* **Part1**
  1. **프로젝트의 목표**
     + 풀고자 하는 문제가 명확하게 제시되었다.
     + 이 문제를 푸는 것이 왜 가치가 있는지 명확히 제시되었다.
  2. **데이터셋 선정**
     + 부트캠프 내 데이터셋이 아닌 다른 데이터셋을 활용하였다.
     + 데이터셋을 선정한 근거가 드러나 있다.
  3. **머신러닝 문제 정의**
     + 부트캠프 내 문제가 아닌 다른 문제를 정의하고 데이터셋으로부터 문제(회귀 or 분류) 정의 과정을 제시하였다.
     + 문제 정의에 따른 적절한 타겟을 설정하였다.
     + 문제를 풀기위해 너무 뻔하지 않은 적절한 가설을 잘 세웠다.
  4. **EDA&전처리**
     + 데이터 분석 과정과 결과 설명이 발표에 나타나 있다.
     + 데이터를 살펴보고 전처리를 한 과정을 설명하였다.
     + EDA를 통해 가설의 타당성을 설명하였다.
* **Part 2**
  1. 모델 학습 및 검증
     + 모델링 이전에 미리 baseline 모델을 선정하였다.
     + 문제 정의에서 언급한 회귀 / 분류 문제에 맞는 모델을 선택하였다.
     + [+] 선택한 모델의 작동 원리 또는 주의할 점 등의 설명을 포함하였다.
     + 회귀/분류 문제에 따른 적절한 평가지표를 선택하였다.
     + 2개 이상의 평가지표를 사용하였다.
     + [+] 평가지표의 해석법이나 사용 이유 등을 발표에 포함하였다.
     + 교차검증이나 hold-out을 사용해 데이터셋을 분리하고 모델의 성능 비교 및 향상을 위해 검증 데이터셋을 활용했다.
     + 모델 최적화를 위한 하이퍼파라미터 튜닝을 진행하였다.
     + [+] 비교하고자 하는 하이퍼파라미터의 역할에 대해 이해하고 발표에서 설명하였다.
     + 모델 결과로서 Test Score(일반화 성능)를 제시하여 모델을 평가하였다.
     + 테스트셋은 모델의 일반화 성능을 평가하기 위해 한 번만 사용했다.
  2. 모델 해석
     + PDP/SHAP/Feature Importance 등을 활용하여 모델 작동을 설명하였다.
     + 모델 학습 결과에 대한 논리적 해석이 드러나 있다. (가설 해소 혹은 문제 해결 등)
     + [+] 발표의 마지막에 모델 학습 결과에 대한 한계점과 추후 개선사항을 제시하였다.
* **Part 3**
  1. 제출한 발표 영상의 길이가 5분 이상 10분 이하이다.
  2. 준수한 언어적 전달력 및 영상 상태를 유지할 수 있는 환경에서 영상을 촬영하였다.
     + 비즈니스 상황에 맞는 발표 매너를 준수하였다.
     + 직관적이고 전달력이 높은 발표 자료를 사용하였다.
     + 내용 전달이 가능한 상태의 영상을 제출하였다.

**제출 시 유의사항 체크**

* 5~10분 이내로 동영상을 촬영하여 제출하였다.
* 제출 파일 이름을 AI\_00\_김코딩\_section2.mp4로 설정하였다.
* UrClass 제출폼(드롭박스)에 **동영상 파일만** 제출하였다. (코드 파일 아님)
* 프로젝트에 사용한 파일을 나의 GitHub Repository(public)에 업로드하고 제출폼에 링크를 제출하였다.

Section2\_project

안녕하세요. 코드스테이츠 AI 17기 김영훈입니다. 발표하기에 앞서 여러분은 런던 여행을 해 본적이 있으신가요? 저는 19년도에 여행했었는데, 날씨가 흐려서 많이 안타까운 기억이 있습니다. 런던은 맑은 날보다 흐린 날이 더 많다고 합니다. 그래서 저는 런던 여행하는 분들이나 거주자들의 편의를 위해 런던의 날씨 데이터를 바탕으로 비가 내릴지 예측하는 모델을 만들었습니다.

목차를 살펴보겠습니다. 첫번째 데이터 EDA와 전처리, 두번째 모델링 및 해석 세번째 향후 계획 순으로 발표하겠습니다.

첫번째로 데이터 EDA 및 전처리입니다. 데이터 소개, 전처리, feature engineer, eda 순으로 발표하겠습니다. 먼저, 데이터 소개를 하겠습니다.

데이터는 런던의 1979년도부터 2020년도까지의 15341개의 데이터입니다.

데이터 특성으로는 date 날짜, cloud\_cover 구름이 하늘을 덮은 정도를 oktas 단위로 표현한 값이구요. sunshine 1시간 동안 햇빛을 측정한 값, global\_radiation 1제곱미터당 복사조도측정값으로 단위는 와트퍼미터제곱을 사용합니다. Max, mean, min temp 최고 평균 최저온도입니다. 타겟 특성인 precipitation 강수량, pressure 대기압이 있습니다.

다음으로는 데이터 전처리입니다. 중복치를 먼저 확인했습니다. 중복치가 존재하지 않으며 중복되는 날짜도 없는 것을 확인할 수 있습니다.

다음으로는 결측치입니다. 예측값인 강수량은 정확한 예측을 위해 결측치를 제거했습니다.

Cloud\_cover, global\_radiation, min\_temp는 연속적으로 결측치가 존재하거나, 앞뒤로 차이가 큰 데이터가 존재하므로 총 40개 데이터를 제거했습니다.

Max\_temp, pressure 특성은 연속 결측치가 존재하지만, 앞뒤로 차이가 크게 없으므로 앞의 데이터로 채우기로 결정했습니다.

Mean\_temp는 max\_temp와 min\_temp의 평균이므로 평균으로 결측치를 채우기로 결정했습니다.

다음으로는 feature engineer입니다. 계절별로 강우에 영향을 미친다고 생각하여 조금 더 넓은 범위인 월별 month 특성을 추가하고, 모델 예측값을 설정하기 위해 비가 오면 즉 강수량이 0이 아니면 1, 비가 안오면 즉 강수량이 0이면 0으로 추가했습니다.

다음은 EDA입니다. 특성별 상관관계를 히트맵으로 표현해봤습니다. 히트맵을 통해 타겟관의 상관관계에서 크게 두가지 인사이트를 도출했습니다. 선샤인, 글로벌 레디에이션, 프레셔 값이 클수록 비가 올 확률이 낮고, 즉 타겟과 음의 상관관계를 가지고, 클라우드 코버 값이 클수록 비가 올 확률이 높다 즉 타겟과 양의 상관관계를 가지는 것을 알 수 있습니다.

추가로 계절별(월별) 강수 확률에 영향을 끼친다고 생각하여 month 특성을 추가했으나 상관관계에서는 작은 값을 나타내는 것을 볼 수 있는데요. 그래서 월별 강수량 및 확률을 체크해봤습니다.

다음은 1979년도부터 1999년도 월별 강수량입니다. 대체적으로 6-8월에 강수량이 낮은 것을 볼 수 있습니다.

1979년도부터 1999년도 월별 강수 확률입니다. 앞서 대체적으로 강수량이 6-8월에 낮은 것을 볼 수 있었는데, 강수량이 낮다고 비가 올 확률이 낮지 않다는 것을 볼 수 있습니다.

다음은 모델링 및 해석입니다. 모델 및 평가지표 설정, 데이터 분리, 기준모델 설정, 모델링 검증 평가, 모델 해석 순으로 발표하겠습니다.

모델 및 평가지표 설정입니다. 예측값을 비가 올지 안 올지 이진분류로 설정했기 때문에, 트리기반 모델인 랜덤포레스트와 xgboost 모델을 사용했습니다.

평가지표로는 단순 예측 정확도 측정하기 위한 지표인 정확도, 비가 안왔는데 왔다고 예측하는 타입2 에러보다 비가 왔는데 안왔다고 예측하는 타입1 에러가 더 중요하다고 생각했기 때문에 재현율, 그리고 비가 올 확률을 비가 오지 않을 확률보다 확률값 자체를 높게 예측하는 것 또한 중요하다고 판단해서 roc\_auc 지표 총 3개를 선정했습니다.

다음은 데이터 분리입니다. 먼저 데이터 누수방지를 위해 강수량 특성을 제거했습니다. 비교적 최근 4년 데이터인 2017년부터 2020년도 데이터를 테스트 셋으로 설정했습니다. 훈련데이터셋과 검증데이터셋을 8:2 비율로 나눴습니다.

기준모델 설정입니다. 기준모델은 최빈값으로 설정했고 기준모델의 정확도는 0.52가 나온 것을 확인할 수 있습니다.

먼저 randomforest 모델입니다. Randomforest 모델은 배깅방식을 통해 독립적으로 학습한 여러 개의 결정트리 모델들이 결합하여 하나의 강력한 모델을 생성하는 앙상블 기법 중 하나입니다. Randomforest 모델로 학습하고 성능을 평가해봤습니다. 검증데이터 셋에서 정확도 0.745, 재현율 0.699, auc 지표 0.826이 나온 것을 볼 수 있습니다. 여기서 저는 하이퍼파라미터로 결정 트리의 깊이를 설정하는 max\_depth, 기본 모델의 수를 결정하는 n\_estimators를 조정하여 gridsearch를 통해 최적의 하이퍼파라미터를 찾았습니다. 하이퍼파라미터 조정 후 성능입니다. 훈련 데이터를 보고 과적합이 된 줄 알았으나 검증 데이터 역시 모든 지표가 조금씩 오른 것을 확인할 수 있습니다.

다음으로는 xgboost 모델입니다. Xgboost는 병렬적으로 결정 트리 모델을 처리하여 앞서 학습한 모델에 학습이 덜 된 부분을 보충해나가 하나의 강력한 모델을 생성하는 앙상블 기법 중 하나입니다. Xgboost 모델로 학습하고 성능을 평가해봤습니다. 검증데이터 셋에서 정확도 0.747, 재현율 0.717, auc 지표 0.822가 나온 것을 볼 수 있습니다. 여기서 하이퍼파라미터로 max\_depth, 관측치에 가중치의 최소 합인 min\_child\_weight, 트리 생성에 필요한 피처의 샘플링 비율인 colsample\_bytree, 학습단계별로 가중치를 얼마나 줄지의 learning\_rate를 조정하여 girdsearch를 통해 최적의 하이퍼파라미터를 찾았습니다. 하이퍼파라미터 조정 후 성능입니다. 훈련데이터의 점수가 떨어진 것을 보아 과적합을 피하고 검증데이터 셋의 지표를 올린 것을 확인할 수 있습니다.

두 가지 모델에 마지막으로 테스트 셋을 통해 성능을 평가했습니다. 모든 지표에서 xgboost가 조금 더 높은 것을 확인할 수 있습니다. 따라서 xgboost를 예측모델로 결정했습니다.

다음으로는 선택한 모델 xgboost의 모델 해석입니다. Feature importances를 통해 xgboost 모델에 특성 중요도를 나타냈습니다. Pressure, sunshine Cloud\_cover 특성 순으로 모델 예측값에 가장 많은 영향을 미치는 것을 볼 수 있습니다.

다음은 각 특성별로 노이즈를 주어 예측값에 얼마나 영향을 주는지를 통해 중요도를 나타내는 perputation importances를 통해 특성 중요도를 나타냈습니다. 역시, pressure sunshine 특성이 모델 예측값에 가장 많은 영향을 미치는 것을 알 수 있습니다. 일반적으로 구름이 하늘을 많이 덮으면 비가 올 것으로 예측을 하지만 해당 특성에 노이즈를 주어도 예측값에 영향을 덜 미치는 것을 볼 수 있습니다. 이것을 통해 구름의 유무 정도만 예측값에 영향을 미치고 수치값은 크게 영향이 없는 것으로 판단했습니다.

마지막 향후 계획을 발표하고 마무리하겠습니다. 런던뿐만 아니라 서울 오타와 같이 다양한 도시의 예측과 비가 올 확률을 예측하는 선형회귀모델도 만들어볼 계획입니다. 이것으로 발표를 마무리하겠습니다. 감사합니다.