건강검진 정보를 활용한 흡연자 예측

산업인공지능 개인 프로젝트

김 윤 성
한양대학교 산업경영공학과
2018042624

Yun-Seong Kim
Industrial Management Engineering Department, Hanyang University at Ansan, Korea

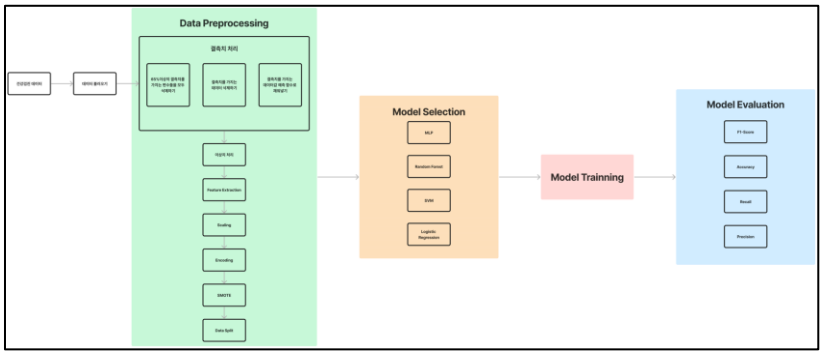
1. Introduction

이 프로젝트는 "건강검진 정보를 활용한 흡연자 예측"이라는 주제로 남자 54646 명, 여자 45354 명, 총 10 만명의 건강검진 정보와 흡연 유무(비흡연, 흡연, 금연) 정보가 있는 데이터셋을 사용한다. 데이터셋은 각 사람마다의 개인 신체 정보(성별, 나이, 키, 몸무게)와 건강검진 기록(시력, 혈압, 콜레스테롤 수치 등)의 변수로 9 개의 범주형 독립변수, 18 개의 수치형 독립변수, 1 개의 종속변수, 총 28 개의 변수로 이루어져있다.

전처리는 1)결측치 처리, 2)이상치 처리, 3)Feature Extraction, 4)Scaling, 5)Encoding, 6)SMOTE, 7)Data Split 순서로 진행된다.

흡연에 중요한 영향을 미치는 건강 변수인 콜레스테롤 수치, 치아우식증 유무 등 6 개의 변수의 데이터가 65%이상의 결측치를 지낸다. 따라서 본 프로젝트에서는 결측치를 가지는 데이터가 중요한 정보를 담는 데이터이기에 결측치 처리 방법을 다르게 한 3 개의 전처리 방법으로 성능평가를 해보았다. 결측치에 대한 전처리 방법으로는 "*65% 이상의 결측치를 가지는 변수들을 모두 삭제하기*", "*결측치를 가지는 데이터 삭제하기*", "*결측치를 가지는 데이터값 예측 함수로 채워넣기*"의 방법으로 평가를 한 후 최적의 결측치에 대한 전처리 방법을 채택하였다.

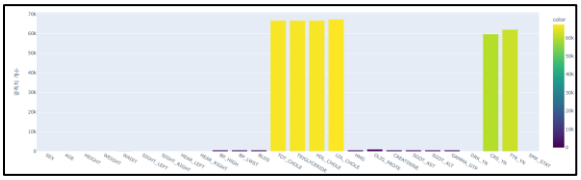
모델은 MLP, Random Forest, SVM, Logistic Regression 을 통해 최적의 모델을 선택한다. 모델에 대한 평가 지표는 F1-Score, Accuracy, Precision, Recall 값을 사용하며 흡연자를 예측하는 문제에서 비흡연자를 흡연자로 예측하는 것보다 흡연자를 놓치는 것이 더 문제이므로 Recall 값을 Precision 값보다 더 의미 있는 지표라고 판단하였다.



<Figure 1: 프로젝트 프레임워크>

2. Results

1. 결측치 개수 파악

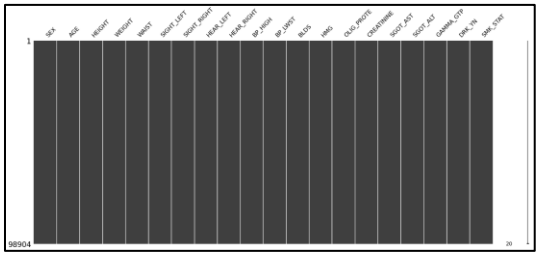


<Figure 2: 변수에 대한 결측치 수 파악>

- 100,000 개의 데이터 중 총 6 개의 변수가 65% 이상의 결측치가 존재함
- 결측치가 65% 이상인 변수들이 흡연자와 비흡연자를 명확하게 나눠주는 변수임(논문 참고)
- 따라서 이 결측치를 처리하는 방법을 3 가지로 나누어 실험을 함

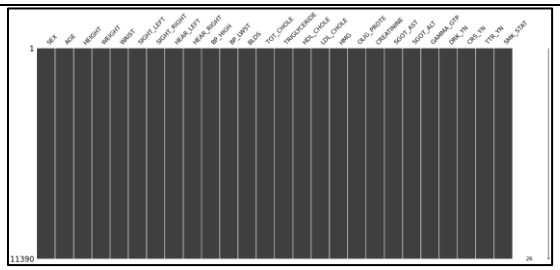
2. 결측치 처리

1) 방법 1: 65%이상의 결측치를 가지는 변수들을 모두 삭제하고 나머지 결측치는 데이터만 삭제



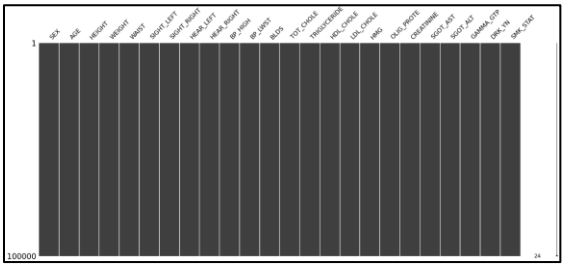
<Figure 3: 65%의 결측치를 가지는 변수 삭제 후 결측치 시각화>

2) 방법 2: 결측치를 가지는 데이터 모두 삭제하기



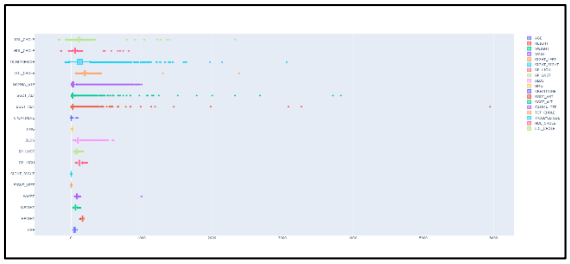
<Figure 4: 결측치를 가지는 데이터 전체 삭제 후 결측치 시각화>

3) 방법 3: 결측치를 가지는 데이터값 IterativeImputer 예측 함수로 채워넣기



<Figure 5: IterativeImputer 함수를 사용하여 수치형 변수 결측치 예측 후 시각화>

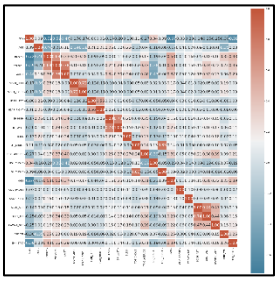
3. 이상치 처리



<Figure 6: 각 변수들 마다 분포 및 이상치 확인 시각화>

- 이상치가 존재하지만 이상치도 결과값에 중요한 역할을 한다고 판단하여 극단적인 값들만 제거

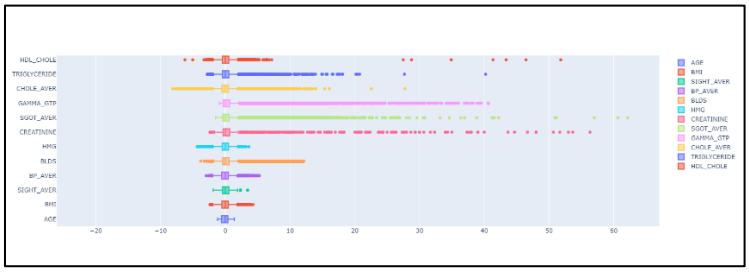
4. Feature Extraction



<Figure 7: 각 변수마다의 상관관계 Heatmap 으로 시각화>

- 성별에 따른 키와 몸무게 같은 변수와 좌시력, 우시력과 같이 상관관계가 높은 변수들이 존재하면 다중공선성의 문제가 있기 때문에 새로운 파생변수를 만들어주고 기존 변수들 삭제

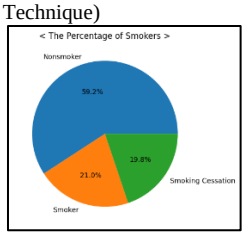
5. Scaling



<Figure 8: Robust Scaling 후 수치형 데이터 분포 시각화>

- 이 데이터셋은 이상치가 많이 존재하므로 Median 과 IQR(Interquartile range)를 이용하여 이상치에 덜 민감하게 스케일링을 해주는 Robust Scaling 을 사용함

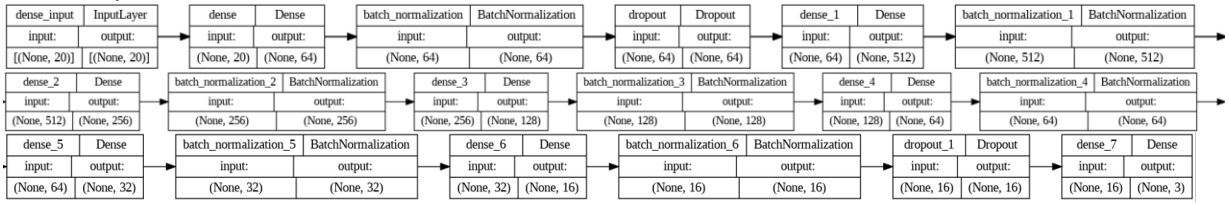
6. SMOTE (Synthetic Minority Over-sampling Technique)



<Figure 9: 종속변수에 대한 클래스 별 비율 시각화>

- 종속 변수의 클래스의 비율이 데이터 불균형 문제가 존재하는 것을 확인하여 합성 샘플링을 사용함

7. Model Selection
   1) MLP

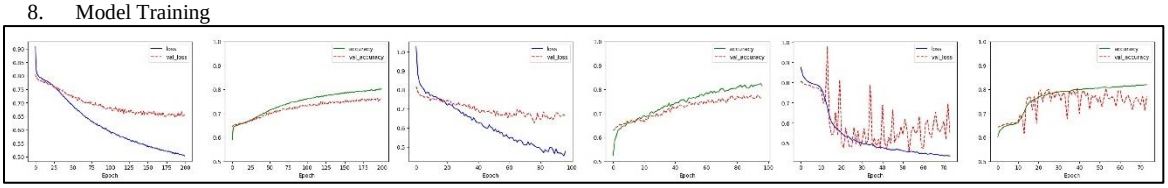


<Figure 10: MLP 모델 구조>

   2) Random Forest, SVM, Logistic Regression

- 각 모델마다 GridSearchCV 를 사용하여 하이퍼 파라미터를 튜닝한 후 모델을 생성한다.
- SVM 모델은 논문을 참고하여 k=1 로 설정한다.

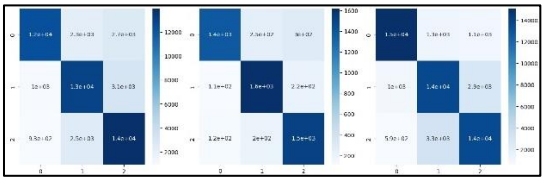
8. Model Training



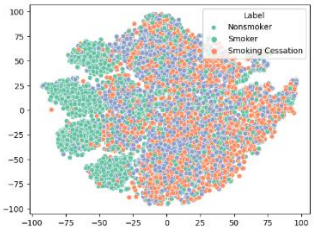
<Figure 11: 결측치 전처리를 다르게 한 3 가지 데이터셋에 대한 MLP 모델 Loss, Accuracy 그래프>

- 방법 3 으로 결측치를 처리할 경우 매우 불안정하게 모델이 예측하는 것을 볼 수 있음

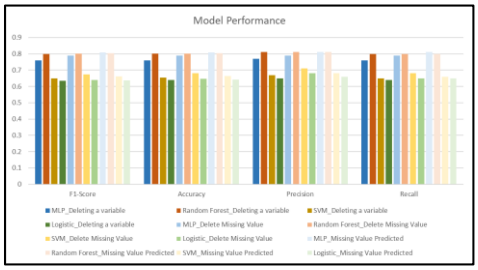
9. Model Evaluation



<Figure 12: 결측치 전처리를 다르게 한 3 가지 데이터셋에 대한 MLP 모델 예측 결과 Confusion Matrix>



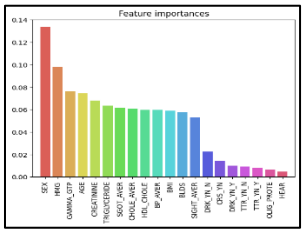
<Figure 13: 랜덤포레스트 모델을 t-SNE 사용해 2 차원 축소하여 시각화>



<Figure 14: 각 모델의 성능을 비교한 그래프>

- 각 모델들을 비교한 결과 전체적으로 MLP 모델과 Random Forest 모델이 평가 지표가 우수하였다.

- Fig.11 에서 볼 수 있듯이 결측치 방법들 중 방법 3 데이터셋을 활용한 모델이 성능은 좋았지만 학습의 불안정으로 최적 모델에서 제외하였다.
- 전반적으로 방법 1 보단 방법 2 의 데이터셋을 사용하여 예측한 모델들이 성능이 더 잘 나오는 것을 볼 수 있어 최적의 결측치 처리 방법은 방법 2 로 선택하였다.
- 방법 2 를 쓴 모델 중 가장 성능이 좋게 나온 Random Forest 모델을 선택하였다.
- 참고 논문에서도 "데이터의 개수가 평가에 미치는 영향이 작다"고 되어 있고 흡연에 미치는 영향이 큰 변수를 삭제하지 않았기 때문에 방법 2 데이터셋이 가장 좋은 성능을 보여준다고 결론을 지었다.



<Figure 15: Random Forest 의 Feature importances>

- 실제로 최적의 모델로 선택한 방법 2 데이터셋을 사용한 Random Forest 모델의 Feature importances 에서 방법 1 에서 삭제한 변수인 콜레스테롤과 트리글리세라이드 수치 변수 중요도가 높게 나온 것을 볼 수 있다.

3. How to run

이 프로젝트의 파일은 하나의 데이터셋과 3 개의 .ipynb 파일로 구성 되어있다. 결측치 처리 방법 1~3 이 다 다른 파일로 구성되어 있다. 3 개의 데이터 분석 파일은 결측치를 처리하는 전처리 부분 외엔 대부분 비슷한 전처리 및 모델링 과정을 거쳤다. 이 때문에 파일을 볼때는 하나의 파일만 전처리 과정을 보고 나머지 파일은 모델 평가지표를 보면 된다.

해당 파일은 Google Colab 환경에서 돌릴 시 따로 패키지 설치가 필요 없다. 압축 파일과 함께 있는 데이터셋을 파일의 (2) Load the dataset 절차에서 넣어준 후 차례대로 코드를 돌리면 된다.

주의할 점은 랜덤 포레스트 모델의 하이퍼 파라미터 튜닝 하는 과정에서 사용하는 GridSearchCV 와 SVM 을 학습시킬 때 많은 시간이 소모되므로 이미 돌린 값을 참고하길 바란다.

4. Code description

전반적인 프로젝트 흐름은 2. Result 에서 적어놨으므로 여기선 생소한 코드들에 대한 설명만 하겠다.

1. **결측치 시각화 라이브러리 msno.matrix(df)**
- **변수마다 결측치를 시각화해주는 코드로 박스 중간에 흰색 가로줄들이 있다면 그 위치에 결측치가 있다는 의미이다.**
2. **Plotly 라이브러리로 시각화 fig.add_trace(go.Box(x=df['AGE'], name='AGE'))**
- **이 라이브러리는 그래프를 더 역동적으로 표현해주므로 아래 시각화된 그래프를 누르면 보고 싶은 값에 대해 그래프를 축소시키거나 제외시킨 후 볼 수 있다.**
3. **방법 3 결측치 처리 함수 IterativeImputer**

```
[12]  # 결측치 예측 함수
      imputer_mice = IterativeImputer(random_state=83)
      df = imputer_mice.fit_transform(df)
```

- **IterativeImputer 을 써서 수치형 변수 결측치를 예측하고 array 형태로 바뀐 데이터를 다시 DataFrame 으로 바꿔준다.**
4. **청력에 대한 변수 처리**

```
[23]  # 범주형 청력 변수 하나로 만들기 -> 1 정상, 0 비정상
      df.loc[df['HEAR_LEFT'] == 2, 'HEAR_LEFT'] = 0
      df.loc[df['HEAR_RIGHT'] == 2, 'HEAR_RIGHT'] = 0

      df['HEAR'] = df['HEAR_LEFT']*df['HEAR_RIGHT']
      df = df.drop(['HEAR_LEFT', 'HEAR_RIGHT'], axis = 1)
```

- **청력 좌, 우로 1(정상), 2(비정상)인 값을 비정상일 때 0 으로 두고 두 값을 곱해서 정상일 때 1, 한쪽이라도 아플시 0 비정상으로 파생 변수를 만들어 주었다.**
5. **RobustScaler**

```
[27]  # 이상치가 존재하는 데이터이므로 이상치에 덜 민감한 RobustScaler로 범위 맞춰주기
      scaler = RobustScaler()
      df_num = pd.DataFrame(scaler.fit_transform(df[df_num]), columns=df_num)

[28]  df = pd.concat([df_num, df[df_cat]], axis=1)
```

- **이상치가 많은 데이터이므로 RobustScaler 을 사용하여 Scaling 을 하였다.**
6. **SMOTE**

```
[32]  # 데이터 불균형 문제 해결하기
      smote = SMOTE()

      X, y = smote.fit_resample(X, y)
```

- **데이터 불균형 문제를 잡아주기 위해 소수 클래스의 샘플을 기존 데이터셋에서 복제하거나 왜곡, 변형하여 새로운 샘플을 생성하는 합성샘플링을 사용하였다.**

7. **MLP 모델 구성**

```
# Input layer
model.add(Dense(64, input_dim=X.shape[1], activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.2))

# Hidden layer
model.add(Dense(512, activation='relu'))
model.add(BatchNormalization())
model.add(Dense(256, activation='relu'))
model.add(BatchNormalization())
model.add(Dense(128, activation='relu'))
model.add(BatchNormalization())
model.add(Dense(64, activation='relu'))
model.add(BatchNormalization())
model.add(Dense(32, activation='relu'))
model.add(BatchNormalization())
model.add(Dense(16, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.3))

# Output layer
model.add(Dense(units = 3, activation='softmax'))
```

```
# 최적화 함수 정의하기
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              # 평가기준을 정확도로 넣음
              metrics=['accuracy'])
```

```
%%time

early_stop = EarlyStopping(monitor='val_loss', patience=20, verbose=2,
                           mode='min', restore_best_weights=True)

history = model.fit(X_train, y_train,
                    epochs = 200,
                    batch_size=256,
                    validation_split=0.2,
                    callbacks=[early_stop])
```

- 6 개의 Dense 레이어와 바로 뒤에 BatchNormalization 레이어를 사용하여 학습 과정에서 발생하는 그래디언트 소실 문제를 해결하고, 학습 속도를 향상시켰다. 또한 2 개의 Dropout 레이어를 사용하여 모델 과적합 문제를 방지한다.
- 모든 은닉층에는 활성화 함수로 relu 함수가 사용되었으며, 출력층엔 다중 클래스 분류 문제이므로 softmax 활성화 함수를 사용하였다.
- 이 모델에 adam 경사하강법을 사용하였고 sparse_categorical_crossentropy 를 손실함수로 사용하여 다중 클래스 분류를 해준다.
- 학습은 epoch 를 200 으로 설정하고 EarlyStopping 을 통해 과적합을 방지한다.

8. 훈련과정에서 모델의 과적합이 되었는지 볼 수 있는 그래프 시각화 코드

```
# Loss 그래프
# loss는 수치이므로 1을 넘어갈 수 있음 -> 확률이랑 학각 ㄴㄴ
plt.subplot(1,2,1)
plt.plot(history.history['loss'], 'b-', label='loss')
plt.plot(history.history['val_loss'], 'r--', label='val_loss')
plt.xlabel('Epoch')
plt.legend()

# Accuracy 그래프
# 확률 값임
plt.subplot(1,2,2)
plt.plot(history.history['accuracy'], 'g-', label='accuracy')
plt.plot(history.history['val_accuracy'], 'r--', label='val_accuracy')
plt.xlabel('Epoch')
plt.ylim(0.5, 1)
plt.legend()
```

9. 랜덤포레스트 모델의 GridSearchCV 를 사용한 하이퍼 파라미터 튜닝 코드와 feature importance 시각화 코드

```
# define the parameters grid
param_grid = {
        'n_estimators': [100, 200, 300],
        'max_depth': [None, 10, 20],
        'min_samples_split': [2, 5, 10],
        'min_samples_leaf': [1, 2, 4]
            }

# create the grid
grid_tree = GridSearchCV(RandomForestClassifier(), param_grid, cv = 5, scoring= 'accuracy')
# the cv option will be clear in a few cells

#training
grid_tree.fit(X_train, y_train)
#let's see the best estimator
print(grid_tree.best_estimator_)
#with its score
print(np.abs(grid_tree.best_score_))
```

```
# feature importance 추출
importances = RFC.feature_importances_
indices = np.argsort(importances)[::-1]

# 시각화
plt.figure()
plt.title("Feature importances")
colors = sns.color_palette('hls',len(importances))
plt.bar(range(X_train.shape[1]), importances[indices],
        color=colors, align="center")
plt.xticks(range(X_train.shape[1]), X_train.columns[indices], rotation=90)
plt.xlim([-1, X_train.shape[1]])
```

5. References

데이터셋

- 2022 년 Field Camp 경진대회 제공 데이터셋

참고 논문

- 김진옥, “현재 흡연자와 비흡연자의 혈중지질 수준 비교”, 연세대 보건대학원, 2002.12, p18~
- 정인경, “제 5 기 국민건강영양조사 자료 중 남성에서 흡연 상태와 고밀도지단백-콜레스테롤 농도의 관련성”, 호남대
- 이혜숙, “정상 성인에서 흡연, 일반적 특성과 혈청지질과의 상관관계”, 한국 간호 교육 학회지, 2004
- 성동경, “청소년 흡연이 구강질환에 미치는 영향”, 연세대 보건대학원, 2000.06, p32~
- 박정진, “비만과 치아우식증의 상관성에 관한 연구”, 연세대 보건대학원, 2002.12, p13~
- 윤지선, “흡연자 판별을 위한 모형별 성능 검증”, 고려대 컴퓨터정보통신대학원, 2019
- 심경란, “성인 흡연자, 간접흡연자, 비흡연자의 DPOAE 비교”, 대구가톨릭대학교 의료보건과대학원, 2017
- 한주희, “성인 남자의 흡연과 BMI 와의 관계”, 연세대 보건대학원, 1998

참고 자료

- 김미리, “금연하면 콜레스테롤 낮춰진다”, Medical Observer, 2010.12, http://www.monews.co.kr/news/articleView.html?idxno=38328
- 이문에 기자, “좋은 콜레스테롤(HDL 콜레스테롤) 수치 낮다면 금연부터 시작하세요”, 푸드앤메드, http://www.foodnmed.com/news/articleView.html?idxno=12511
- 진주창, “중성지방, 트리글리세라이드 수치 낮추는 법”, 건강상식 블로그, https://blog.naver.com/cvdasa4cfx/222561061563
- 국가지표체계, 현재흡연율, https://www.index.go.kr/unify/idx-info.do?idxCd=4237
- 통계청, 연령별 성별 흡연 관련 문항, KOSIS, https://kosis.kr/statHtml/statHtml.do?orgId=350&tblId=DT_35007_N045
- 한상헌, “흡연하면 청력도 나빠진다… 비흡연자에 비해 1.7 배 저하”, 서울경제, https://www.sedaily.com/NewsView/1RY41VC6QR