from sklearn.model\_selection import cross\_validate

import pandas as pd

from sklearn.model\_selection import cross\_val\_predict, StratifiedKFold

from sklearn.metrics import precision\_recall\_curve, auc

from sklearn.model\_selection import GridSearchCV

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

import matplotlib.pyplot as plt

import numpy as np

data\_file\_path = r"C:\Users\21539\Downloads\ex2data1.txt"

data = pd.read\_csv(data\_file\_path)

X = data.iloc[:, : -1]

y = data.iloc[:, -1]

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

model = LogisticRegression()

cv = 5

scoring = ['accuracy', 'precision', 'recall', 'f1', 'roc\_auc']

cv\_results = cross\_validate(model, X\_scaled, y, cv = cv, scoring = scoring, return\_train\_score = True)

plt.figure(figsize = (10, 5))

plt.plot(cv\_results['train\_accuracy'], label = 'Train Accuracy')

plt.plot(cv\_results['test\_accuracy'], label = 'Test Accuracy')

plt.title('Training and Test Accuracy over Cross-Validation Folds')

plt.xlabel('Fold')

plt.ylabel('Accuracy')

plt.legend()

plt.show()

cv = StratifiedKFold(n\_splits = 5)

model = LogisticRegression()

y\_pred\_prob = cross\_val\_predict(model, X\_scaled, y, cv = cv, method = 'predict\_proba')

y\_pred\_prob = y\_pred\_prob[:, 1]

precisions, recalls, thresholds = precision\_recall\_curve(y, y\_pred\_prob)

auc\_pr = auc(recalls, precisions)

plt.figure(figsize = (8, 6))

plt.plot(recalls, precisions, color = 'b', label = f'AUC-PR = {auc\_pr:.2f}')

plt.xlabel('Recall')

plt.ylabel('Precision')

plt.ylim([0.0, 1.05])

plt.xlim([0.0, 1.0])

plt.title('Precision-Recall Curve')

plt.legend(loc = "lower left")

plt.show()

param\_grid = {

'C': np.logspace(-4, 4, 50),

}

model = LogisticRegression(solver = 'lbfgs', max\_iter = 10000)

grid\_search = GridSearchCV(model, param\_grid, cv = 5, scoring = 'f1', return\_train\_score = False)

grid\_search.fit(X\_scaled, y)

f1\_scores = grid\_search.cv\_results\_['mean\_test\_score']

param\_C\_values = grid\_search.cv\_results\_['param\_C'].data

plt.plot(param\_C\_values, f1\_scores, marker = 'o')

plt.title('F1 Score vs C (Logistic Regression)')

plt.xlabel('C')

plt.ylabel('F1 Score')

plt.xscale('log')

plt.show()

from sklearn.metrics import roc\_curve, auc

fpr, tpr, thresholds = roc\_curve(y, y\_pred\_prob)

auc\_roc = auc(fpr, tpr)

# 绘制ROC曲线

plt.figure(figsize = (10, 5))

plt.plot(fpr, tpr, color = 'darkorange', lw = 2, label = 'ROC curve (area = %0.2f)' % auc\_roc)

plt.plot([0, 1], [0, 1], color = 'navy', lw = 2, linestyle = '--')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Receiver Operating Characteristic')

plt.legend(loc = "lower right")

plt.show()

print(f'AUC-ROC: {auc\_roc:.2f}')