**워드 사용법 정리**

* **사전 세팅**

1. (워드) 단락 뒤에 공백 제거
2. (워드) 도형 – 상자 – 채우기: 회색, 테두리: 검은색, 글씨: 검은색 – 기본 도형으로 설정
3. (워드) 자동 글머리 해제 : 파일 > 옵션 > 언어 교정 > 자동 고침 옵션 > 입력할 때 자동 서식 > 자동 글머리 기호 목록, 자동 번호 매기기 목록 해제
4. (워드) 글머리용 빈 사각형, 빈 원 찾아두기 - 메모장  
   글머리 기호 및 번호는 아래와 같음

**1. 대제목 14pt, bold**

**1) 소제목 13pt, bold**

**🞏 세 번째 제목 11pt, bold**

⭘ 네 번째 제목 10pt

- 내용 10pt

1. 캡처도구 켜두기
2. C:/windows/fonts에서 맑은 고딕 파일(malgun.ttf) Jupyter Notebook으로 옮겨두기  
   (fonts 폴더에서 ‘맑은’ 검색하면 됨)
3. Jupyter Notebook에 파일 업로드, 다운로드 테스트
4. 미리 코드 쳐두기(라이브러리 등)
5. 분석 문제 파일 확인

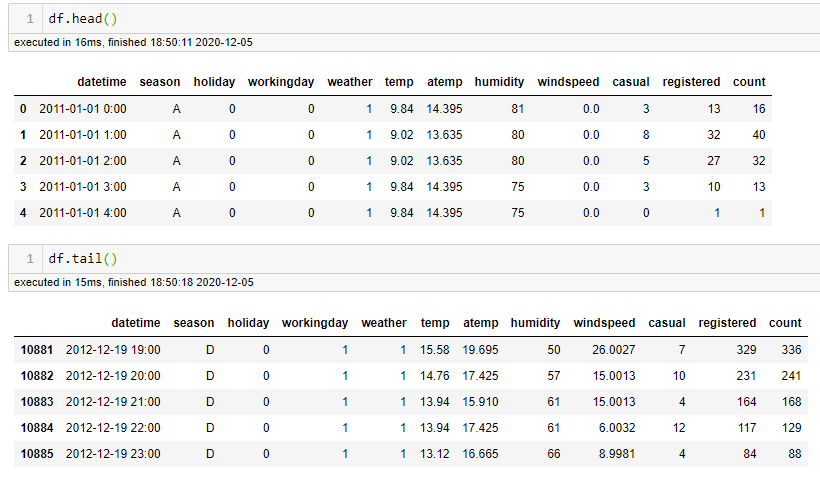
**2020. 12. 13**

**1. 기계학습 1.1 분류**

**문제 1-1**

**1. 데이터 구조 확인**

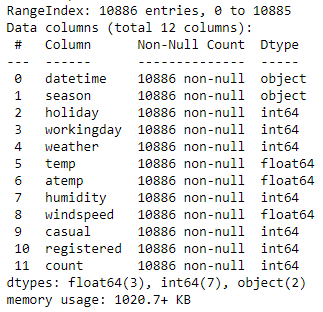
🞏 head, tail을 통해 데이터 처음과 마지막 5행을 확인



🞏 shape로 데이터 구조 확인 : 10,886행 x 12열

🞏 info로 전체 컬럼명과 데이터 타입 확인

⭘ 12개 컬럼 중 float64 타입 3개, int64 타입 7개, object 타입이 2개로 구성



🞏 데이터 타입 보정

⭘ 컬럼 특성에 맞게 데이터 타입을 보정해 주었음

- datetime : object 🡪 datetime64

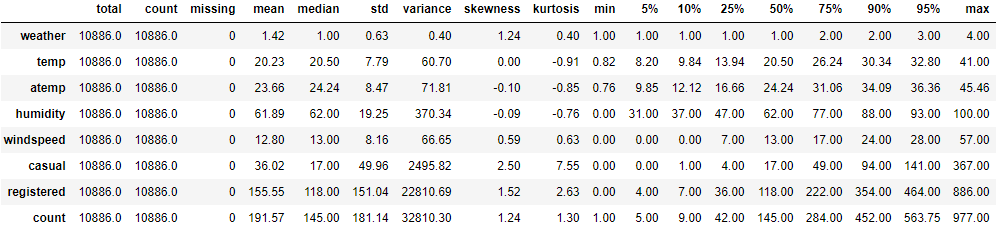
**2. DQ Check(빈도분석, 분포분석)**

🞏 연속형 변수

⭘ 연속형 변수들의 기초통계량과 분포분석을 진행함

- 분포가 90% 이상이 동일한 변수는 모델링 시 설명력이 떨어진다고 판단하여 제외하기로 함

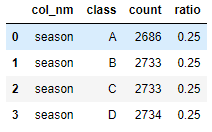
- 이 데이터에서는 모든 변수들이 고른 분포의 데이터를 가지고 있음



🞏 범주형 변수

⭘ 범주형 변수들의 빈도분석과 분포분석을 진행함

- 이 데이터에서는 모든 변수들이 고른 분포의 데이터를 가지고 있음

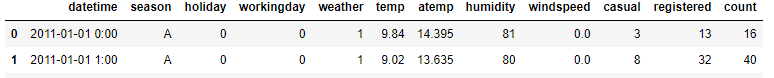


**3. 전처리(중복값, 결측값, 이상치 처리)**

**1) 중복값**

🞏 중복값 확인(duplicated)

⭘ ~~사유로 중복이 발생한 것으로 생각됨



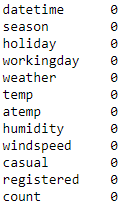
🞏 중복값 제거(drop\_duplicates)

⭘ ~~사유로 중복값은 마지막 데이터만 남기기로 함

**2) 결측값**

🞏 결측값 확인

⭘ ~~와 @@에 결측값이 n개씩 있음을 확인함



🞏 결측값 처리

⭘ ~~사유로 결측값은 0으로 대체함 / 결측값이 있는 데이터는 제거함

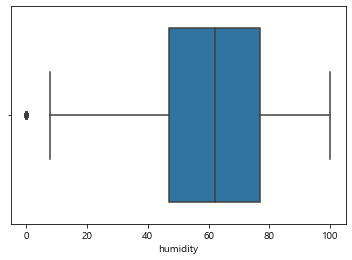
**3) 이상치**

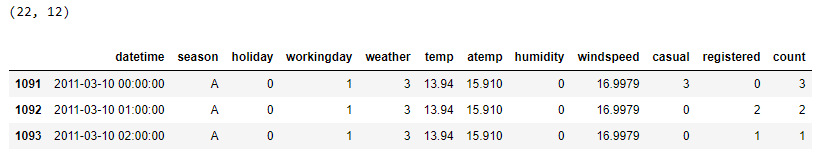
🞏 이상치 확인

⭘ 각 컬럼별로 boxplot을 그려 IQR 방식으로 이상치를 확인함

⭘ 본 보고서에는 일부 변수에 대해서만 plotting함

- humidity 변수에서는 총 22개의 이상치를 확인함





🞏 이상치 처리

⭘ 데이터를 최대한 활용하기 위해 min/max값 경계 밖의 데이터를 min/max 값으로 보정함

또는

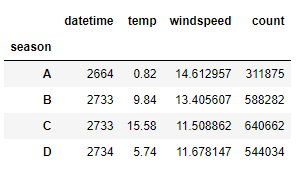
⭘ min/max값 경계 밖의 데이터는 제거함 : 제거 후 데이터는 10,864건

**4. 요약데이터로 변환(groupby)**

🞏 season 변수를 기준으로 요약데이터를 생성함

⭘ datetime의 count

⭘ temp의 최소값



**5. 파생변수 생성**

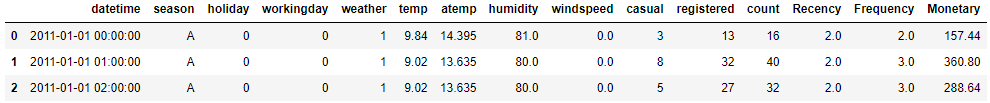
🞏 RFM의 요인을 기준으로 파생변수를 생성함 // 아니면 datetime에서 연월일, 시간대 등

⭘ Recency(최근성) : datetime이 3000일 이상이면 1, 2000일 이상면 2, 그 외는 3

⭘ Frequency(빈도) : ID별 거래 횟수(ID의 count가 10 이내면 1, 20이내면 2, 그 외는 3

⭘ Monetary(거래규모) : count와 temp의 곱(교호작용의 효과를 확인하고자 한다)

⭘ 회당 값 : ~~을 파악하기 위해 temp를 count로 나눔



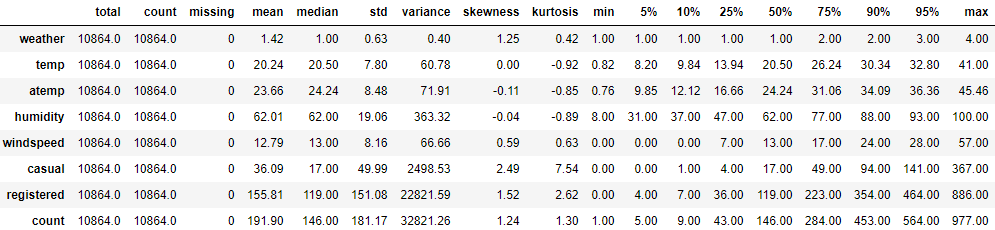
**6. 데이터 마트 DQ Check, 변수선택 및 EDA**

🞏 데이터 전처리 후 마트 DQ Check 재수행함

⭘ 연속형 변수 분포분석

- 분포가 90%이상이 동일한 변수는 모델링 시 설명력이 떨어진다고 판단하여 제외하기로 함

- 이 데이터에서는 모든 변수들이 고른 분포의 데이터를 가지고 있음

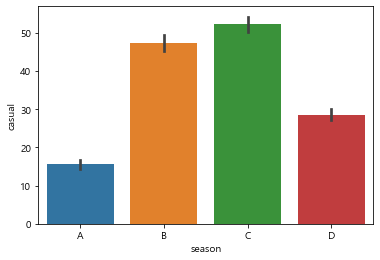


⭘ 범주형 변수 빈도, 분포분석

- 한 구간의 분포가 90%이상인 변수는 모델링 시 설명력이 떨어진다고 판단하여 제외 필요

- Frequency가 class 3에 91%가 집중되어 있어 동 변수를 제외하기로 함





🞏 EDA(시각화)

⭘ 변수별 시각화를 통해 각 변수의 특성을 분석하였음

- season class에 따라 casual의 평균 값에 차이가 있음을 확인할 수 있었음

- season이 C일 경우 casual의 평균은 50이 넘으며 전체 season 중 가장 높았으며, 반대로 A일 경우 casual은 15정도로 가장 낮았음

 - season별 빈도는 모두 2500정도로 고르게 분포하고 있음

**7. 종속변수 전처리(이항 형태로 변환 / 4개 클래스로 변환)**

🞏 종속변수의 클래스를 4개 클래스의 범주형으로 변환함

⭘ count <= 150 : 1

⭘ 150< count <= 300 : 2

⭘ 300< count <= 450 : 3

⭘ count > 450 : 4

**8. 변수 더미화**

🞏 특정 알고리즘을 활용한 모델링 과정에서의 에러 발생을 방지하기 위해 범주형 변수에 대해 더미화를 진행하였음

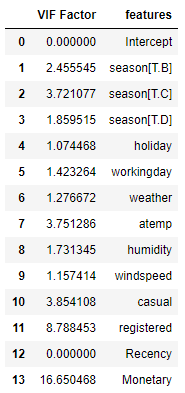
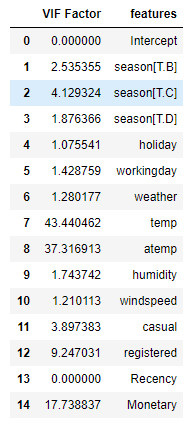
**9. 다중공선성 제거**

🞏 다중공선성은 예측변수 사이에 높은 상관관계가 있을 때 발생하며, 회귀계수 추정치의 신뢰성과 안정성에 문제를 발생시키므로 VIF(분산팽장계수)를 활용하여 사전 제거함

⭘ temp와 atemp의 VIF Factor값이 각각 43.4와 37.3으로 다중공선성 발생이 관찰되어 둘 중 temp 변수를 제거함

⭘ temp 제거 후 atemp의 VIF Factor값도 3.8로 안정화 된 것을 관찰함

⭘ Monetary변수의 VIF Factor값은 16.7로 10을 초과하나 데이터의 특성을 잘 반영하고 있으므로 유지하여 모델에 사용하기로 함



**10. Train, Test set 분리**

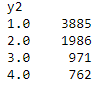
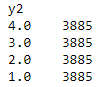
🞏 데이터 마트를 Train, Test set 7:3 비율로 분리함

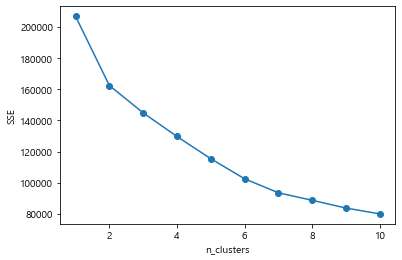
**11. 데이터 정규화**

🞏 StandardScaler로 X\_train과 X\_test를 정규화함

**12. 오버샘플링 진행**

🞏 종속변수의 class별 빈도가 불균형하여 SMOTE 방식 오버샘플링을 진행함

 🡪 

**13. 군집화**

🞏 K-means clustering을 통해 군집 수 2개~10개로 변화시켜가며 오차제곱합(SSE)의 변화와 실루엣계수를 통해 최적군집수를 찾아보았음

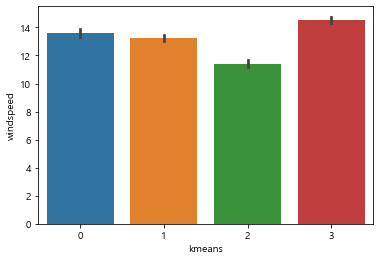
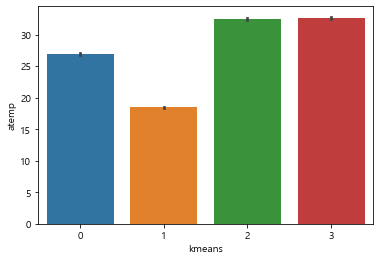
⭘ 군집 개수가 4개일 때가 최적인 것으로 판단됨

**13. 군집화 결과 프로파일링**

🞏 군집별 통계적 특성을 확인해 보았음(본 보고서에는 일부 그래프만 첨부)

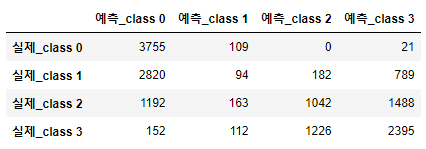
⭘ 군집2과 3의 atemp의 평균은 30을 상회하나, 군집1은 20을 밑돌았음

⭘ 군집0과 1의 windspeed의 평균은 ~~~



**14. 군집화 결과 성능평가**

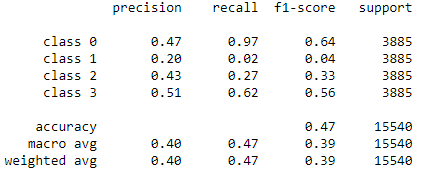
🞏 Confusion Matrix를 통해 class별 예측이 얼마나 맞았는지 확인함



🞏 classification\_report를 통해 accuracy 및 F1 score를 확인함

⭘ acc는 0.47로 나타남

⭘ F1 score는 0.39로 나타남



**14. 데이터마트에 군집화 결과 활용**

🞏 X\_train과 X\_test에 대한 군집화를 수행한 후 군집화 결과를 데이터 마트에 더하여 모델 구축에 활용하기로 함

**15. 분류 모델링(기본 모델)**

🞏 다양한 알고리즘을 활용하여 기본 모델을 구축함

⭘ Logistic Regression

⭘ Linear Discriminant Analysis

⭘ K Neighbors Classifier

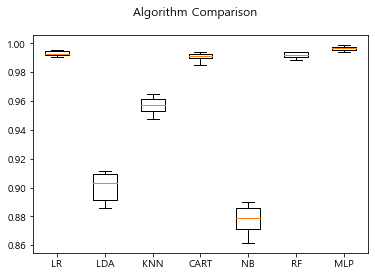
⭘ GaussianNB

⭘ RandomForest Classifier

⭘ MLP Classifier

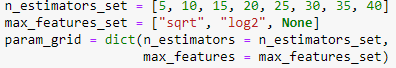
🞏 Kfold=5 Cross validation을 통한 분류 정확도 기준으로 기준으로 보았을 때 MLP Classifier가 평균 0.997로성능이 가장 높았음

🞏 Training 시간을 감안하여 평균 정확도 0.992로 두 번 째로 성능이 좋은 RandomForest Classifier를 최종 모델로 사용하기로 함



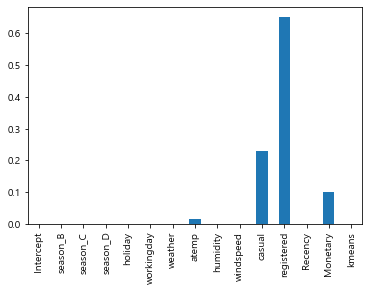
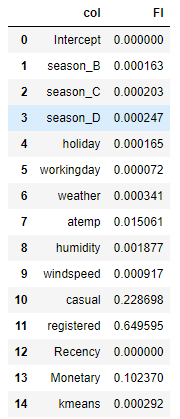
**16. 파라미터 튜닝 및 최종 모델 선정**

🞏 앞서 선택한 RandomForest Clasifier의 파라미터를 Gridsearch CV로 튜닝하여 성능을 개선함



🞏 best parameter로 max\_feature : None, n\_estimators : 30이 선정됨

🞏 본 모델의 변수중요도는 registered, casual 순으로 높았음



**17. Test set 활용하여 예측 수행 및 성능평가**

🞏 class별 F1 score는 아래와 같음

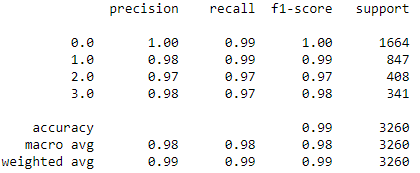
⭘ class 0 : 1.00

⭘ class 1 : 0.99

⭘ class 2 : 0.97

⭘ class 3 : 0.98

⭘ 전체 평균 : 0.99



🞏 ROC AUC

⭘ 4개 class에 대해 ROC를 도식화하고 ROC\_AUC를 계산함

