**인공지능 응용 과제**

**축구선수의 유망 여부 예측 분류 모델 개발 보고서**

201845818 안한주

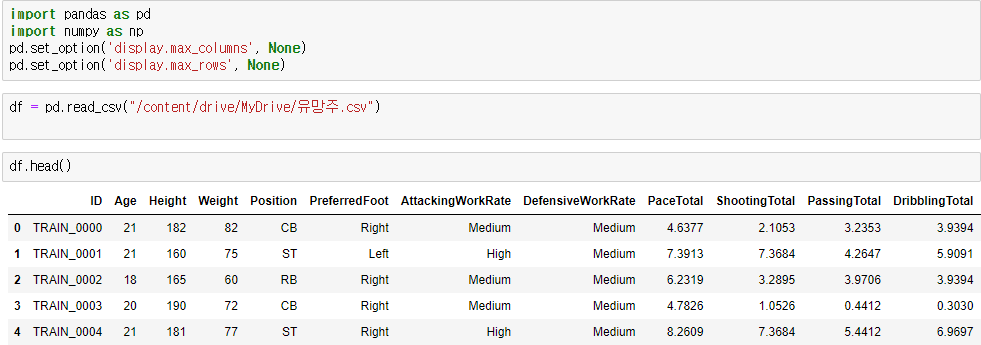
**목차**

1. **데이터 소개**
2. **데이터 전처리**
3. **기계학습 및 모델 개발 및 성능 분석**
4. **결론(느낀점)**
5. **데이터 소개**

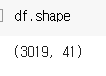
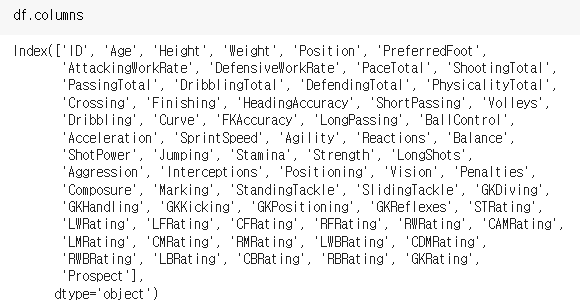
AI 해커톤 플랫폼 DACON의 최근 대회 중 축구선수의 유망 여부 예측 AI 경진대회의 데이터를 활용해 선수 정보를 활용한 AI분류 모델을 만들어 보았다.

툴은 파이썬 Anaconda를 설치해 Vscode에서 sklearn의 classsifier모델들을 사용해 머신러닝 모델들을 활용하였고, 딥러닝은 구글의 colab환경에서 tensorflow와 keras를 사용해 진행하였다.

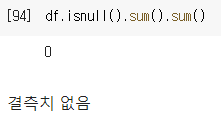
맨 처음 데이터를 프레임화 하기 위해 pandas와 numpy 라이브러리를 이용해 데이터 프레임을 확인해보았다.



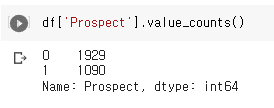
데이터의 구성을 보았고, 어떤 속성이 있는지 확인해 보았다.



데이터는 3019행, 41열로 구성되어 있었고, 속성은 다음과 같았다



결측치는 존재하지 않았음



target이 되는 유망 여부 데이터로 편향되지 않고 고루 분포 되어있음

1. **데이터 전처리**

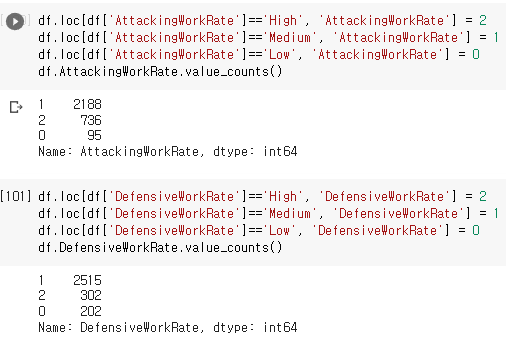
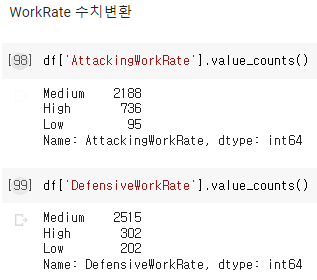
현재 데이터에 수치값이 아닌 문자열 데이터가 있는데, 선수ID, 포지션, 주로 사용하는 발, 공격비율, 수비비율이다.

선수ID는 삭제,

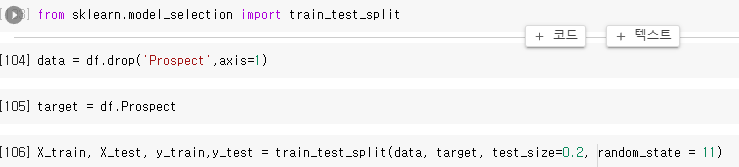
포지션 또한 전체 능력치로 유망 여부를 판단할 것이므로 삭제,

주발 또한 유망 여부에 근거되지 않으므로 삭제,

공격 비율과 수비 비율은 value값을 보았을 때 Medium, high, low 3개의 값이 분포되어 있어 순서대로 2,1,0값으로 치환하였다.



이후 분류 모델에서 자주 사용하는 Decision Tree 모델을 사용해 모델을 적용해 보았다.

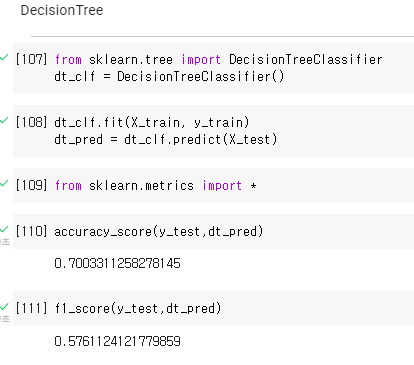


먼저 train데이터와 test데이터를 8:2 비율로 나누었고 data에는 target값인 prospect를 제외한 값, target에는 prospect값을 넣어 진행하였다.

이후 Decision Tree모델 적용

1. **기계학습 및 모델 개발 및 성능 분석**

**1)DecisionTree**



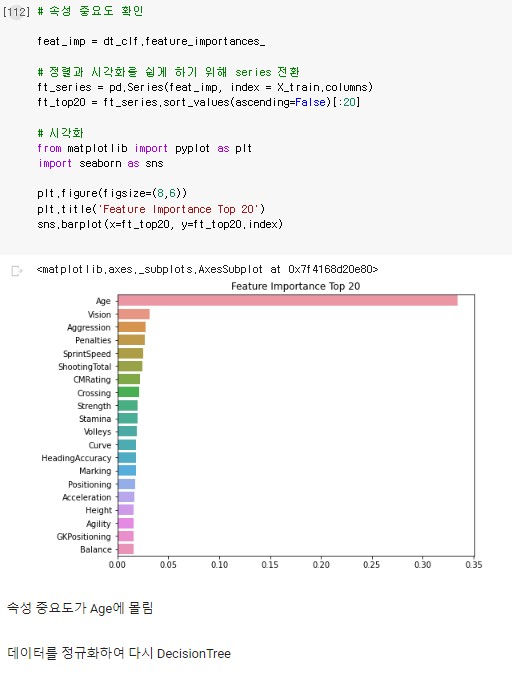
모델이름을 dt\_clf로 하였고, .fit을 통해 데이터를 학습시켰다.

이후 predict함수를 통해 test를 통해 예측하였다.

다음, 모델의 정확도와 F1score를 확인하였을 때, 정확도는 70%이상, F1score는 0.57로 나타났다.

F1스코어가 높지 않아 적합하지 않다고 생각하였고, 모델의 각 속성 중요도를 확인해 보았다.

속성 중요도



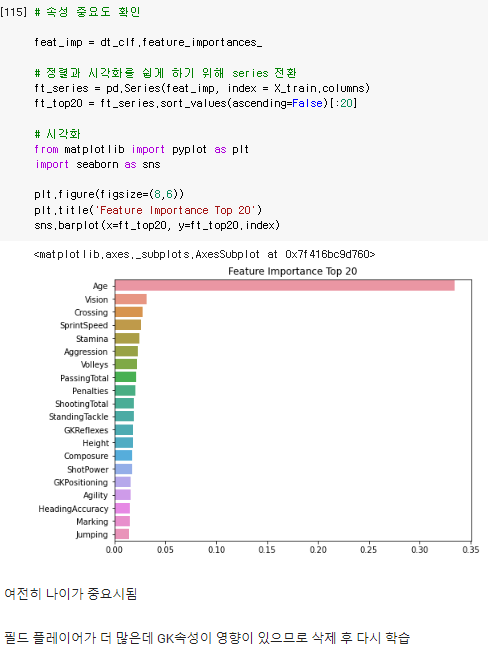
다음을 보면 대부분의 중요도가 Age속성에 높게 잡혀있었다.

데이터를 정규화하여 다시 모델을 학습시켜 보았다.



Sklearn의 StandardScaler를 사용해 정규화를 진행하였지만 오히려 감소하는 모습이 보였다.

다시 속성 중요도를 보았을 때



Position의 value\_counts를 보면

테이블이(가) 표시된 사진

자동 생성된 설명

골키퍼의 비율이 필드플레이어보다 적은데, 중요도 목록에 GK가 높이 있으므로 다시 GK속성을 삭제하고 학습을 진행하였다.

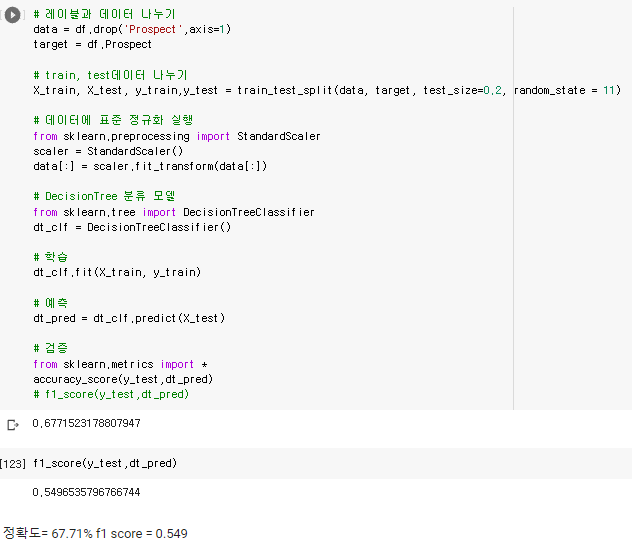


하지만 오히려 감소하는 모습을 볼 수 있었다.

속성 중 포지션별 Rating속성을 제거하고 다시 학습을 진행해 보았다.

텍스트이(가) 표시된 사진

자동 생성된 설명



오히려 감소하는 모습을 볼 수 있었다

현재까지 처리된 데이터를 사용해 다른 모델을 사용해 보았다.

**2)RandomForest**

앙상블 모델인 랜덤 포레스트를 사용해 보았다

**테이블이(가) 표시된 사진

자동 생성된 설명**

RandomForest모델로 바꾸자 정확도와 f1score가 증가한 모습을 볼 수 있었다.

**3) SVM(support vector machine)**

이진 분류 모델에서 자주 사용되는 SVM모델을 사용해 보았다.



SVM의 하이퍼 파라미터중 kernel 옵션을 linear로 하여 선형 분류로 하였다.

정확도가 약 80%정도로 굉장히 높게 나왔음을 알 수 있다. F1스코어 역시 가장 높았다.

**4) LogisticRegression**

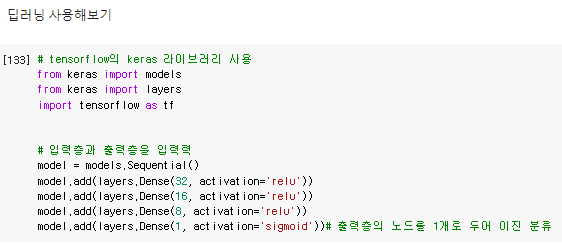
이진 분류 모델 중 로지스틱 회귀 모델을 사용해 보았다.



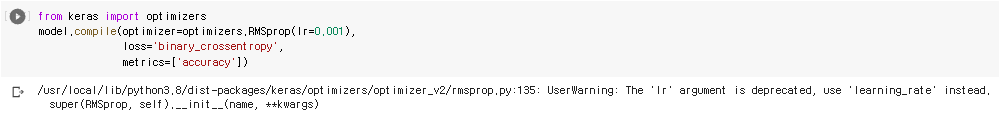
SVM모델만큼은 아니었지만 정확도가 높았고, f1score도 다른 모델들 보다 높았다.

이후 딥러닝도 한번 사용해 보았다.

개인 노트북에 이전에 anaconda를 설치하였지만, 계속 tenorflow와 keras 라이브러리를 찾을 수 없다는 에러가 발생해 따로 환경설정을 하지 않아도 되는 colab을 사용해 진행하였다.



딥러닝 프레임 워크 중 사용해 보았던 tensorflow와 keras를 사용해 먼저 시퀀셜 model을 호출해 4개의 층을 구성해 보았다. 입력층은 32개, 활성화 함수는 relu함수를 사용하였고, 출력층에는 노드를 하나로 주어 이진 분류가 가능하게 하였고, 이진 분류 시에는 sigmoid함수를 활용해 사용한다.



이후 모델의 옵티마이저를 모멘텀을 가진 rmsprop으로 두었고 손실함수는 binary\_crossentropy를 사용하고, 정확도를 확인해 보았다.

텍스트이(가) 표시된 사진

자동 생성된 설명

모델에는 numpy값이 들어가지 않아 tensorflow의 tensor로 변환하여 사용했다.

테이블이(가) 표시된 사진

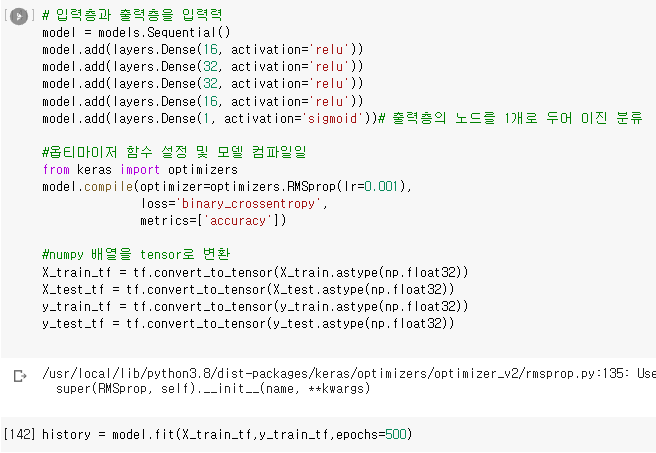
자동 생성된 설명

총 epoch를 500번 돌렸고, batch size는 데이터양이 많다고 생각하지 않아 따로 지정하지 않고 디폴트 값을 사용하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

500번의 epoch이후 정확도와 loss율이다. loss율이 너무 높아 은닉층을 수정해 보았다.



중간 은닉층을 추가하고 노드 개수를 변화시켰다.

테이블이(가) 표시된 사진

자동 생성된 설명

정확도가 상승하였지만, 높은 loss율을 가지고 있다.

**성능분석**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ML알고리즘 | DecisionTree | RandomForest | SVM | LogisticRegression |
| Accuracy | 0.677152 | 0.776490 | 0.799668 | 0.786423 |
| F1 score | 0.549653 | 0.665012 | 0.702702 | 0.679900 |

딥러닝 – Accuracy = 0.7723, Loss=0.4907

SVM알고리즘에서 가장 좋은 결과를 얻었다.

1. **결론(느낀점)**

같은 데이터여도 어떻게 전처리를 진행하고, 어떤 모델을 사용하는지에 따라 다른 결과가 나오고, 같은 모델도 어떤 방식을 사용하느냐에 따라, 지도학습인지 비지도학습인지, 분류인지 회귀인지, 분류도 이진 분류인지, 다중 분류인지에 따라 다 성능이 다르기 때문에 여러 모델을 사용해 방법방 찾아보는 것이 중요하다고 생각했다. 또한 딥러닝이라고 모두 성능이 좋지는 않다는 것을 확인하였다. 모델을 만들었다고 바로 사용할 수 없고, 정확도 뿐 만 아닌f1 score나 재현율, 정밀도 등을 통해 어떤 알고리즘을 선택해야 하는지가 중요하다고 생각했다.

아쉬운 점)

다른 공모전에서도 느꼈지만 항상 데이터의 전처리에서 아쉬운 점이 든다. 데이터의 상관관계를 보고, 전진선택법이나, 후진 소거법을 이용해 변수를 선택했다면 어땠을까 하는 생각이 들고 또한 데이터 특성 상 포지션 별로 나누어 모델을 개발하였다면 좀더 성능이 뛰어난 모델을 개발할 수 있었을 것이란 생각이 든다. 또한 모델의 하이퍼 파라미터를 좀 더 수정해 보면 어땠을까 라는 아쉬움도 있다.

인공지능은 이제 굉장히 밀접한 분야가 되었고, 어떤 분야에 어떻게 접목할지가 중요한 것 같다. 또한 데이터의 중요성을 다시 한번 느꼈고, 데이터의 처리가 정말 중요하다고 생각이 들었다.