빅데이터 분석

김희찬 교수님

중간고사 대체 보고서

20163231 소프트웨어학부 신승은

**데이터 전처리**

1. 정확도를 위해 분당통화요금의 결측치를 제거하고 다시 불러온다.

2. 다시 불러온 데이터의 index는 31743개, column은 40개다. (이전 데이터의 index는 31769개)

3. 우리가 사용하지 않는 column은 ‘id’, ’고객id’, ‘개시일’ 이므로 제거해주어 우리가 분석할 Data Frame(변수를 df로 씀)을 완성한다.

4. 실습에서는 np.percentile로 통계값을 알고 싶은 column을 구간을 나눠 분위 수(0,25,75,100과 같이)를 구해줘 기술 통계값을 알 수 있다.. 본 보고서에서는 Pandas패키지에서 series/dataframe의 describe메소드로 모든 column의 기술 통계값을 한번에 계산하는 명령어를 사용하여 기초 통계량을 산출한다.

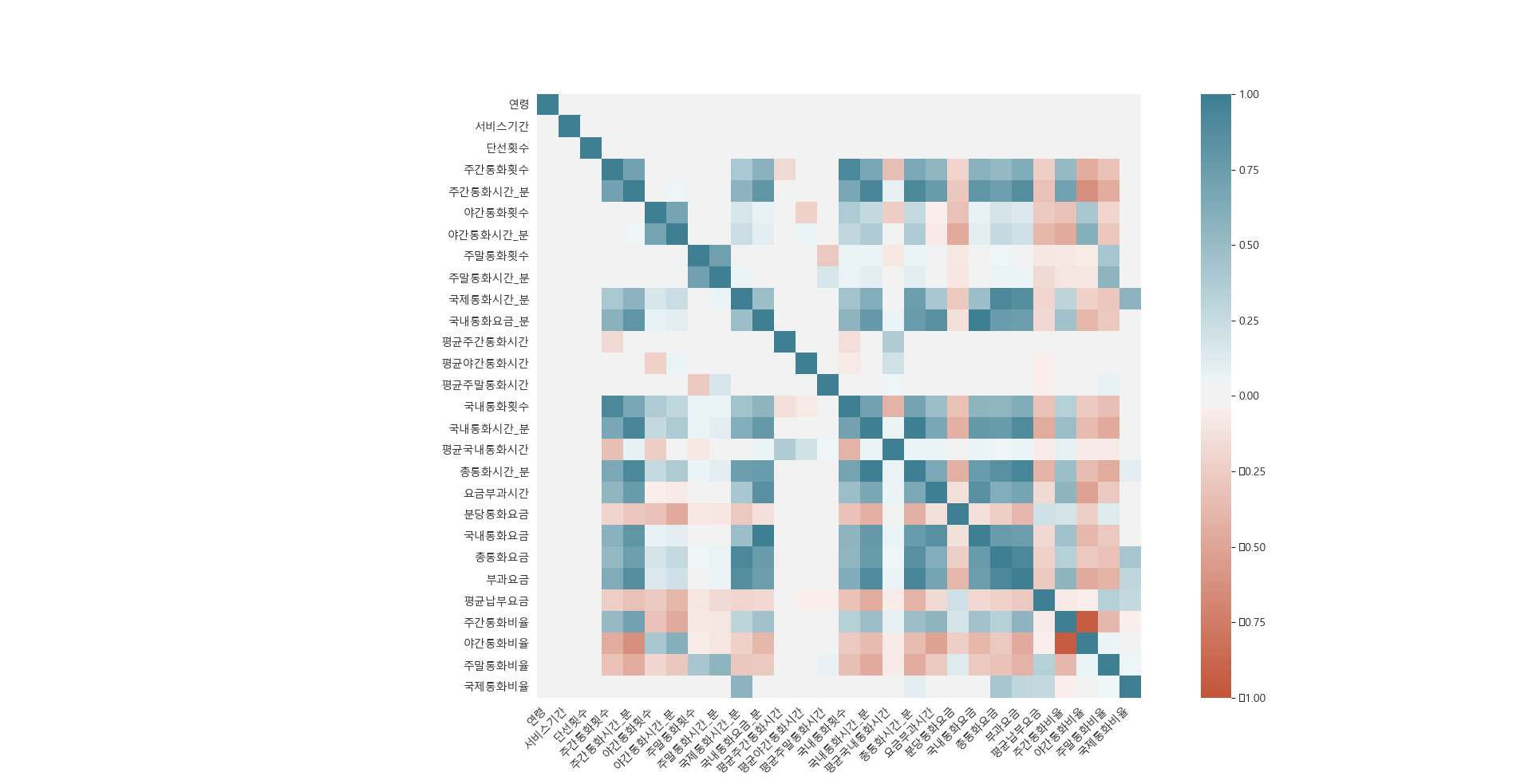
*#console에 df.describe()을 입력한 후 결과*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **연령** | **서비스기간** | **단선횟수** | **주간통화횟수** | **주간통화시간\_분** | **야간통화횟수** | **야간통화시간\_분** | **주말통화횟수** | **주말통화시간\_분** | **국제통화시간\_분** | **...** | **요금부과시간** | **분당통화요금** | **국내통화요금** | **총통화요금** | **부과요금** | **평균납부요금** | **주간통화비율** | **야간통화비율** | **주말통화비율** | **국제통화비율** |
| **count** | 31743 | 31743 | 31743 | 31743 | 31743 | 31743 | 31743 | 31743 | 31743 | 31743 | ... | 31743 | 31743 | 31743 | 31743 | 31743 | 31743 | 31743 | 31743 | 31743 | 31743 |
| **mean** | 30.331349 | 33.223005 | 3.139495 | 232.832971 | 699.020842 | 99.593454 | 297.17832 | 16.623728 | 49.795977 | 170.481781 | ... | 70.416272 | 10.19869 | 19.568431 | 73.987202 | 185.80562 | 0.17711 | 0.609162 | 0.325175 | 0.065662 | 0.16512 |
| **std** | 12.858735 | 14.273435 | 3.970912 | 239.114971 | 516.38787 | 95.408306 | 197.199107 | 16.849964 | 36.481825 | 145.069997 | ... | 472.122428 | 2.147123 | 28.588446 | 62.146981 | 85.688727 | 0.106362 | 0.225761 | 0.20976 | 0.071939 | 0.112286 |
| **min** | 12 | 9.533333 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.009789 | ... | -1749.6 | 2 | 0 | 0.003313 | 59.953259 | 0.045692 | 0 | 0 | 0 | 0.00001 |
| **25%** | 20 | 20.433333 | 1 | 53 | 284.399999 | 26 | 135.900001 | 4 | 21.6 | 58.938252 | ... | -129 | 8.62875 | 0 | 29.433249 | 119.253979 | 0.137342 | 0.460351 | 0.160944 | 0.020811 | 0.081213 |
| **50%** | 28 | 33.1 | 2 | 154 | 603 | 69 | 272.100001 | 11 | 43.5 | 133.497323 | ... | 69.6 | 9.615248 | 7.157354 | 58.74918 | 173.234245 | 0.156996 | 0.650477 | 0.288138 | 0.045571 | 0.16286 |
| **75%** | 38 | 45.633333 | 4 | 336 | 1009.799999 | 147 | 428.1 | 24 | 72.6 | 243.848617 | ... | 293.699999 | 11.874327 | 29.988557 | 100.422727 | 234.543186 | 0.182842 | 0.786734 | 0.458005 | 0.085391 | 0.246117 |
| **max** | 82 | 58.2 | 15 | 1725 | 2901.600001 | 647 | 1146 | 124 | 205 | 1014.818532 | ... | 2350.8 | 25 | 202.490395 | 474.433782 | 624.433782 | 5.552083 | 1 | 1 | 1 | 6.26182 |

**시각화 분석**

**\*상관관계 분석**

각 column들의 상관관계를 파악하면 직관적인 분석을 할 수 있어 상관관계를 heatmap이라는 그래프로 그립니다.



숫자 데이터를 가진 각 column의 상관계수가 1에 가까워질수록 하나가 증가할 때 다른 하나도 함께 증가합니다. -1에 가까울수록 하나가 증가할 때 다른 하나는 감소합니다.

-1에 가장 가까운 column은 야간통화비율과 주간통화비율이고, 그 다음으로 가까운 column은 야간통화비율과 주간통화시간\_분입니다. 둘 중 하나가 증가하면 다른 하나는 감소합니다

자기 자신에 대한 상관계수를 제외한 1에 가장 가까운 column은 국내통화요금과 국내통화요금\_분(분당 통화요금)이고, 총통화요금과 국제통화시간\_분입니다. 둘 중 하나가 증가할 때 다른 하나도 같이 증가합니다.

모든 column의 상관관계를 시각화한 다음 각 변수별로 상관관계를 시각화하여 그래프를 그립니다.

**\*종속 변수의 항목별 수 확인**

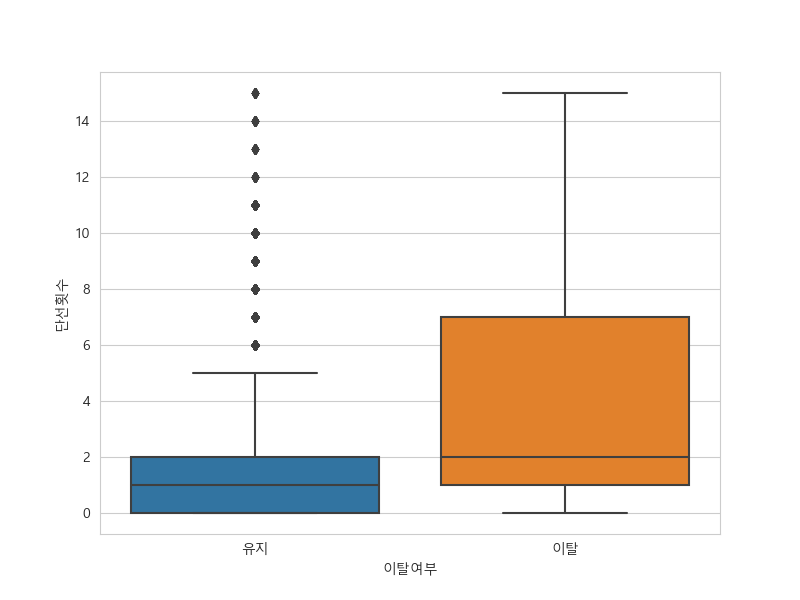
스크린샷, 그리기이(가) 표시된 사진

자동 생성된 설명

여기서 우리는 유지가 17048건, 이탈이 14695건이 존재한다는 것을 알 수 있습니다.

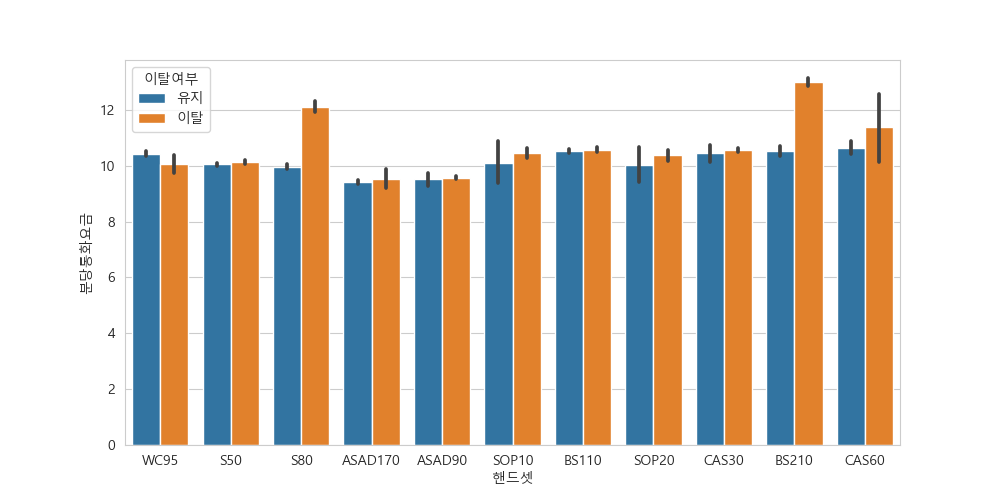
**\*이탈여부와 단선횟수의 박스그래프**

그 다음 이탈여부와 단선횟수의 관계를 그래프로 시각화하여 알아보겠습니다. 우리는 연속형변수 대해서 분석하고 있으므로 seaborn으로 여러 개의 그룹간 비교하는데에 쓰이는 박스그래프를 그려봅니다.



이탈여부에 따라서 단선횟수가 큰 차이를 보입니다. 이탈의 경우가 단선횟수가 월등히 많은 것을 알 수 있습니다.

**\*핸드셋 종류와 이탈여부 별 분당 통화요금의 평균분포**



핸드셋의 종류에 따라서 이탈하는 사람들과 유지하는 사람들의 분당통화요금의 평균분포를 나타낸 그래프를 seaborn의 barplot을 이용하여 그립니다.

S80과 BS210, CAS60핸드셋 종류는 분당통화요금이 이탈여부에 따라 크게 달라지는 것을 알 수 있습니다.

**\*통화품질 불만여부와 이탈여부별 평균납부요금의 선그래프**

지도, 남자이(가) 표시된 사진

자동 생성된 설명

Seaborn의 lineplot으로 통화 품질 불만여부와 이탈여부별 평균납부요금의 선 그래프를 그립니다.이탈의 경우가 유지하는 경우보다 평균납부요금이 높은 편에 속하는 것을 알 수 있다. 또한 통화품질의 불만도 없는 것을 알 수 있습니다.

**모델링**

모델은 RandomForest를 선택한다. 그 이유는 MLPClassifier, LogisticRegression, Support Vector Machine, Naïve Bayesian등의 방법으로 기본모델을 설정해 정확도를 산출했으나, 모두 정확도 지표가 0.6에서 0.7사이의 정확도로 Ensemble 기법인 RandomForest기법에 비해 상당히 미비합니다.일반적인 Linear model이나 Tree model보다는 Ensemble 기법이 적합한 것으로 판단하여 RandomForest 모델을 선택했습니다.

**\*범주형 변수들 라벨링**

1. 범주형 변수들은 ['지불방법', '요금제', '이탈여부', '핸드셋', '통화량구분', '납부여부', '미사용', '통화품질불만', '성별'] 이다. 위 변수들을 매핑한다.

2. 범주형 변수들을 숫자형으로 변환하기 위해 클래스 분포를 확인한다.

3. 미사용과 지불방법은 한가지 항목밖에 없으므로 제거한다.

4. 범주형 변수들을 항목별로 숫자형 변수들로 바꿔준다.

5. 이제 범주형 변수를 변환한 후 모두 숫자형으로 매핑되어있는 것을 확인할 수 있고, 변수로 투입이 가능하다.

**\*모델구축**

1. X는 독립변수로 ‘이탈여부’를 제외한 data frame, Y는 종속변수로 ‘이탈여부’만 있는 data frame을 생성한다.

2. 데이터를 7:3으로 나눈다. (학습 : 7, 테스트 : 3)

3. 학습 데이터, 학습 레이블, 평가 데이터, 평가 레이블을 shuffle한 인덱스를 넣어 생성한다.

4. RandomForest 모델생성 후 학습 셋으로 학습 시킨다.

5. test 셋을 이용해 예측한다.

6. 학습데이터와 테스트데이터를 내부적으로 10번 쪼개어 각각에 대해 10fold cross validation을 수행한다.

7. 10번에 대한 평균으로 평균 정확도를 산출한다.

* 10fold cross validation을 통해 92.60%의 정확도을 얻을 수 있다.

**\*모델튜닝 - (하이퍼 파라미터 튜닝)**

강의시간에 진행했던 multiprocessing방법으로 pool클래스를 이용하여 파라미터 튜닝을 하려고 했으나

multiprocessing.pool.MaybeEncodingError: Error sending result: '<multiprocessing.pool.ExceptionWithTraceback object at 0x00000207527CB2B0>'. Reason: 'PicklingError("Can't pickle <class 'MemoryError'>: it's not the same object as builtins.MemoryError",)

가 발생했습니다. 메모리 문제인지 파악하지 못하여 결국엔 시간의 한계로 해결하지 못했습니다.

시간이 오래 걸리지만 RandomForest는 트리기반의 하이퍼 파라미터에 Bagging, boosting, learning, normalizing 등을 위한 하이퍼 파라미터까지 추가되므로 튜닝할 파라미터가 많아 RandomForest의 하이퍼 파라미터 튜닝을 GridSearchCV를 통해 진행했습니다.

1. 모든 하이퍼 파라미터를 튜닝하는 것은 시간이 많이 소요되므로 모든 범위에 대해서 수행하는 것은 불가합니다. 따라서 RandomForest의 주요 하이퍼 파라미터들에 대한 튜닝, 가장 높은 정확도를 보이는 파라미터를 산출합니다.

2. Rf=GridSearchCV(model,param\_grid=params,cv=5)

Cv=5이므로 train셋에서 자체적으로 학습 검증셋으로 나누어 5fold-cv를 실시해 가장 좋은 평균정확도를 가지는 하이퍼 파라미터 조합을 산촐합니다.

3. 각 파라미터 조합에 대한 정확도 값을 산출한 다음, 가장 높은 정확도를 보이는 파라미터를 산출합니다.

4. 최적 하이퍼 파라미터일 때의 정확도는 0.92605가 나옵니다.

5. 교정된 하이퍼 파라미터를 적용한 모델로 학습데이터와 테스트데이터를 내부적으로 10번 쪼개어 각각에 대해서 10fold-cv를 수행합니다. 10번에 대한 평균으로 평균 정확도를 산출합니다.

6. 평균 정확도는 0.92633가 나옵니다. 튜닝 후 10fold-cv가 0.92605에서 0.92633으로 증가했습니다. 이로서 튜닝 후 정확도의 개선을 알 수 있었습니다.

**\*참고 사이트**

시각화 분석 -> 상관관계 그래프

https://blog.naver.com/PostView.nhn?blogId=kiddwannabe&logNo=221205309816&parentCategoryNo=&categoryNo=38&viewDate=&isShowPopularPosts=true&from=search

RandomForest의 GridSearchCV를 통한 하이퍼 파라미터 튜닝

https://injo.tistory.com/30