2023 날씨 빅데이터 콘테스트 공모

수치모델 앙상블을 활용한 강수량 예측

1. 공모 개요

공모 분야	주제1	**접수번호**	240484
팀명	위닝	**소속**	부산대학교
팀 구성원	김민정(DA/BE-flask), 배지현(FE/BE-springboot, DB)		
주제	수치모델 앙상블을 활용한 강수량 예측		
분석도구	R(주요 활용 패키지: random forest, scikit-learn)		
역할 분담	김민정 : 데이터 분석(DA), 백엔드(BE-flask) ① DA : 데이터 분석/전처리 및 예측 모델 구축 ② BE-flask : DB 기반 그래프 구현, 모델 기반 예측값을 출력하는 모델을 서버와 연결 배지현 : 프론트엔드(FE), 백엔드(BE-spring boot), DB 설계 ① 웹 서비스 화면 구현 : 로그인, 회원가입, 게시판 구현, 예측 결과 시각화, 누적 강수량 그래프		

2. 프로젝트 수요기업과 필요성, 목적

2.1 프로젝트 추진 배경과 필요성

- 프로젝트 추진 배경 : 강수량은 다양한 변수들이 영향을 미치기 때문에, 예측의 신뢰도를 높이려면 다양한 변수를 사용하여 예측 모델 구현하는 것이 중요하다. 본 공모안에서는 기상자료를 활용하여 예측 모델을 개발하였다.
- 필요성 : 강수량은 국민의 실생활에 큰 영향을 미치는 기상요소 중 하나이다. 그렇기에 이러한 시간대별 누적 강수량을 비교적 정확하게 예측할 수 있다면 수해 대비 방면의 편의 및 국민의 생활 편익을 증대시킬 수 있다.

2.2 프로젝트 목적

- 데이터 분석 목적 : 강수량은 여러 위치의 온도, 습도, 압력, 바람 등의 요소를 포함하는 관측 데이터이다. 기상 관측소, 위성, 레이더 및 기타 장비에서 수집되며, 학습시킨 모델이 정확한 누적 강수량을 예측하고, 그 강수량을 기반으로 강수량 계급구간을 산정한다.
- 기대 성과 : 다양한 상황에서 기상자료를 활용하여, 정확한 강수량을 예측하는 기술은 각 지역에서 일상생활에 필요한 기상 정보 생산 시 유용한 역할수행을 할 것이다.

3. 데이터 분석 계획 서비스 설계

3.1 데이터 분석 설계

1) EDA : 대회 제공 데이터 탐색
2) 변수 : 기상 요소 활용 및 수치모델(3시간 단위) 앙상블 강수 확률 자료 전처리
3) 5~9월의 누적 강수량(계급구간) 예측 (단일 모델 개발)

3.2 웹 서비스 기능 설계

1) 예측 모델 시각화
2) 로그인/회원가입 기능
3) 게시판 기능

4) 데이터베이스 기반 누적 강수량 그래프로 시각화

(이때 DB의 데이터와 검증 데이터를 만들 때의 데이터는 상이하며, 별개의 모델링을 통해 시각화된 그래프이다.)

	사용 데이터	설명
DB 및 시각화	combined_data.csv	rainfall_train.csv 파일과 rainfall_test.csv 파일에서 vv, class_interval 값을 비운 채로 합쳐진 데이터
제출 모델	rainfall_train.csv	train_data
	rainfall_test.csv	test_data : test score 산출 및 검증 단계에만 이용

4. 수집 데이터세트 출처, 건수, 주요 내용

4.1 데이터세트

- 데이터세트 출처 : 기상청(날씨마루)
- 데이터세트 건수 : train 데이터 1,048,575건, test 데이터 122,001건
- 주요 내용 : 앙상블 구간별 누적 확률, 기준 발표 시각, 예측 시간, AWS 지점 코드, 실강수량, 강수 계급 등

5. 프로젝트에서 다루고자 하는 문제(해결해야 하는 현안) 분석과 주요 설계

- 웹 서비스 설계: 웹 서비스 화면 설계, 로그인, 수집 데이터 가시화, 분석 결과 가시화

1) 로그인 및 회원가입 후 중앙의 아이콘 클릭 시 예측 모델 결과값 시각화 화면 전환

2) DB에 있는 데이터로 누적 강수량 예측 그래프 시각화

3) 로그인, 회원가입, 게시판 구현

4) DB 설계: 웹 서비스에 필요한 테이블과 필드를 정의

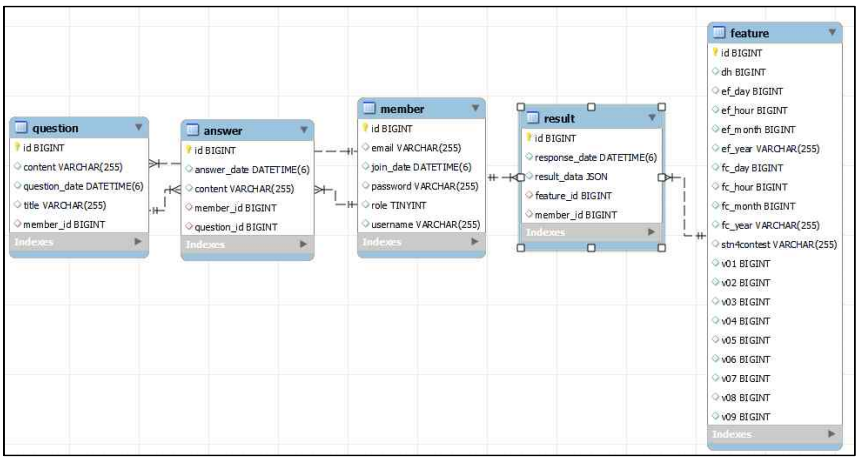
① member : 회원가입을 위한 테이블

② data : 대회에서 주어진 train data&test data를 합쳐서 feature 값을 만들어냄

③ question : 질문 게시판을 위한 테이블

④ answer : 답변을 위한 테이블

⑤ result : flask에서 넘어온 결과를 저장하는 테이블



- 데이터 학습 모델 설계 (데이터 전처리, 학습 모델, 모델 성능 개선, 모델 평가)

1) 분석 데이터 정의

① 수집 대상 데이터세트의 출처 : 기상청

본 공모안에서는 대회에서 제공한 데이터의 시간별 강수량 및 강수량의 앙상블 구간별

누적확률을 이용하여 실체 관측값인 vv(아래 표의 rainfall_train.vv와 동일)를 구하고, vv의 값을 기반으로 강수 계급을 산출하고자 하였다.

② 파생변수 생성

기준변수	파생 변수명	변수 설명 등
rainfall_train.fc_year	fc_year	기준 발표년도
rainfall_train.fc_month	fc_month	기준 발표월
rainfall_train.fc_day	fc_day	기준 발표일
rainfall_train.fc_hour	fc_hour	기준 발표시각
rainfall_train.stn4contest	stn4contest	AWS 지점 코드
rainfall_train.dh	dh	기준시간에서 예측시간을 뺀 값
rainfall_train.ef_year	ef_year	실제 예측 년도
rainfall_train.ef_month	ef_month	실제 예측 월
rainfall_train.ef_day	ef_day	실제 예측 일
rainfall_train.ef_hour	ef_hour	실제 예측 시각
rainfall_train.v01	v01	0.1mm 이상 누적 확률
rainfall_train.v02	v02	0.2mm 이상 누적 확률
rainfall_train.v03	v03	0.5mm 이상 누적 확률
rainfall_train.v04	v04	1.0mm 이상 누적 확률
rainfall_train.v05	v05	2.0mm 이상 누적 확률
rainfall_train.v06	v06	5.0mm 이상 누적 확률
rainfall_train.v07	v07	10.0mm 이상 누적 확률
rainfall_train.v08	v08	20.0mm 이상 누적 확률
rainfall_train.v09	v09	30.0mm 이상 누적 확률

기준변수	파생 변수명	변수 설명 등
rainfall_test.vv	vv	실 강수량
rainfall_test.class_interval	class_interval	강수 계급

2) 데이터 전처리

① 데이터 결측치 처리

isnull, sum 등을 활용하여 결측값이 있는 부분을 확인했지만, test data의 class_interval 열 외에 결측치가 있는 구간이 없었다.

② 데이터 타입 변환

변수 중 하나로 쓰일 fc_year와 ef_year 열의 값을 cat code를 통해 object 타입에서 int 타입으로 변환하였다.

③ 이상치 처리

이상치 처리 과정을 거친 결과 test score 값이 현저하게 낮게 나와 이상치 처리 과정을 주석으로 처리하였다.

④ 컬럼 매핑

train 데이터와 test 데이터 컬럼을 매핑 후 test 데이터를 리네이밍했다. 모델 학습 후 결과물 출력 시 원래의 열 이름으로 복구한 후 출력했다.

⑤ 확률 처리

학습시킬 rainfall_train.v01부터 v09까지의 값과 정확도 검증을 위해 쓰일 rainfall_test.v01부터 v09까지의 값을 실제 확률로 바꾸기 위해 100으로 나눈 후 모델에 학습시켰고, 학습이 끝난 후 결과물 출력 시 원래의 열값으로 복구한 후 출력하였다.

또한 리네이밍된 test data의 각 구간의 중간값에 해당 확률의 값을 곱한 후 임의로 예상 강수량을 계산한 것과 학습한 모델이 산출하는 데이터를 비교하고자 하였다.

3) 학습 모델

① 분석 도구 :

a. 의도

ensemble 모델을 활용하여 Random Forest Regressor, Extra Tree Regressor, LightGBM Regressor, GBMRegressor, XGboost Regressor 등의 모델의 성능 비교를 통해 단일 모델을 개발하고자 하였다.

b. 과정

Extra Tree와 Random Forest의 성능이 비슷하여 두 모델을 더 정밀하게 비교하였다.

c. 결과

Random Forest의 test score가 높아서 Random Forest 모델을 사용하게 되었다.

② 분석 도구 검증 :

a. 의도

gridSearchCV와 randomizedSearchCV의 이용하여 최적의 하이퍼 파라미터를 도출하고자 하였다.

b. 과정

gridSearchCV와 randomizedSearchCV를 사용한 모델의 하이퍼 파라미터를 동시에 산출하여 비교했을 때, gridSearchCV를 활용하여 하이퍼 파라미터를 도출하는 데에 소요되는 시간이 randomizedSearchCV를 활용했던 경우보다 현저하게 길었다.

c. 결과

randomizedSearchCV를 사용하여 50만 건의 데이터를 학습시켰다. 이때 20만 건의 데이터를 학습시키는 것과 50만 건의 데이터를 학습시키는 경우의 정확도는 큰 차이를 보이지 않았다. 본 공모안에서는 50만 건의 데이터를 학습시킨 코드를 제출하였다.

4) 모델 평가

data leakage 원칙을 지키기 위해 본 공모안과 함께 제출하는 모델에는 train data와 test data는 test score를 산출하기 위한 용도로만 사용되었다.

a. 과정

randomSearchCV를 활용하여 모델을 초회 학습시켜서 초기의 test score와 최적의 하이퍼 파라미터를 찾은 다음, 이를 적용하여 재학습시켜 train score와 test score를 얻을 수 있었다.

b. 결과

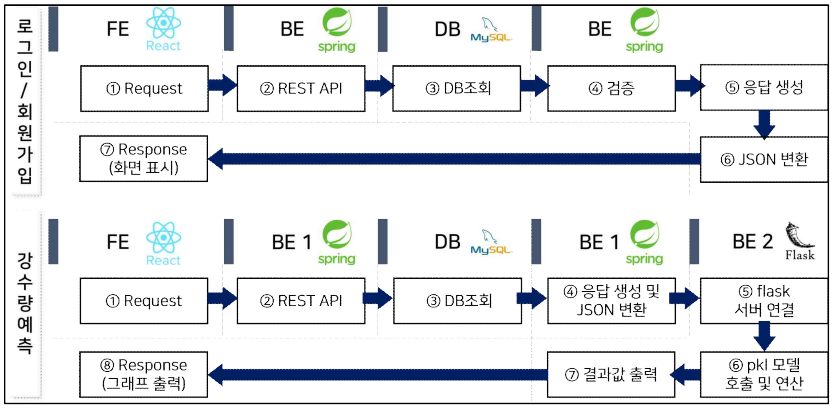
test score는 약 0.3 혹은 0.2 후반대의 값이 대부분이었고, 웹페이지에서 검증 결과 확인 시 약 6-70%의 정확도를 보이는 것으로 추정되었다.

6) 데이터셋 제출 과정

결과 데이터를 csv 파일로 출력 시 컬럼 매핑 후, feature값 및 재리네이밍 범위에 들어가지 않은 rainfall_train.vv 열의 값이 포함되어 나와서 vv 열을 drop한 상태로 재출력하고자 하였다. 이때 데이터셋 제출 시 열 및 행 건수를 확인해도 기존 rainfall_test.csv 데이터와 동일함에도 데이터셋 양식이 맞지 않는다는 오류 메시지가 출력되었다. 이에 기존 rainfall_test.csv 파일에 rainfall_test.class_interval 열을 따로 엑셀을 통해 붙여넣은 후 제출하였다.

6. 시스템 구성도 설계

프레임워크



- FE(Front End) : react, node js, tailwind, js
- BE(Back End) : spring boot, flask
- DA(Data Analysis) : ipython notebook, python, google colab
- 데이터 교환/공유 format : github, google drive

7. 프로젝트 추진 일정(주별 추진 일정)

기간	일정
1주차 (5/30-6/7)	프로젝트 주제 선정
2주차 (6/10-6/14)	FE : 화면 설계 BE-spring boot : DB 설계 DA : 제공 데이터 분석 및 전처리
3주차 (6/17-6/21)	FE : 화면 설계 및 생성 BE-spring boot : DB 생성 및 로그인/회원가입 기능 구현 BE-flask : 모의 모델 pkl 형식 저장 DA : 데이터 전처리 및 모델 생성
4주차 (6/24-6/28)	FE : 화면 구현 BE-springboot : 게시판 기능 구현 및 예측 모델 결과값 시각화 BE-flask : 성능 개선 모델 pkl 형식 저장 및 서버에서 모델 호출 DA : 모델 성능 개선

8. 프로젝트 기대 효과 및 실제 실행

8.1 기대되는 웹서비스 기능/화면

1) 누적 강수량 예측 모델의 결과값을 시각화하여 다양한 사람들에게 서비스
① 로그인 및 회원가입 후 중앙의 아이콘 클릭 시 예측 모델 결과값 시각화 화면 전환
② DB에 있는 데이터로 누적 강수량 예측 그래프 시각화
③ 로그인, 회원가입, 게시판 구현
2) 누적 강수량을 그래프로 시각화하여 데이터 이해에 용이
3) 게시판을 통해서 모델의 결과값 및 의문점 질문 가능

8.2 기대되는 데이터 분석 예측 성과

1) 구간별 앙상블 예측 확률을 통해 누적 강수량 예측 모델을 만들 수 있다.
2) 누적 강수량을 예측하고, 그 값을 기반으로 강수 계급구간을 산정할 수 있다.

8.3 구현 화면 (제출본과 별개 모델을 만들어 가동. DB에 train data와 test data의 데이터를 v09값이 있는 열까지 존재하며 vv값과 class_interval 값을 출력하는 웹페이지)

1) flask 서버 부분
① app.py 실행
② http://127.0.0.1:5000 접속
③ postman에서 localhost:5000/predict를 post 형식으로 입력 후 Body 부분의 raw 선택 후 REST API 입력될 때 만들어질 json 값을 예시 하나로 가져와서 입력
{ fcYear=A, fcMonth=7, fcDay=20, fcHour=9, dh=240, v01=69, v02=61, v03=44, v04=33, v05=22, v06=10, v07=2, v08=0, v09=0, stn4contest=STN009, efYear=A, efMonth=7, efDay=30, efHour=9 }
④ 결과값 다시 JSON으로 출력 : vv, class_interval

9. 참고문헌/자료 (인용)

DA	sklearn.preprocessing.standardscaler	https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html#sklearn.preprocessing.StandardScaler
	label encoder	https://stackoverflow.com/questions/51102205/how-to-know-the-labels-assigned-by-astypecategory-cat-codes
	catcodes	https://stackoverflow.com/questions/51102205/how-to-know-the-labels-assigned-by-astypecategory-cat-codes
	RandomForest Regressor	https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
	ExtraTree Regressor	https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html
	모델 학습	https://www.tensorflow.org/guide/keras/customizing_what_happens_in_fit?hl=ko
	모델 저장	https://neptune.ai/blog/saving-trained-model-in-python
flask	(REST API)	https://justkode.kr/python/flask-restapi-1/ https://velog.io/@mingming_eee/Flask-day2 https://tutorials.pytorch.kr/intermediate/flask_rest_api_tutorial.html
REACT	게시판	https://www.tailwindawesome.com/resources/d-board
	tailwind	https://tailwindui.com
FIGMA	전체 이미지 구상	https://www.figma.com/files/team/1378169529263048086/recents-and-sharing/recently-viewed?fuid=1378169527495189006
spring boot	로그인, 회원가입	https://github.com/LeeYeongin
	JWT filter	https://velog.io/@qiwisil_227/%EC%98%88%EC%A0%9C-%EC%BD%94%EB%93%9C-%EB%A1%9C%EC%A7%81-%EC%9D%B4%ED%95%B4%ED%95%98%EA%B8%B0
	Authorization	https://velog.io/@hyex/HTTP-Authorization-header%EC%97%90-Bearer%EC%99%80-jwt-%EC%A4%91-%EB%AC%B4%EC%97%87%EC%9D%84-%EC%82%AC%EC%9A%A9%ED%95%A0%EA%B9%8C