신용카드사기 예측

학번:2218064

이름: 나민하

Github address:
https://github.com/2218064/Credit-card-fraud-Prediction

1. 안전 관련 머신러닝 모델 개발의 목적

학습 모델 활용 대상

신용카드 사기 탐지 모델은 금융 산업에서 활용된다. 이 모델은 신용카드 거래에서 사기를 식별하고 차단하는데 사용된다. 이 모델은 정상적인 신용카드 거래 패턴을 학습하고, 이상 행동을 식별하여 사기 거래를 탐지한다. 이를 통해 신용카드 사기로부터 고객을 보호하고 금융 기관의 손실을 최소화할 수 있다. 그리고 대규모의 데이터를 신속하게 처리하고 패턴을 식별할 수 있으므로, 자동화된 신용카드 사기 탐지 시스템에 효과적으로 통합된다. 이 모델을 통해 금융 기관은 고객의 자산을 보호하고, 안전한 거래 환경을 제공함으로써 고객들에 대한 신뢰를 강화할 수 있다. 또 금융기관이 미래에 발생할 가능성이 있는 사기를 예측하고 방지하는 데 도움이 된다.

데이터의 어떠한 독립 변수를 사용하여 어떠한 종속 변수를 예측하는 지

'pd.read_csv()' 함수를 사용하여 csv 파일에서 데이터를 불러온다.

종속 변수인 'fraud'를 제외한 나머지 열은 독립 변수로 설정한다.

StandardScaler 를 사용하여 숫자형 독립 변수를 정규화한다.

'train_test_split' 함수를 사용하여 전체 데이터를 학습 데이터와 테스트 데이터로 나눈다.

'LogisticRegression' 클래스를 사용하여 로지스틱 회귀 모델을 생성하고, 학습 데이터를 사용하여 모델을 학습한다.

학습된 모델을 사용하여 테스트 데이터에 대한 종속 변수('fraud')를 예측한다.

다양한 평가 지표를 사용하여 모델의 성능을 측정한다. 이 평가 지표로는 정확도, 정밀도, 재현율, F1 점수가 있다.

'confusion_matrix' 함수를 사용하여 모델의 예측 결과와 실제 결과를 비교한 혼동 행렬을 출력한다.

개발의 의의: 학습 모델 개발 시 어떠한 가치를 생성하는지

주된 가치는 모델이 신용카드 거래에서 사기를 탐지하는 데에 있다. 모델은 학습 데이터를 기반으로 사기 거래의 패턴을 학습하고, 이를 새로운 데이터에 적용하여 사기 여부를 예측한다. 따라서 금융기관은 사기 행위를 미리 감지하고 적절한 조치를 취할 수 있다.

2. 안전 관련 머신러닝 모델의 네이밍의 의미

이 모델의 네이밍은 신용카드 사기 예측 이 의미는 말 그대로 신용카드 사기를 예측하는 머신러닝이라는 의미이다.

3. 개발 계획

데이터에 대한 요약 정리 및 시각화

데이터셋은 주로 숫자형 특성을 가지고 있다. 'fraud'열은 이진 분류이며, 사기 거래인지 여부를 나타낸다. 먼저, 각 특성의 기본 통계량과 데이터의 구조를 확인한다. 데이터 시각화는 사기와 정상 거래의 비율을 시각화하여 클래스 불균형 여부를 확인한다. 그리고 상관 행렬을 통해 특성 간의 상관 관계를 시각화하여 모델에 사용할 특성을 선택하는 데 도움을 준다.

데이터 전처리 계획

데이터에서 결측치가 있는지 확인하고 있다면 처리한다. 그리고 이상치를 확인하고 필요하면 처리한다. 또, 사기 거래와 정상 거래의 클래스 불균형이 있는 경우, 적절한 리샘플링 기법을 사용한다. 마지막으로 불필요한 특성이나 다중공선성이 있는 특성을 제거한다.

어떠한 머신러닝 모델을 사용할 것인지 (해당 머신러닝 모델의 이론 추가)

랜덤 포레스트를 사용할 것이다. 랜덤 포레스트는 회귀 및 분류 문제를 해결하는데 사용할 수 있는 머신 러닝 기법이다. 여러 개의 의사 결정 트리로 구성되는데 각각의 의사 결정 트리는 출력 결과를 내놓는다. 최종 결과를 얻기 위해 앙상블 기법 중 하나인 배깅을 사용한다.

머신러닝 모델 예측 결과가 어떠할 지

여러 개의 의사 결정 트리를 사용하므로 각 트리의 예측을 조합하여 최종 예측을 생성한다. 이는 앙상블의 특성으로, 모델이 일반적으로 안정적이고 과적합을 피할 수 있게 한다.

사용할 성능 지표

정확도(Accuracy) : 전체 예측 중 올바르게 예측한 비율

정밀도(Precision) : 사기 거래로 예측한 것 중에서 실제 사기 거래의 비율

재현율(Recall) : 실제 사기 거래 중에서 모델이 사기 거래로 올바르게 예측한 비율

F1 Score : 정밀도와 재현율의 조화 평균. 모델의 균형을 나타낸다.

성능 검증 방법 계획 등

교차 검증 : 데이터를 여러 부분으로 나누어 모델을 여러 번 학습하고 평가하여 일반화 성능을 높인다.

혼동 행렬 분석 : 모델의 성능을 더 자세히 평가하고 어떤 클래스에서 오류가 발생했는지 확인한다.

ROC 곡선 및 AUC : 모델의 분류 성능을 시각화하고 평가한다.

4. 개발 과정

계획 후 실제 학습 모델 개발 과정을 기록 (*개발 과정 캡쳐 필수)

데이터 전처리 (간단히 숫자형 특성만 사용하고 정규화를 수행한다.)

```
X = data.drop('fraud', axis=1)
y = data['fraud']

scaler = StandardScaler()
X = scaler.fit_transform(X)
```

학습 데이터와 테스트 데이터로 나누기

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

로지스틱 회귀 모델 생성 및 학습

```
model = LogisticRegression()
model.fit(X_train, y_train)
```

예측

```
y_pred = model.predict(X_test)
```

모델평가

```
print(f'Accuracy: {accuracy_score(y_test, y_pred)}')
print(f'Precision: {precision_score(y_test, y_pred)}')
print(f'Recall: {recall_score(y_test, y_pred)}')
print(f'F1 Score: {f1_score(y_test, y_pred)}')
```

혼동 행렬 출력

```
conf_matrix = confusion_matrix(y_test, y_pred)
print(f'Confusion Matrix:\n{conf_matrix}')
```

각 함수는 어떻게 동작하는 지 구체적으로 설명

데이터 불러오기: pd.read_csv() 함수를 사용하여 CSV 파일에서 데이터를 읽어온다. r"C:\Users\82105\OneDrive\문서\GitHub\Credit-card-fraud-Prediction\data\card_transdata.csv" 경로의 CSV 파일을 읽어와 DataFrame 으로 저장한다

데이터 전처리: X = data.drop('fraud', axis=1)를 통해 종속 변수인 'fraud'를 제외한 나머지 열을 독립 변수로 설정한다.

y = data['fraud']를 통해 종속 변수를 설정한다.

scaler = StandardScaler()를 사용하여 독립 변수를 정규화한다.

(fit_transform 을 사용하여 평균과 표준편차를 계산하고 데이터를 변환한다.)

학습 데이터와 테스트 데이터 분리: train_test_split 함수를 사용하여 전체 데이터를 학습 데이터와 테스트 데이터로 나눈다. test_size=0.2 는 테스트 데이터의 비율을 나타낸다.

로지스틱 회귀 모델 생성 및 학습: LogisticRegression()을 사용하여 로지스틱 회귀 모델을 생성한다.

model.fit(X_train, y_train)을 통해 학습 데이터를 사용하여 모델을 학습한다.

예측: y_pred = model.predict(X_test)를 사용하여 테스트 데이터에 대한 종속 변수('fraud')를 예측한다.

모델 평가: 다양한 평가 지표를 사용하여 모델의 성능을 측정한다.

혼동 행렬 출력: confusion_matrix 함수를 사용하여 모델의 예측 결과와 실제 결과를 비교한 혼동 행렬을 출력한다.

에러 발생 지점 및 해결 과정

위 코드에서는 주로 데이터 전처리, 모델 학습, 예측, 평가 등의 기본적인 단계를 포함하고 있어서 큰 에러가 발생하지 않을 것으로 예상된다. 그러나 경로나 파일의 존재 여부 등에서 발생할 수 있는 에러에 대비하여 다음을 확인할 수 있다. 파일 경로(C:\Users\82105\OneDrive\문서\GitHub\Credit-card-fraud-Prediction\data\card_transdata.csv)가 정확한지 확인한다.

파일이 실제로 존재하는지 확인한다.

데이터에 결측치나 이상치가 있는지 확인한다.

학습 모델의 성능 평가

accuracy_score, precision_score, recall_score, f1_score 등의 평가 지표를 통해 모델의 성능을 측정한다.

혼동 행렬(confusion matrix)을 통해 각 클래스에 대한 예측과 실제 결과를 비교한다. 성능 지표를 통해 모델의 강점과 약점을 파악하고 필요에 따라 모델을 조정하거나 다른 모델을 고려할 수 있다.

결과 시각화

PDF 파일로

5. 개발 후기

개발 후 느낀 점 설명

개발하면서 데이터를 먼저 충분히 이해하는 것이 중요하다는 것을 깨달았다. 특히, 각 특성의 의미와 데이터의 분포에 대한 이해는 모델의 성능을 향상시키는데 도움이 되었다. 로지스틱 회귀 모델은 간단하면서도 효과적인 모델 중 하나이지만, 다른 모델들과의 비교 및 실험도 중요하다는 것을 알았다. 그리고 정규화를 통해 특성 간의 단위 차이를 줄이고 모델이 안정적으로 학습할 수 있게 되었다. 또 모델을 개발한 후에도 지속적인 개선이 필요하다는 것을 인지했다. 사용된 라이브러리인 scikit-learn 의 기능과 활용법을 더 깊이 학습하고 활용할 수 있는 능력을 길러야되겠다고 생각하였다.