黃偉祥 X1136010

Model description

Combination of `StackingClassifier` , `XGBClassifier` , `LGBMClassifier` , `RandomForestClassifier` , `AdaBoostClassifier`

Use Lightgbm, XGBoost, Random Forest to do the first decision and use stacking classifier(Final model) to do the final decision.

Final model use AdaBoost, and Random Forest as the base of AdaBoost.

```
# Model information
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
from sklearn.ensemble import StackingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier

def get_models(scale_pos_weight):
    rf_model = RandomForestClassifier(
        n_estimators=201,
        max_depth=7,
        class_weight={0: 1, 1: scale_pos_weight},
        max_samples=0.8,
        max_features=0.7,
        min_samples_leaf=1,
        random_state=42,
        n_jobs=-1
    )

    xgb_model = XGBClassifier(
        n_estimators=201,
        max_depth=7,
        learning_rate=0.1,
        scale_pos_weight=scale_pos_weight,
        subsample=0.8,
        colsample_bytree=0.7,
        gamma=0.1,
        tree_method='hist',
        # use_label_encoder=False,
        eval_metric='logloss',
        random_state=42
    )

    lgb_model = LGBMClassifier(
        n_estimators=201,
        max_depth=7,
        learning_rate=0.1,
        class_weight={0: 1, 1: scale_pos_weight},
        subsample=0.8,
        colsample_bytree=0.7,
        reg_alpha=0.1,
        reg_lambda=0.1,
        random_state=42
    )

    meta_model = XGBClassifier(
        n_estimators=300,
        max_depth=9,
        learning_rate=0.05,
        scale_pos_weight=scale_pos_weight,
        subsample=0.9,
        colsample_bytree=0.8,
        gamma=0.1,
        tree_method='hist',
        # use_label_encoder=False,
        eval_metric='logloss',
        random_state=42
    )

    base_rf = RandomForestClassifier(
        n_estimators=200,
        max_depth=5,
        min_samples_split=5,
        min_samples_leaf=2,
        max_samples=0.5,
        max_features=0.65,
        bootstrap=True,
        class_weight='balanced',
        random_state=42,
    )

    ada_model = AdaBoostClassifier(
        estimator=base_rf,
        n_estimators=100,
        learning_rate=0.1,
        random_state=42
    )

    return rf_model,xgb_model,lgb_model,meta_model,ada_model
```

Method 1

- Using XGBoost with all dataset directly, `F1 score = 0.38`

```
model = XGBClassifier(
    n_estimators=301,
    max_depth=7,
    learning_rate=0.05,
    scale_pos_weight=scale_pos_weight,
    subsample=0.8,
    colsample_bytree=0.7,
    gamma=0.1,
    tree_method='hist',
    random_state=42
)
```

Method 2

- Using neural network, with 2 stage training
  - first stage use all dataset without considering the imbalance and outliers
  - second stage use all dataset but considering the imbalance of class 0 and class 1
  - F1 score = 0.46

Method 3

- Using Adaboost with considering outliers
  - use random forest as base
- F1 score = 0.46

Final method

- Using Staking Classifier and considering outliers
- F1 score = 0.479

```
stacked_boost = StackingClassifier(
    estimators=[
        ('xgb', xgb_model),
        ('lgb', lgb_model),
        ('rf', rf_model)
    ],
    final_estimator=ada_model,
    passthrough=True
)
```

Outliers detection

```
numeric_cols = df_train.select_dtypes(include=[np.number]).columns
for col in numeric_cols.drop("target"):
    print(f"train : {df_train[col].describe()}\ntest : {df_test[col].describe()}\n")
```

Discover

```
train : count    156076.000000
mean         35.421710
std          58.834915
min          -3.220955
25%           -0.512021
50%            0.200537
75%          127.814373
max          146.519045
Name: 0, dtype: float64
test : count    194330.000000
mean           0.002194
std            1.005935
min           -3.748814
25%           -0.668087
50%           -0.221901
75%            0.427884
max           18.774325
Name: 0, dtype: float64
```

- I realise that some column will have very big different with test set, from above example Q3 = 127.81 at train set but Q3 is 0.42 only
- After trying some columns, I found that they all have something similar result
  - all dataset has **156076** data, remove "outliers" has **114514** data, "outliers" has **41562** data
- So I decide use column "0" and if this column has value greater than 100 then delete the whole row.

Imbalance class 0 and class 1

- After remove outliers, `class 0` still have **93028** data, but `class 1` only left **21486** data.
- In the model above we can use **scale_pos_weight** to duel with imbalance using
  - `scale_pos_weight = (len(y) - sum(y)) / sum(y)`
  - since class only 0 and 1, so sum(y) is the total number of class 1
  - then pass this as a parameter to the model

Separate continuous and structural

- I found that from columns 0 - 35 looks like a signal(continuous) data while columns 36 - 95 looks like structural data
- So I try to separate them and use different model to train and combine them but this not give a better result
- F1 score = 0.43
- This method **not** used in final result

References

[1] ChatGPT free version. (2025, 9 May of queries). "請給我使用stacking來做訓練的方式",”請問對於不平衡的資料怎麼使用neural network來做訓練”,”請幫我將資料columns 0-35 與其他 columns 分開”“對於以下模型給我一些不錯的參數，針對114514筆資料96個features的dataset，model:xgb,lgb,randomforest”,”怎麼使用PCA降緯並且畫出2D與3D的圖” Generated using OpenAI. https://chatgpt.com