3주차 예비보고서

전공 : 경영학과 학년 : 4학년 학번 : 20190808 이름 : 방지혁

1. feature engineering의 개념에 대해서 조사하고, 주요한 기법 3가지를 자세하게 작성해보세요.

우선 feature engineering이란 원시 데이터에 있는 기존의 특징을 변환하거나 새로운 특성을 생성하여 모델의 성능을 향상 및 정확도를 높이는 과정입니다. 중요한 기법 3가지에 대해 서술해보겠습니다.

첫번째는 스케일링입니다. 수집한 데이터가 서로 다른 스케일을 가지기에 같은 스케일로 맞춰주는 것입니다. 가장 유명한 스케일링 기법으로는 standard scaling과 min-max scaling이 존재합니다. Standard scaling이란 평균을 0, 표준 편차를 1로 만드는 기법입니다. Min-max scaling이란 값을 0과 1 사이로 조절하는 것입니다.

두번째는 encoding입니다. 텍스트로 된 범주형 데이터를 숫자로 변환하는 방법입니다. 이는 모델이 문자열 데이터를 이해할 수 있도록 하기 위해서 입니다. One-hot encoding과 label-encoding이 존재합니다. One-hot encoding이란 각 범주를 이진 변수로 변환하는 것입니다. 범주 간에 관계성이 존재하지 않을 때 유용합니다. 광주가 [1, 0, 0], 대구가 [0, 1, 0], 부산이 [0, 0, 1] 이런 식으로 저장되는 것입니다. 이렇듯 대부분의 값이 0이기에 범주가 너무 많아지면 비효율적일 수도 있습니다. Label-encoding이란 각각의 값을 정수로 변환하는 것입니다. 광주를 1, 대구를 2, 부산을 3 이런 식으로 변환하는 것입니다. 이를 통해 이전과 달리 메모리 사용을 아낄 수 있어 효율적이며 만약 범주 간의 관계성이 존재한다면 그 관계까지 모델에 반영할 수 있습니다.

마지막으로는 특성 생성이 있습니다. 기존 데이터를 바탕으로 새로운 의미 있는 특성을 만들어 나가는 것입니다. 이는 모델이 관계성 혹은 패턴을 더 잘 이해하도록 도와줍니다. 예를 들자면 총점을 과목 수로 나눈다면 평균 점수 라는 새로운 비율 특성을 도출해낼 수 있고 날짜를 통해 요일, 월, 계절이라는 시간 특성을 도출해낼 수 있습니다. 이는 예측 성능을 향상시킬 수 있습니다.

2. 데이터에 결측치가 포함되어 있으면 발생할 것 같은 문제점을 생각해보고, 어떻게 처리하면 좋을지 방법을 조사해서 정리해보세요.

결측치에 대한 설명부터 해야 할 것 같습니다. NULL 혹은 NA라고도 하며 데이터에 값이 없는 것을 말합니다. 이는 분석에 있어서 문제점을 야기합니다. 처음부터 특정 항목에 대해 응답을 하지 않거나, 데이터 입력 과정에서 오류가 발생했을 수도 있고, 저장이나 전송 과정에서 손실이 일어났을 수도 있습니다. 결측치가 있다면 이를 허용하지 않는 모델인 scikit-learn 같은 경우 NaN을 처리하지 못해 아예 실행되지 않을 수 있습니다. 또한, 이 결측치가 무작위로 발생하지 않고 특정 패턴으로 발생한다면 예측이 왜곡될 수 있습니다.

우리는 이를 해결하기 위한 첫번째 방법으로 삭제할 수 있습니다. 삭제에는 행 삭제와 열 삭제가 있을 수 있습니다. 행 삭제의 경우 말 그대로 결측치가 포함된 행 전체를 삭제하는 경우로 행 삭제를 하여도 데이터가 충분히 많고 결측치를 포함하는 행이 적을 경우 유용합니다. 열 삭제의 경우 결측치가 특정열에 집중되어 있고 중요성이 낮을 경우 유용합니다.

또한, 두번째 방법으로는 대체를 할 수 있습니다. 결측치를 평균, 중앙값, 최빈값으로 대체할 수 있습니다. 이는 가장 간단하고 보편적이며 분포를 크게 왜곡하지 않습니다. 고정값으로 대체할 수도 있는데 이는 결측치가 의미 있는 정보일 때 유용합니다. 예측 모델을 이용한 대체도 있는데 이는 연산 비용이 많이 든다는 단점이 존재합니다.

3. Random Forest 모델의 특징에 대해 정리해보고, scikit-learn에서 Random Forest를 사용하는 방법에 대해서 설명해보세요. 예시 코드 작성하시면 좋습니다.

우선 random forest란 독립성을 가진 여러 decision tree들을 생성하고 이 tree들의 예측 결과들을 합쳐 최종 예측을 도출하는 모델입니다. 특징들에 대해 설명하자면 과적합을 방지할 수 있습니다. 단일 트리의 경우 과적합되기 쉽지만 random forest 같은 경우 여러 트리의 결과를 평균화하여 무작위 특성을 선택하기 때문에 이를 해결할 수 있습니다. 또한 특성 중요도를 제공합니다. 학습 후 각 특성이 어떤 역할을 얼만큼 중요하게 수행했는지 수치로 제공하기 때문에 영향력을 파악할 수 있습니다. 또한 다양한 데이터를 처리할 수 있습니다. 범주형, 수치형 데이터 모두 처리 가능

하며 결측치 처리도 가능합니다.

Random forest를 사용하기 위한 방법은 다음과 같습니다. 우선 sklearn.ensemble 모듈의 RandomForestClassifier (분류) 혹은 RandomForestRegressor (회귀)를 import해야 합니다.

```
from sklearn.ensemble import RandomForestClassifier

from sklearn.ensemble import RandomForestRegressor
```

이후 예시 데이터를 생성해줍니다.

```
X, y = make_classification(n_samples=1000, n_features=20, n_informative=10, n_redundant=10,

random_state=42)
```

여기서 n_samples는 데이터 샘플 개수, n_features는 특징 개수, n_informative는 유용한 특징 수, random_stat는 난수 생성 시드를 의미합니다.

이후 훈련 데이터와 테스트 데이터를 분리합니다.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

이후 모델 초기화 및 학습의 과정이 이루어집니다.

```
model = RandomForestClassifier(n_estimators=100, max_features='sqrt', random_state=42)

model.fit(X_train, y_train)
```

n_estimators는 생성할 트리의 개수, max_features는 노드에서 분할에 사용할 특징의 최대 개수, random_state는 앞서 말했듯 재현성 확보를 위한 시드 값입니다.

마지막으로 예측 및 정확도 평가가 이루어집니다.

```
y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
```