Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи №3 з дисципліни
«Програмування інтелектуальних інформаційних систем»

„**Методи RandomForest та XGBoost**”

Виконав(ла) *ІП-11 Прищепа В.С.* _____
(шифр, прізвище, ім'я, по батькові)

Перевірив *Баришич Л. М.* _____
(прізвище, ім'я, по батькові)

Київ 2023

Завдання

1. Пройти тутор:

https://www.kaggle.com/code/jhoward/linear-model-and-neural-net-from-scratch#Deep-learning

2. Побудувати рендом форест звідси:

https://www.kaggle.com/code/jhoward/how-random-forests-really-work/

2.1. Натрейнити на датасеті звідси**:**

'/kaggle/input/car-evaluation-data-set/car_evaluation.csv'

Class - залежна змінна

Важливо! Незабудьте енкодер

encoder = ce.OrdinalEncoder(cols=['buying', 'maint', 'doors', 'persons', 'lug_boot', 'safety'])

2.2 Вивести **confusion matrix, auc, Classification report**

3 Зробити буст попередньої моделі XGBoost. Порівняти результати
https://machinelearningmastery.com/random-forest-ensembles-with-xgboost/

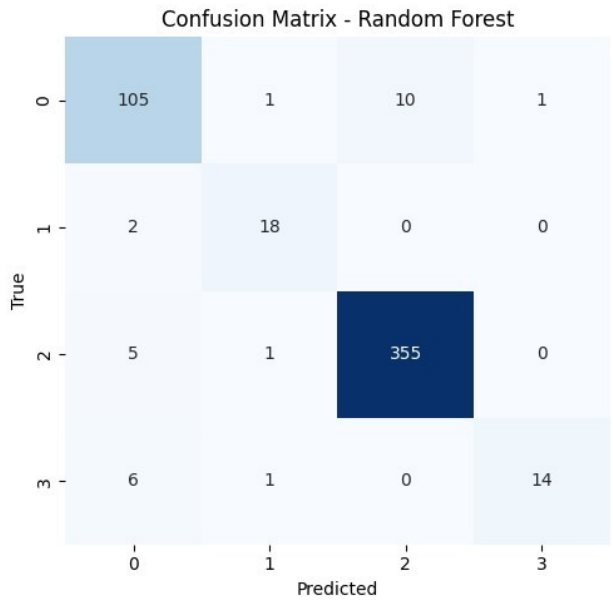
Хід роботи:

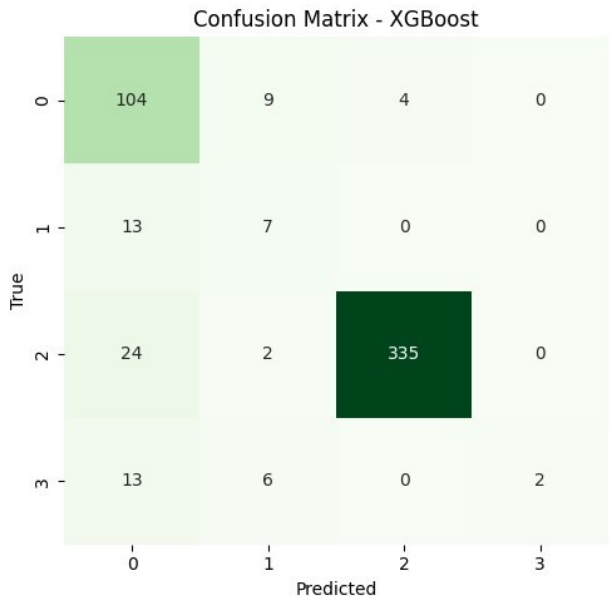
Код програми:

```
import pandas as pd
from sklearn.model_selection import train_test_split
import category_encoders as ce
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder, label_binarize
from xgboost import XGBRFClassifier
from sklearn.metrics import confusion_matrix, roc_auc_score, classification_report
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv("car_evaluation.csv", header=None)
col_names = ['buying', 'maint', 'doors', 'persons', 'lug_boot', 'safety', 'class']
df.columns = col_names
X = df.drop(['class'], axis=1)
Y = df['class']

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3,
random_state=20)
encoder = ce.OrdinalEncoder(cols=['buying', 'maint', 'doors', 'persons', 'lug_boot',
'safety'])
X_train = encoder.fit_transform(X_train)
```

```
X_test = encoder.transform(X_test)

rfc = RandomForestClassifier(random_state=10)
rfc.fit(X_train, y_train)
y_pred = rfc.predict(X_test)
label_encoder = LabelEncoder()
xgb = XGBRFClassifier(n_estimators=1000, random_state=1)
xgb.fit(X_train, label_encoder.fit_transform(y_train))
y_pred_xgb = label_encoder.inverse_transform(xgb.predict(X_test))

plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues',
cbar=False)
plt.title('Confusion Matrix - Random Forest')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.subplot(1, 2, 2)
sns.heatmap(confusion_matrix(y_test, y_pred_xgb), annot=True, fmt='d',
cmap='Greens', cbar=False)
plt.title('Confusion Matrix - XGBoost')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.tight_layout()
plt.show()

class_labels = ['unacc', 'acc', 'good', 'vgood']
y_test_encoded = label_binarize(y_test, classes=class_labels)
y_pred_encoded = label_binarize(y_pred, classes=class_labels)
y_pred_boosted_encoded = label_binarize(y_pred_xgb, classes=class_labels)
roc_auc_scores = [roc_auc_score(y_test_encoded[:, i], y_pred_encoded[:, i]) for i in
range(len(class_labels))]
roc_auc_scores_boosted = [roc_auc_score(y_test_encoded[:, i],
y_pred_boosted_encoded[:, i]) for i in range(len(class_labels))]
print('\nROC AUC scores:\n', pd.DataFrame({'Random Forest':roc_auc_scores,
'XGBoost':roc_auc_scores_boosted}).set_index(pd.Index(class_labels)), sep='')
print('\n\nClassification report (Random Forest):\n', classification_report(y_test,
y_pred))
print('\nClassification report (XGBoost):\n', classification_report(y_test,
y_pred_xgb))
```

Результат:






```
ROC AUC scores:
       Random Forest   XGBoost
unacc       0.960044  0.951331
acc         0.932549  0.882255
good        0.946994  0.657966
vgood       0.832329  0.547619


Classification report (Random Forest):
              precision    recall  f1-score   support

         acc       0.89      0.90      0.89       117
        good       0.86      0.90      0.88        20
       unacc       0.97      0.98      0.98       361
       vgood       0.93      0.67      0.78        21

    accuracy                           0.95       519
   macro avg       0.91      0.86      0.88       519
weighted avg       0.95      0.95      0.95       519


Classification report (XGBoost):
              precision    recall  f1-score   support

         acc       0.68      0.89      0.77       117
        good       0.29      0.35      0.32        20
       unacc       0.99      0.93      0.96       361
       vgood       1.00      0.10      0.17        21

    accuracy                           0.86       519
   macro avg       0.74      0.57      0.55       519
weighted avg       0.89      0.86      0.86       519
```

Висновок:

Отже, під час виконання лабораторної роботи я випробовував моделі Random Forest та дану модель, покращену за допомогою XGBoost. Я дослідив різні метрики і виявив, що буст дав результати, в загальному гірші за непокращену модель. Отже, застосування даного покращення не завжди є доцільним.