서울시 따릉이 자전거 이용 예측

CUAI 4기 I팀
김동영(소프트웨어), 서준영(AI), 윤다인(소프트웨어), 이승연(소프트웨어), 이주영(소프트웨어), 이효근(소프트웨어)

[요약] 본 연구의 목적은 공공 빅데이터인 서울시 따릉이 데이터를 이용하여 인공지능 모델을 개발하는 데 있다. 본 연구에서는 서울시 마포구의 날짜별, 시간별 기상상황과 따릉이 대여 수 데이터를 활용한다. 데이터의 모든 컬럼이 수치형 데이터이며 결측치를 채워 넣는 작업이 중요하다. 시각화로 대여 수 추이와 상관 계수가 높은 컬럼을 확인하였고 결정 트리의 개수를 200으로 설정한 랜덤 포레스트 모형이 가장 높은 성능을 보였다.

1. 서 론

따릉이 대여 수는 날짜별, 시간별 기상상황에 영향을 받는다. 이를 통해 따릉이 대여 수도 예측해볼 수 있다. 본 연구에서는 1시간 전 기온, 비 정보, 풍속(평균), 습도, 특정 기상 상태에 따른 가시성, 오존, 미세먼지 데이터를 통해 서울시 마포구의 시간별 따릉이 대여 수를 예측하고자 한다.

따릉이는 기후변화에 매우 유용하게 대처하는 친환경 교통 수단으로 자리매김하였고 시민들의 수요에 부족함 없이 공급하겠다는 것이 서울시 방침이다. 오세훈 서울 시장은 "시간대별 쏠림 현상에 대한 해법은 물량을 늘려가는 것밖에 없다" 말했다. 이에 본 연구에서는 시각화를 통해 상관관계가 높은 3개의 컬럼만을 모델의 학습에 사용하기로 결정하였다. 결측치를 채워 넣는 작업을 진행한 후 앙상블 기법 중 하나인 랜덤 포레스트의 결정 트리의 개수, 트리의 최대 깊이를 달리한 총 세 가지 모델을 구축하여 대여 수를 예측하고 모형 별 성능을 비교하였다.

2. 본 론

1) EDA

따릉이 데이터 컬럼의 종류는 다음과 같다.

Id	날짜와 시간 별 ID
hour_bef_temperature	1시간 전 기온
hour_bef_precipitation	1시간 전 비 정보, 비가 오지 않았으면 0, 비가 오면 1
hour_bef_windspeed	1시간 전 풍속(평균)
hour_bef_humidity	1시간 전 습도
hour_bef_visibility	1시간 전 시정, 시계(특정 기상 상태에 따른 가시성을 의미)
hour_bef_ozone	1시간 전 오존
hour_bef_pm10	1시간 전 미세먼지(머리카락 굵기의 1/5에서 1/7 크기의 미세먼지)
hour_bef_pm2.5	1시간 전 미세먼지(머리카락 굵기의 1/20에서 1/30 크기의 미세먼지)
count	시간에 따른 따릉이 대여 수

모든 컬럼의 데이터는 숫자로 되어있어 수치형 데이터로 변환하는 작업이 필요하지 않다. test데이터는 우리가 예측하려는 count값을 제외한 나머지 10개의 컬럼을 갖고 있으며 submission 파일은 예측한 count값을 입력할 수 있도록 모든 내용이 NaN처리 된다. Train 데이터에 몇몇 컬럼은 결측치가 존재하므로 후에 결측치를 채워 넣는 작업을 진행하기로 결정하였다.

시간에 따른 따릉이 대여 수에 관한 데이터를 시각화 하면 오전 10시가 가장 수치가 낮고 오후 6시가 수치가 가장 높은 것을 확인할 수 있다(새벽 제외). 이를 토대로 출근시간에 사용량이 급감할 것임을 추측할 수 있다.

학습을 위해 모든 컬럼을 사용하는 것은 모델에 악영향을 미치므로 count 값과 상관관계가 높은 컬럼을 활용해 학습을 진행하도록 결정하였다. Hit map 을 통해 시각화 해 본 결과 hour, hour_before_temperature, hour_bef_windspeed 3 개의 컬럼이 count 와의 상관 계수가 가장 높음을 확인하였다.

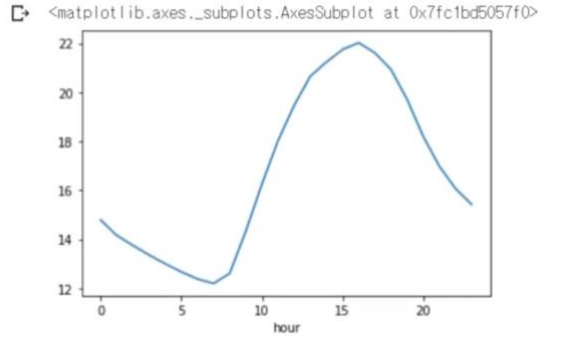
2) 데이터 전처리

모델의 학습에 사용하기로 결정한 변수 **hour, hour_bef_temperature, hour_bef_windspeed**중에서 **hour_bef_temperature**와 **hour_bef_windspeed** 변수는 결측값(Null)을 가지고 있으며, 결측값은 모델링을 하기 전에 반드시 수정되어야 한다. 온도 변수와 풍속

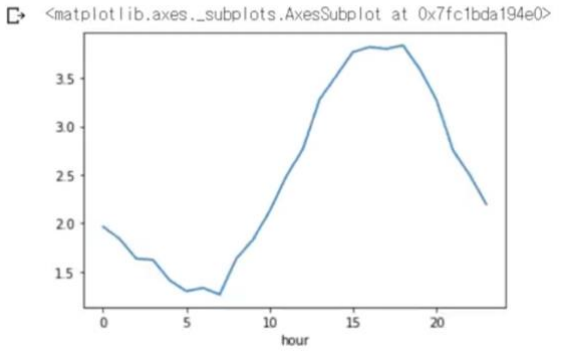
Copyright © 2022 CUAI

변수는 모두 언제든지 고유의 값을 가지고 있으므로 결측값을 단순히 0으로 만드는 것은 바람직하지 않다. 그리고 두 변수는 모두 시간에 따른 경향성을 띄고 있기 때문에 각 변수의 전체 평균을 구하여 결측값에 대입하는 방법은 그런 시간과의 관계성을 무시해버릴 수 있다.

```
[77] train.groupby('hour').mean()['hour_bef_temperature'].plot()
```




```
[85] train.groupby('hour').mean()['hour_bef_windspeed'].plot()
```



예를 들어 시간에 따른 온도 변수를 관찰하면 새벽에는 낮은 온도를, 낮에는 높은 온도를 보인다. 이어서 시간에 따른 풍속 변수를 관찰하면 온도 변수와 유사하게 새벽에는 낮은 풍속을, 낮에는 높은 풍속을 보인다. 이런 경향을 가진 변수들의 결측값을 단순히 평균으로 대체한다면 결측값 주변의 변수 값들과 이질적인 값을 가질 수 있기 때문에 적절하지 않은 결과가 도출될 수 있다.

따라서 본 연구에서는 온도와 풍속이 시간에 따라 규칙적인 흐름을 보인다는 사실을 기반으로 결측값이 발생한 그 시간(hour 변수)의 온도 또는 풍속의 변수를 각각 모아 평균을 산출하고, 그 평균값을 결측값에 대입하여 데이터가 시간 관계성을 보존할 수 있도록 결측치 처리를 수행하였다.

3) 모델링

1. 랜덤 포레스트

본 연구에서 사용된 기계 학습 모형은 분류, 회귀 분석 등에 자주 사용되는 랜덤 포레스트를 사용하였다.

랜덤 포레스트는 다수의 결정 트리들을 학습하는 앙상블 방법이다. 앙상블 알고리즘 중 비교적 빠른 수행 속도를 가지고 있고, 다양한 영역에서 높은 예측 성능을 보이고 있다. 랜덤 포레스트의 기반 알고리즘은 결정 트리로서, 결정 트리의 쉽고 직관적인 장점을 그대로 가지고 있다. 여러 개의 데이터 세트를 중첩되게 분리하는 부트스트래핑(bootstrapping) 분할 방식을 통해 데이터가 중첩된 개별 데이터 세트에 결정 트리 분류기를 각각 적용하는 방식으로 작동한다.

주요 랜덤 포레스트 하이퍼 파라미터는 다음과 같다.

파라미터 명	설명
n_estimators	- 결정트리의 개수를 지정 - Default = 10 - 무작정 트리 갯수를 늘리면 성능 좋아지는 것 대비 시간이 걸릴 수 있음
min_samples_split	- 노드를 분할하기 위한 최소한의 샘플 데이터수 → 과적합을 제어하는데 사용 - Default = 2 → 작게 설정할수록 분할 노드가 많아져 과적합 가능성 증가
min_samples_leaf	- 리프노드가 되기 위해 필요한 최소한의 샘플 데이터수 - min_samples_split 과 함께 과적합 제어 용도 - 불균형 데이터의 경우 특정 클래스의 데이터가 극도로 작을 수 있으므로 작게 설정 필요
max_features	- 최적의 분할을 위해 고려할 최대 feature 개수 - Default = 'auto' (결정트리에서는 default 가 none 이었음) - int 형으로 지정 →피처 갯수 / float 형으로 지정 →비중 - sqrt 또는 auto : 전체 피처 중 √(피처개수) 만큼 선정 - log : 전체 피처 중 log2(전체 피처 개수) 만큼 선정
max_depth	- 트리의 최대 깊이 - default = None → 완벽하게 클래스 값이 결정될 때 까지 분할 또는 데이터 개수가 min_samples_split 보다 작아질 때까지 분할 - 깊이가 깊어지면 과적합될 수 있으므로 적절히 제어 필요
max_leaf_nodes	리프노드의 최대 개수

Copyright © 2022 CUAI

〈표1〉 랜덤 포레스트 하이퍼 파라미터 명 및 설명

2. 모델 구축

결정 트리의 개수, 트리의 최대 깊이에 의한 성능 비교를 위해 총 세 가지 모델을 구축하고자 하였다. 첫 번째 모델은 결정 트리의 개수를 100으로 지정하였고, 두 번째 모델은 첫 번째 모델과 같이 결정 트리의 개수를 100으로 지정하고, 트리의 최대 깊이를 5로 지정하였다. 세 번째 모델은 결정 트리의 개수를 200으로 지정하여 모델을 구축하였다.

model100 = RandomForestRegressor(n_estimators = 100, random_state = 0)
model100_5 = RandomForestRegressor(n_estimators = 100, max_depth = 5, random_state = 0)
model200 = RandomForestRegressor(n_estimators = 200)

3. RMSE

본 연구에서는 랜덤 포레스트 모델을 RMSE(Root Mean Square Error; 평균 제곱근 오차)를 이용하여 평가하였다. 평균 제곱 오차를 사용하지 않은 이유는 오차 합의 값이 굉장히 크게 나오는 경우 연산 속도가 느려진다는 단점이 있어 이 값에 루트를 적용한 평균 제곱근 오차를 이용하였다.

$$\text{평균제곱근오차}(RMSE) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\text{실제값} - \text{예측값})^2}$$

4. 결과

전처리를 거친 데이터 세트를 세 가지 종류의 랜덤 포레스트를 활용하여 기계학습을 하였다. 모형마다 하이퍼 파라미터를 다르게 설정한 후 테스트 데이터로 최종적으로 성능을 평가하였다. 이의 결과는 다음의 표와 같다.

모형 이름	RMSE 값
Model100	50.1905420613
Model100_5	51.7719422024
Model200	49.2887916522

세 모형 모두 RMSE가 50정도를 기록하였다. 결정 트리의 개수를 200으로 설정한 model200이 RMSE 값이 약 49 정도로 세 모형 중 가장 높은 성능을 보였다. 결정 트리의 개수를 100으로 지정하고, 트리 깊이를 5로 설정한 model100_5가 세 모형 중 가장 낮은 성능을 보였다.

3. 결 론

본 연구는 서울시 마포구의 날짜별, 시간별 기상상황과 따릉이 대여 수 데이터를 분석하여 test데이터의 대여 수를 예측하였다. Hit map을 통한 시각화를 활용하여 상관 계수가 높은 hour, hour_before_temperature, hour_bef_windspeed 3개의 컬럼을 활용해 학습을 진행하였다. 결정 트리의 개수, 트리의 최대 깊이에 의한 성능 비교하고자 총 세 가지 렌덤 포레스트 모델을 구축하였다. 빠른 연산 속도를 위해 평균 제곱근 오차를 이용하였고 이들의 예측 결과는 다음과 같다.

세 모형 모두 RMSE가 50정도를 기록하였지만 결정 트리의 개수를 200으로 설정한 model200이 RMSE 값 49.2887916522로 세 모형 중 가장 높은 성능을 보였고 결정 트리의 개수를 100으로 지정하고, 트리 깊이를 5로 설정한 model100_5가 50.1905420613로 세 모형 중 가장 낮은 성능을 보였다.

본 연구의 한계점은 다음과 같다. 본 연구에서는 랜덤 포레스트 하이퍼 파라미터 중 n_estimators, max_depth에만 변화를 주었다. 따라서 각각의 하이퍼 파라미터들을 모두 활용하여 다양한 조건의 모델을 만들어서 결과의 변화를 확인하고 성능이 더 뛰어난 모델에 도달하기 위해 튜닝을 해 볼 필요가 있다. 또한 본 연구에서는 온도와 풍속이 시간에 따라 규칙적인 흐름을 보인다는 사실을 기반으로 결측값이 발생한 그 시간(hour 변수)의 온도 또는 풍속의 변수를 각각 모아 평균을 산출하고, 그 평균값을 결측값에 대입하여 결측치 처리를 수행하였다. 결측값을 단순히 0으로 만드는 방법 또는 각 변수의 전체 평균을 구하여 결측값에 대입하는 방법 보다는 시간과의 관계성을 고려하였다고 볼 수 있지만 보다 적합한 데이터 전처리 방법을 연구할 필요가 있다.

참고 문헌

1) DACON 서울시 따릉이 자전거 이용 예측 AI모델. Available: https://dacon.io/competitions/open/235576/overview/description
2) 오세훈 "따릉이, 부족함 없이 더 많은 숫자 공급하겠다는 것이 서울시 방침". Available: https://n.news.naver.com/article/056/0011142381
3) 랜덤포레스트. Available: https://injo.tistory.com/30
4)RMSE. Available: https://blog.naver.com/owl6615/221537580561

Copyright © 2022 CUAI