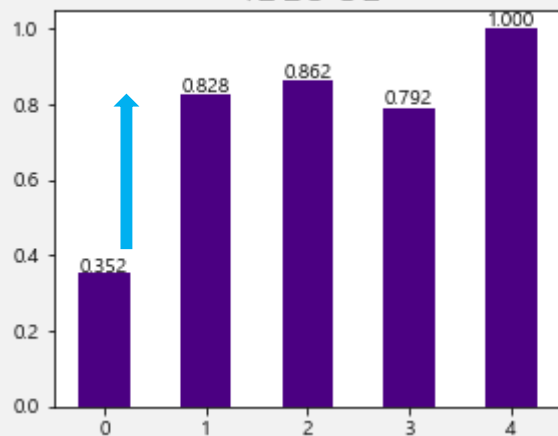
### - 문제 정의 :

승패팀의 여러 속성중, baronkills, dragonkills, inhibitorkills, riftheraldkills, towerkills 를 제외하고  
나머지 속성에서는 승패팀의 유의미한 차이가 보이지 않았음

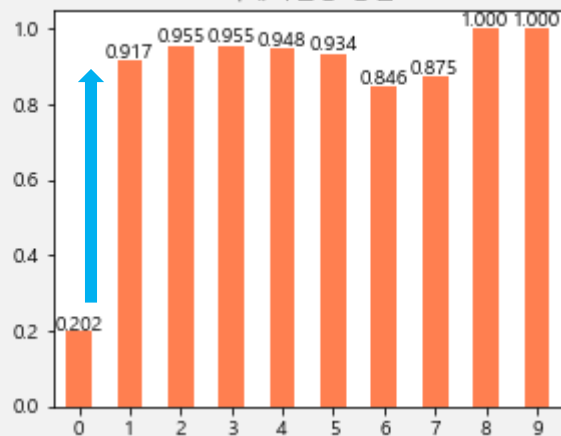
**그렇다면, baronkill과 dragonkill, inhibitorkills, riftheraldkills, towerkills가  
승리에 미치는 영향력은 어느정도 일까?**

# 1) baronkill과 dragonkill, inhibitorkills, riftheraldkills, towerkills가 승리에 미치는 영향력

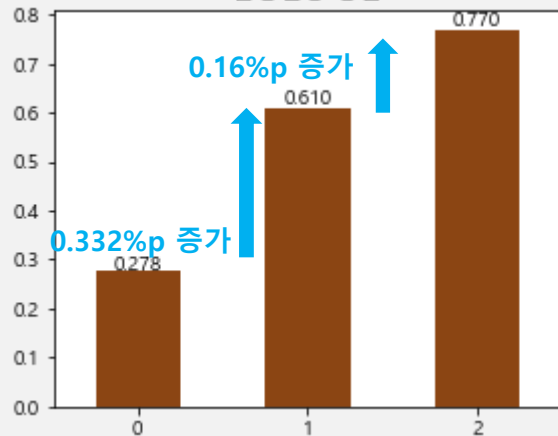
바론킬당 승률



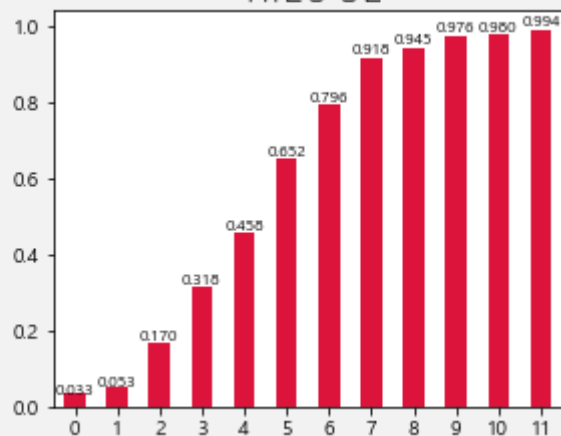
억제기킬당 승률



전령킬당 승률



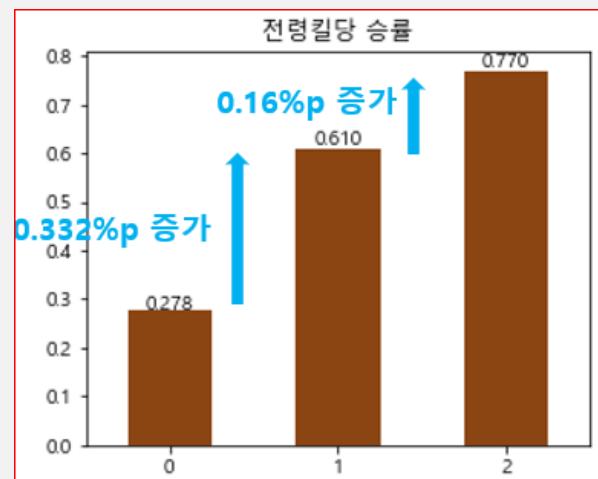
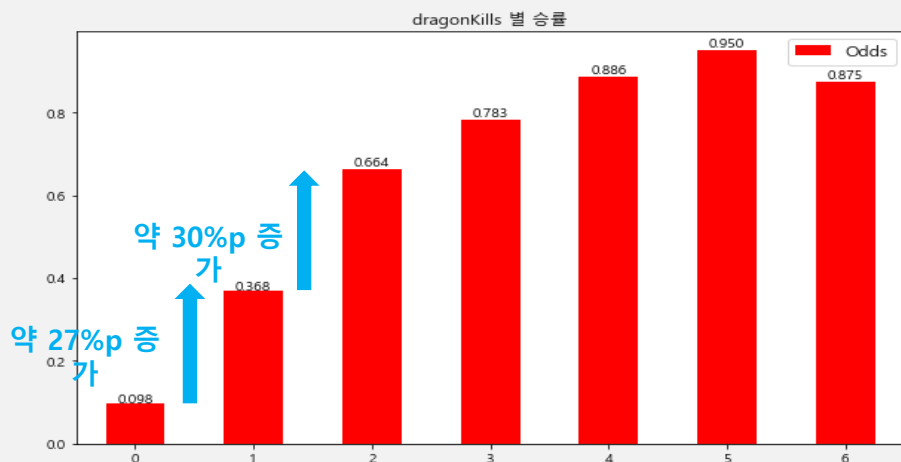
타워킬당 승률



- 바론1킬, 억제기1킬부터 승률이 눈에 띄게 높아짐  
(각각, 47%p, 71%p 상승)

- 타워킬, 전령킬과 승률은 정비례

# 1) baronkill과 dragonkill, inhibitorkills, riftheraldkills, towerkills가 승리에 미치는 영향력



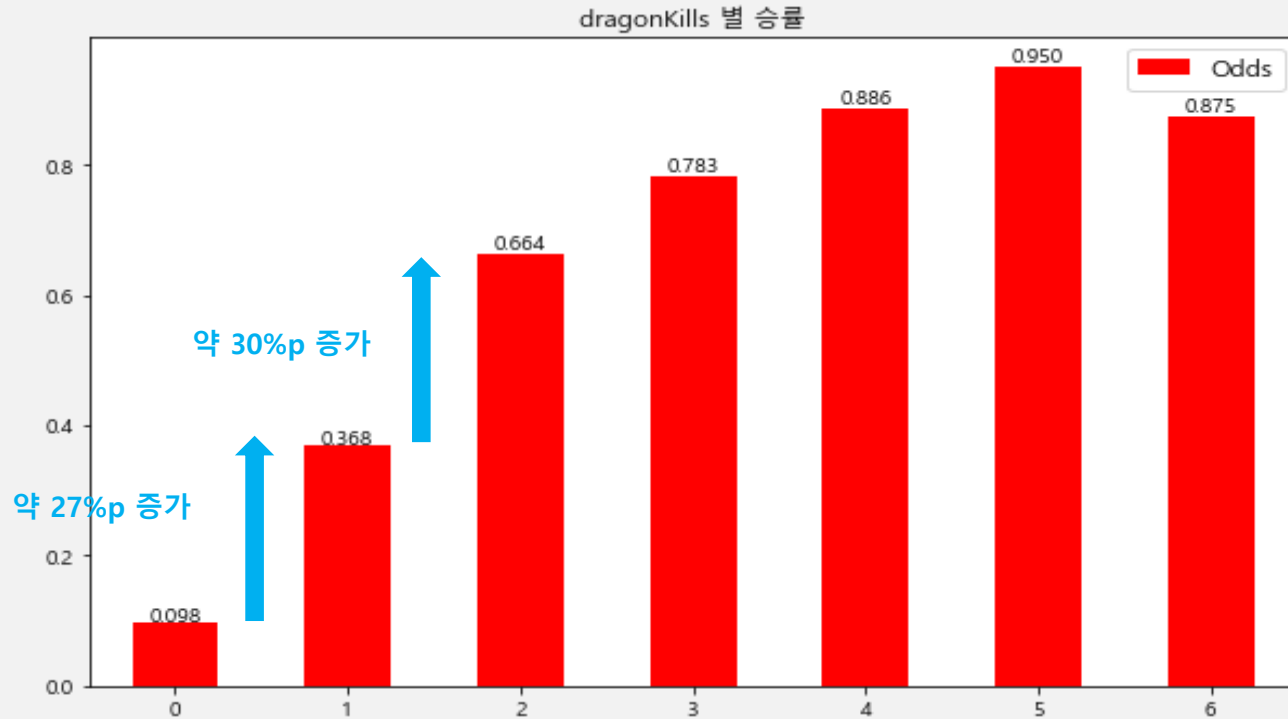
용보다 전령을 먹었을 때,  
승률이 높아보이지만,  
초반 평균 변화율은 용을 먹었을 때가 크다

전령 : 0 → 1 (+33%p)	/	1 → 2 (+16%p)	→	+24.5%p
용 : 0 → 1 (+27%p)	/	1 → 2 (+30%p)	→	+30%p

초반 게임의 변수창출 :  
용 > 전령

그렇다면, 최소 몇 개의 용을 먹어야 승리를 확신할 수 있을까?  
또한, 전령은 용의 대체재일까? 아니면, 어쩔수 없는 선택인걸까?

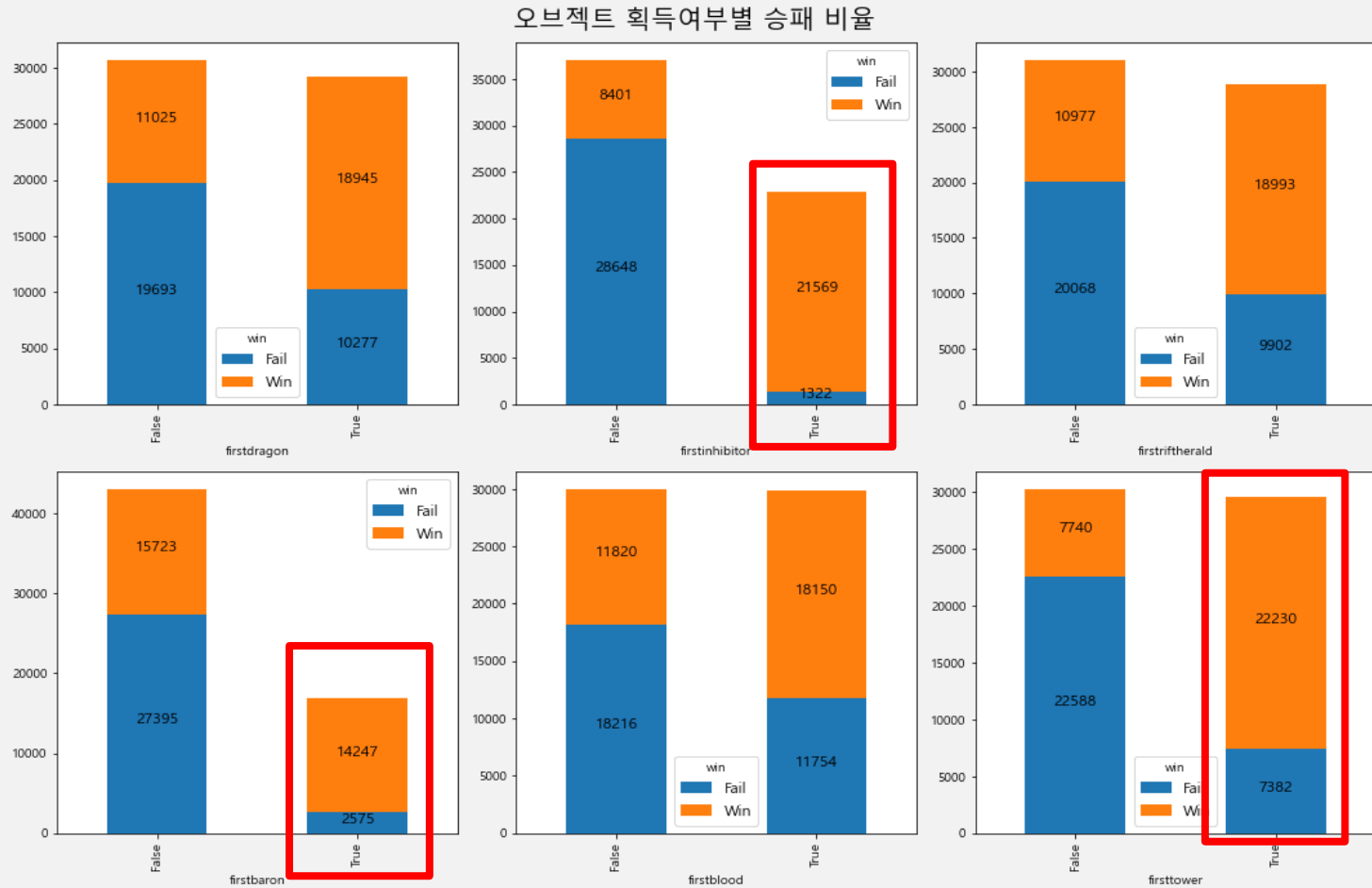
## 2) 게임에서 승리하기위해서 최소 몇 개이상의 용을 먹어야할까?



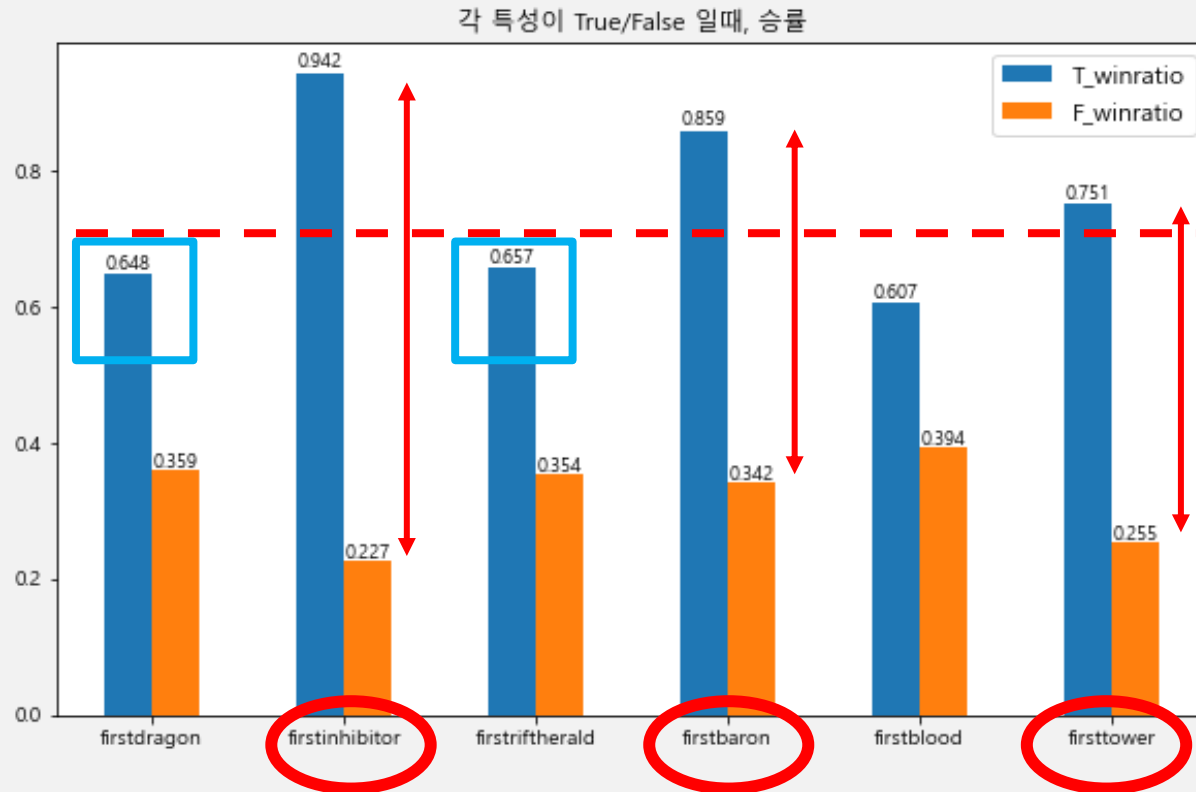
0->1, 1->2용의 승률에서 가장 크게 승률그래프가 높아지는 것을 확인  
3용부터 승률이 80%부근으로 상승

최소 **3용**부터 승리를 확신하는 것  
이 가능할것

### 3) 뽕 대신 닭 = 용 대신 전령? 전령의 영향력은 얼마나 될까?



### 3) 펑 대신 닭 = 용 대신 전령? 전령의 영향력은 얼마나 될까?



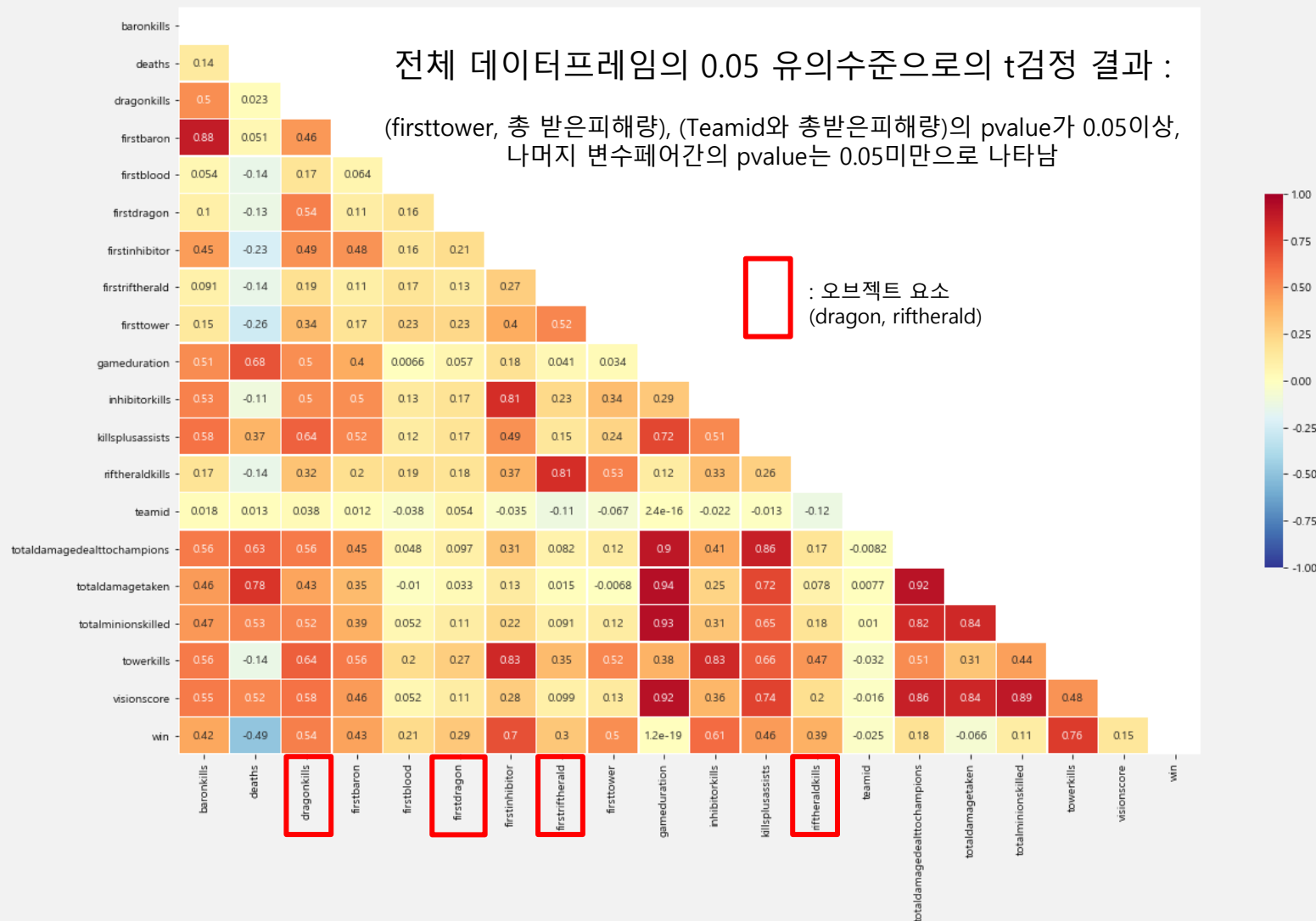
'첫 억제기', '첫 바론', '첫 타워'가 True일 때가 False일 때보다 승률이 월등히 높은 것을 확인

다만, '첫 용'과 '첫 전령'의 승률이 각각 0.648, 0.657로 근소한차이를 보이고 있음.

**첫 용 과 첫 전령이 경기에 미치는 영향이 비슷**



### 3) 전체 속성간의 상관분석



### 3) 전체 속성간의 상관분석 - 분석 결과 (용과 전령에 대하여)

Firstrifthermald – Win (0.3)

FirstDragon – Win (0.29)

Firstrifthermald – FirstTower (0.52)

FirstTower – Win (0.5)

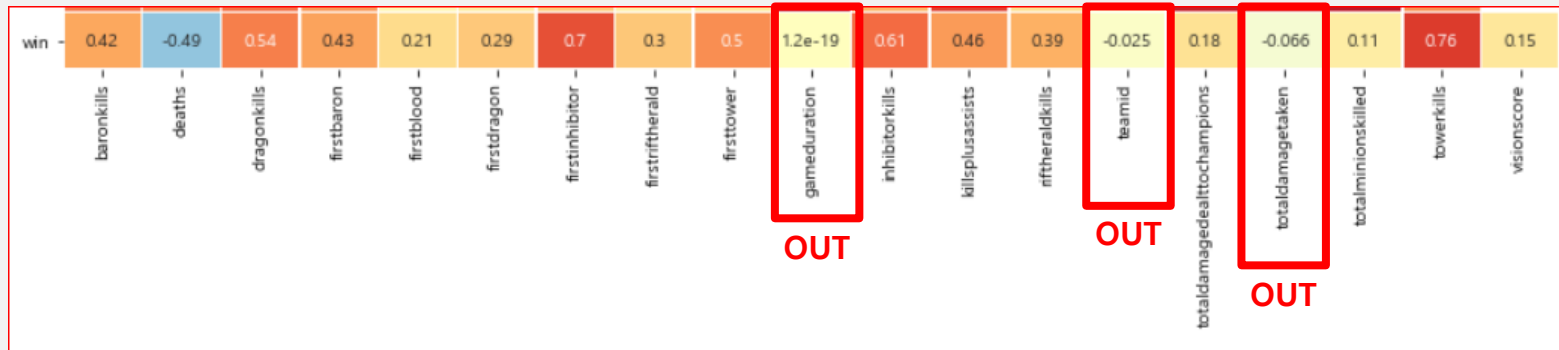
(첫용, 승리), (첫 전령, 승리)와의 상관관계 :  
0.3, 0.29로 비슷한 수준



전령의 전략적 사용은 직접적인 타워 또는 억제기 킬로 이어지기 때문에,  
용을 선택하는 것과는 다르게 운영적측면에서의 성과와 변수를 창출하는  
변수가 될 수 있을 것.

정글러가 용 대신 전령 이라는 관념에서 전령을 유의미하게  
사용할 수 있도록 라인상황, 적 정글 포지션 등을 파악하고  
라이너들과 합을 맞춘다면,  
용과 버금가는 혹은 용 이상의 운영적 효과를 얻을 수 있을  
것이라 생각

#### 4) 각 오브젝트 및 인게임 환경지표를 활용한 승리 예측 모델



Win 에 영향을 거의 미치지 못하는 속성 :

Teamid (블루진영/레드진영)  
받은총피해량  
게임시간

해당, 속성들을 제외하고 나머지 속성을 가지고 승리  
예측모델을 생성

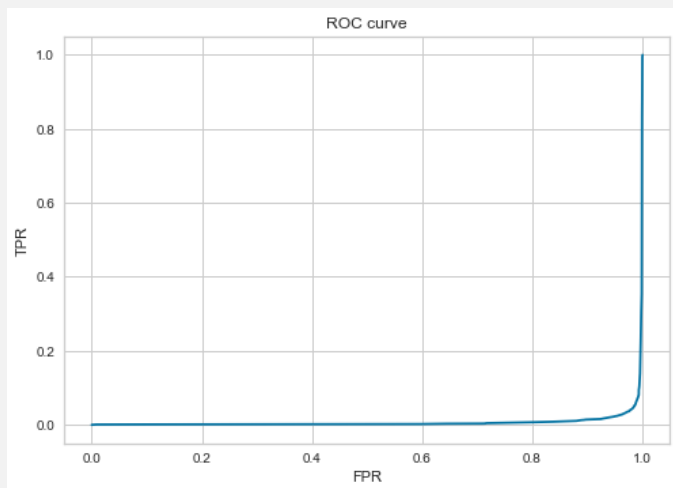
#### 4) 각 오브젝트 및 인게임 환경지표를 활용한 승리 예측 모델

Ex) Decision Tree

```
1 X,y=data.iloc[:, :-1], data['win']
2 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)
3
4 tree = DecisionTreeClassifier(max_depth=7, random_state=1)
5 model=tree.fit(X_train, y_train)
6 print('훈련셋 정확도 : ', model.score(X_train,y_train))
7 print('테스트셋 정확도 : ', model.score(X_test,y_test))
```

훈련셋 정확도 : 0.9744744744744744

테스트셋 정확도 : 0.9689133578022467



```
1 model.predict_proba(X_test)
```

```
array([[6.57894737e-04 9.99342105e-01],
       [9.88313856e-01 1.16861436e-02],
       [6.57894737e-04 9.99342105e-01],
       ...,
       [6.57894737e-04 9.99342105e-01],
       [9.88313856e-01 1.16861436e-02],
       [9.89583333e-01 1.04166667e-02]])
```

Lose일 확률

Win일 확률

#### 4) 각 오브젝트 및 인게임 환경지표를 활용한 승리 예측 모델

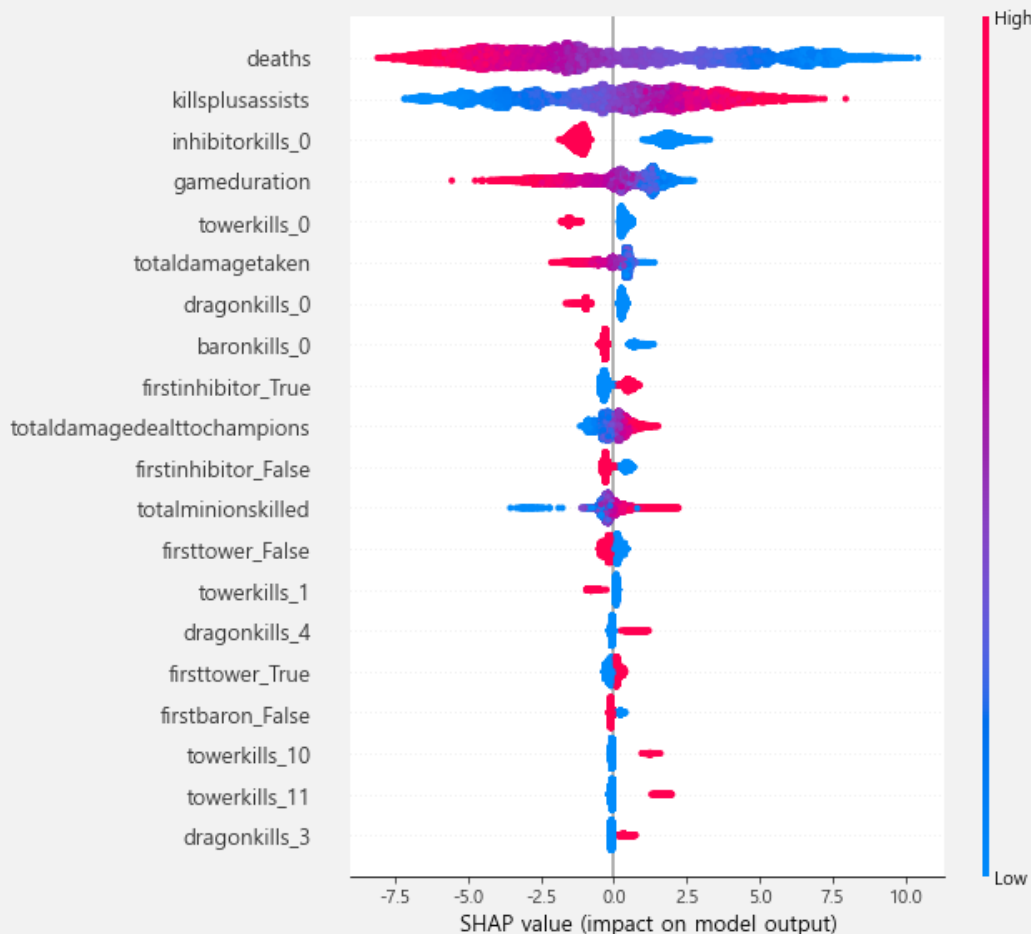
Pycaret을 이용해, Win or Lose를 예측하는 머신러닝 모델 중,  
Test Accuracy가 높았던 Top10 모델 생성

```
1 best3_2 = compare_models(n_select = 3)
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
0	CatBoost Classifier	0.9771	0.9977	0.9803	0.9740	0.9771	0.9541	0.9541	16.5359
1	Extreme Gradient Boosting	0.9765	0.9976	0.9798	0.9735	0.9766	0.9531	0.9531	1.9516
2	Light Gradient Boosting Machine	0.9757	0.9976	0.9801	0.9715	0.9758	0.9513	0.9514	0.4389
3	Gradient Boosting Classifier	0.9741	0.9969	0.9758	0.9724	0.9741	0.9481	0.9481	5.4143
4	Ada Boost Classifier	0.9734	0.9969	0.9749	0.9721	0.9735	0.9469	0.9469	1.5918
5	Ridge Classifier	0.9712	0.0000	0.9709	0.9715	0.9712	0.9423	0.9423	0.0518
6	Linear Discriminant Analysis	0.9711	0.9957	0.9706	0.9715	0.9711	0.9422	0.9422	0.2219
7	Random Forest Classifier	0.9681	0.9934	0.9703	0.9661	0.9682	0.9363	0.9363	0.1185
8	Extra Trees Classifier	0.9653	0.9945	0.9736	0.9578	0.9656	0.9307	0.9308	0.7821
9	Decision Tree Classifier	0.9617	0.9617	0.9613	0.9620	0.9617	0.9234	0.9234	0.2127
10	Logistic Regression	0.9492	0.9867	0.9502	0.9484	0.9492	0.8983	0.8984	0.3340

#### 4) 각 오브젝트 및 인게임 환경지표를 활용한 승리 예측 모델

가장 모델 정확도가 높았던 'catboost Classifier'의 SHAP value 확인 :



Dragon의 경우,  
4kills부터 그래프에서 특성중요도와  
함께 shap value가 높아짐.

#### 4) 각 오브젝트 및 인게임 환경지표를 활용한 승리 예측 모델

Top10 모델중, Top3의 모델을 blending하여 최적의 모델 생성

```
1 blended_2 = blend_models(estimator_list = best3_2, fold = 5)
```

평가셋 :

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.9784	0.0000	0.9830	0.9740	0.9785	0.9568	0.9568
1	0.9736	0.0000	0.9813	0.9665	0.9738	0.9472	0.9473
2	0.9767	0.0000	0.9779	0.9755	0.9767	0.9534	0.9534
3	0.9796	0.0000	0.9803	0.9789	0.9796	0.9591	0.9591
4	0.9768	0.0000	0.9813	0.9727	0.9769	0.9537	0.9537
Mean	0.9770	0.0000	0.9807	0.9735	0.9771	0.9540	0.9541
SD	0.0020	0.0000	0.0017	0.0041	0.0020	0.0040	0.0040

테스트셋 :

```
predictions = predict_model(final_model)
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Voting Classifier	0.9875	0	0.989	0.9861	0.9875	0.9751	0.9751

\* 프로 경기에서, 선수들의 실력을 데이터 기반으로 정량적인 측정한다고 했을 때,  
어떤 지표들이 필요하며 그 지표를 만들기 위해서 필요한 데이터가 무엇이 있을까?

## - 챔프 상성대비 CS, 골드, 경험치지표

예를들어, 애쉬 vs 케이틀린 이라고 한다면,  
애쉬와 케이틀린이 바텀 원딜로 붙었을때의  
평균 CS, 골드, 경험치지표들의 평균을 기준으로  
+,-지표로 바꾸는 것.



서폿을 제외한 라인의 선수들에게  
보다 정확한 라인전의 기준점을  
제시할 수 있음

## - 챔프 특성 대비 생존율

KDA

---

(총 입은 데미지량 / 챔피언에게가한 총데미지량)

OR

챔피언에게 가한 데미지량

---

15분마다 (체력+방어력+마법저항력)의 평균



라인별 챔프 특성을 반영한 생존율을  
알 수 있음



프로젝트 명	서울시 강남구 스쿨존 교통사고 위험지역 예측모델 개발 프로젝트	프로젝트 명	자연어처리 기반의 KOSPI 및 YG엔터주가 예측 프로젝트	프로젝트 명	리그오브레전드 인게임 오브젝트 와 승률간 데이터분석
수행기간	2020.08 ~ 2020.09	수행기간	2020.09 ~ 2020.11	수행기간	2020.11

## 해당 프로젝트들을 통해 얻게 된 역량

1. Python을 이용한 시각화,데이터 분석, 모델링 등 프로그래밍 역량
2. 데이터 크롤링 및 텍스트 데이터 전처리 역량 강화
3. 시계열 데이터 모델링, 머신러닝 및 딥러닝 등 AI관련 지식
4. Github 등 협업 프로젝트 플랫폼 경험

# 비전 및 핵심역량

데이터를 기반으로  
유의미한 정보를 제공하며 가치를 발견하는 개발자

## 머신러닝에 대한 이해

- 인공지능 교육훈련을 통한 핵심역량 배양  
(혁신성장 청년인재 사업 중 실무중심의 인공지능 개발자 3기 과정 11월 수료예정)
- MOOC 등을 활용한 온라인 교육 참여
- 서울시와 ICT콤플렉스가 함께하는 2020 ICT콰이엇 공모전 장려상 입상

## 유관 프로젝트 수행 경험

- 서울시 강남구 스킵존 교통사고 위험 지역 예측 프로젝트 수행(2020)
- 자연어처리 기반의 KOSPI 및 YG엔터 주가 예측 프로젝트 수행(2020)
- 리그오브레전드 인게임 오브젝트와 승률간 데이터분석 (2020)

## 다양한 개발 툴, Git 경험

- Pytorch, Tensorflow, Bert모델 경험
- Jupyter notebook, Google colab을 활용한 Python 프로그래밍
- Pycharm, Javascript를 이용한 웹 개발
- Github 주소 :  
<https://github.com/kimseojeong6533>

**Thank You**