**Machine Learning (Autumn 2023)**

**Homework 6th week-2 (October 21)**

**Student ID 2022712151**

**Name 정지원**

**Instruction: This report is a data analysis report using LightGBM and is the second task of week 6 of Machine Learning Class**

**(1)** 데이터 소개

이 데이터셋은 Aihub의 상하수도 데이터로, 총 5가지 레이블을 가지고 있다. : out, in. noise, other, normal. 해당 데이터셋을 활용하여 데이터 분석을 진행했다.

**(2)** 데이터 전처리

**(2.1)** 레이블 변환

초기에는 문자열 형태의 레이블을 0,1, 2, 3, 4로 변환하여 학습에 활용하였다.

**Code:**

|  |
| --- |
|  |

**(2.2)** 데이터

데이터를 훈련 및 검증용으로 나누고, Min-Max Scaling을 통해 데이터를 정규화했다.

**Code:**

|  |
| --- |
| from sklearn.model\_selection import train\_test\_split  target = y\_data  x\_train,x\_valid,y\_train,y\_valid = train\_test\_split(data,target,test\_size=0.2,shuffle=True,stratify=target,random\_state=42)  from sklearn.preprocessing import MinMaxScaler  minmaxscaler = MinMaxScaler()  train\_data\_scale = minmaxscaler.fit\_transform(x\_train.iloc[:,3:])  train\_data\_scale = pd.DataFrame(train\_data\_scale,columns=x\_train.iloc[:,3:].columns)  test\_data\_scale = minmaxscaler.fit\_transform(x\_valid.iloc[:,3:])  test\_data\_scale = pd.DataFrame(test\_data\_scale,columns=x\_valid.iloc[:,3:].columns) |

**(3)** 모델 학습

**(3.1)** LightGBM 모델

LightGBM 모델을 활용하였으며, GridSearchCV를 통해 최적의 하이퍼 파라미터를 탐색했다.

* Learning\_rate : Boosting learning rate. You can use callbacks parameter of fit method to shrink/adapt learning rate in training using reset\_parameter callback. Note, that this will ignore the learning\_rate argument in training.
* N\_estimators : Number of boosted trees to fit.
* Max\_depth : Maximum tree depth for base learners, <=0 means no limit.
* Boosting\_type : ‘gbdt’, traditional Gradient Boosting Decision Tree. ‘dart’, Dropouts meet Multiple Additive Regression Trees. ‘rf’, Random Forest.

**Code :**

|  |
| --- |
| from sklearn.model\_selection import GridSearchCV  from lightgbm import LGBMClassifier  lgb = LGBMClassifier(random\_state=42)  lgb\_paramgrid={  'learning\_rate': [0.05,0.1,0.15, 0.2],  'n\_estimators': [100,200,400,600,1000],  'max\_depth':[4,6,8,10],  'boosting\_type' : ['gbdt', 'dart'], # for better accuracy -> try dart  }  lgb\_grid = GridSearchCV(lgb,param\_grid=lgb\_paramgrid,scoring='accuracy',verbose=3).fit(train\_data\_scale,y\_train)  {'boosting\_type': 'dart', 'learning\_rate': 0.2, 'max\_depth': 10, 'n\_estimators': 1000} |

**(4)** 모델 평가

학습된 모델을 검증 데이터에 적용하여 정확도를 평가했다.

**Codes and Results :**

|  |
| --- |
| Test Accuracy: 0.879995 |

**(5)** 데이터 시각화

데이터 분석 결과를 더 명확하게 전달하기 위해 시각화를 진행했다.

**(5.1)** 피처 중요도 시각화

LightGBM모델에서 얻은 feature importance를 시각화하여 어떤 feature가 모델에 영향을 주는지 확인할 수 있다.

**Codes and Resuls :**

|  |
| --- |
| # 피처 중요도 시각화  plot\_importance(lgb\_grid.best\_estimator\_, figsize=(10, 8))  plt.show() |

**(5.2)** 예측 결과 시각화

**Codes and Results:**

|  |
| --- |
| # 예측 결과 시각화  plt.figure(figsize=(10, 6))  sns.countplot(x='leaktype', data=pd.DataFrame({'True': y\_valid, 'Predict': lgb\_pred}))  plt.title('True vs Predicted Leak Types')  plt.show() |

**(6)** 결과 및 토의

**Results :**

|  |
| --- |
| LightGBM모델을 사용한 결과, 모델은 테스트 데이터에서 88%의 정확도를 보였다.  . |