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

Звіт № 3 з
дисципліни
«Програмування інтелектуальних
інформаційних систем»

Виконав студент ІП-13 Романюк Діана Олексіївна
(шифр, прізвище, ім'я, по батькові)

Перевірив Баришич Лука Маріянович
(прізвище, ім'я, по батькові)

Київ 2023

Лабораторна робота 3

Постановка задачі:

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

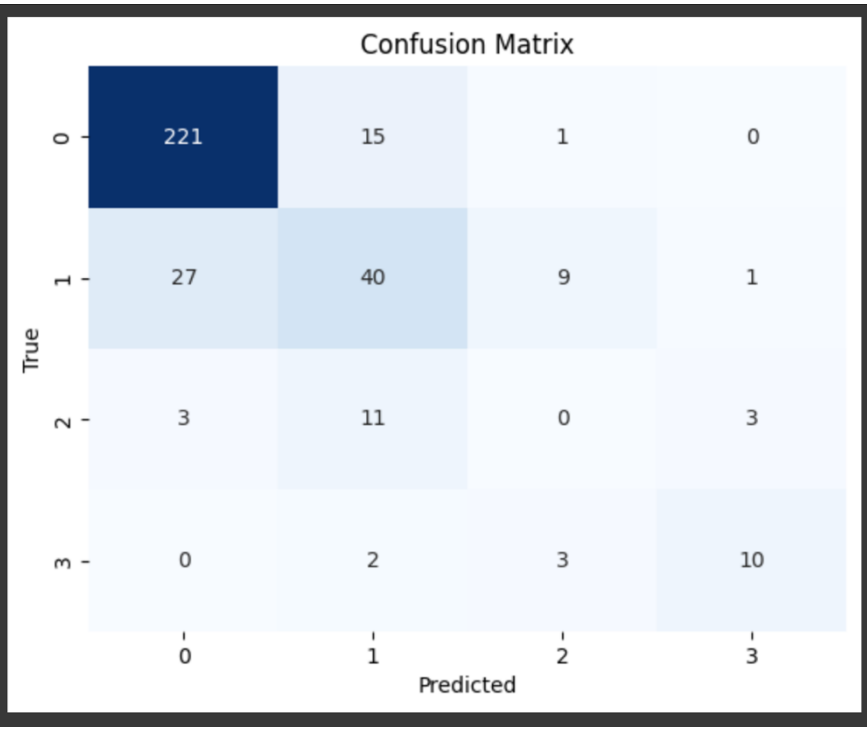
```
[ ] # Split the data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

[ ] # Encode categorical features
    x_encoder = ce.OrdinalEncoder(cols=X.columns)
    X_train = x_encoder.fit_transform(X_train)
    X_test = x_encoder.fit_transform(X_test)

[ ] y_encoder = ce.OrdinalEncoder(cols=['class'])
    y_train = y_encoder.fit_transform(y_train)
    y_test = y_encoder.fit_transform(y_test)

    # Train the Random Forest model
    rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
    rf_model.fit(X_train, y_train)

    <ipython-input-50-00375849f40b>:3: DataConversionWarning: A column-vector y was passed when a 1d array was expect
```

```
[ ] # Make predictions on the testing set
    y_pred = rf_model.predict(X_test)

[ ] # Evaluate the model
    accuracy = accuracy_score(y_test, y_pred)
    print(f'Accuracy: {accuracy:.2f}')

    Accuracy: 0.78
```

Confusion matrix

```
[ ] from sklearn.metrics import confusion_matrix, roc_auc_score, classification_report
    import seaborn as sns
    import matplotlib.pyplot as plt
```

```
# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()
```



AUC

```
[ ] # AUC Score
    auc_score = roc_auc_score(y_test, rf_model.predict_proba(X_test), multi_class='ovr')
    print(f'AUC Score: {auc_score}')
```

```
AUC Score: 0.8172047041695518
```

Classification Report

```
# Classification Report
class_report = classification_report(y_test, y_pred)
print('Classification Report:\n', class_report)
```

```
Classification Report:
              precision    recall  f1-score   support

           1       0.88      0.93      0.91       237
           2       0.59      0.52      0.55        77
           3       0.00      0.00      0.00        17
           4       0.71      0.67      0.69        15

    accuracy                           0.78       346
   macro avg       0.55      0.53      0.54       346
weighted avg       0.76      0.78      0.77       346
```

Task 3

```
[ ] from sklearn.preprocessing import LabelEncoder

    # Use LabelEncoder for target variable
    label_encoder = LabelEncoder()
    y_train = label_encoder.fit_transform(y_train)
    y_test = label_encoder.transform(y_test)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_label.py:116: DataConversionWarning: A column-vect
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_label.py:134: DataConversionWarning: A column-vect
  y = column_or_1d(y, dtype=self.classes_.dtype, warn=True)
```

```
import xgboost as xgb

from sklearn.preprocessing import LabelEncoder

clf_xgb = xgb.XGBClassifier( max_depth=4, random_state = 42, n_jobs = 4)

clf_xgb.fit(X_train, y_train)
```

```
▾                                XGBClassifier
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              multi_strategy=None, n_estimators=None, n_jobs=4,
              num_parallel_tree=None, objective='multi:softprob', ...)
```

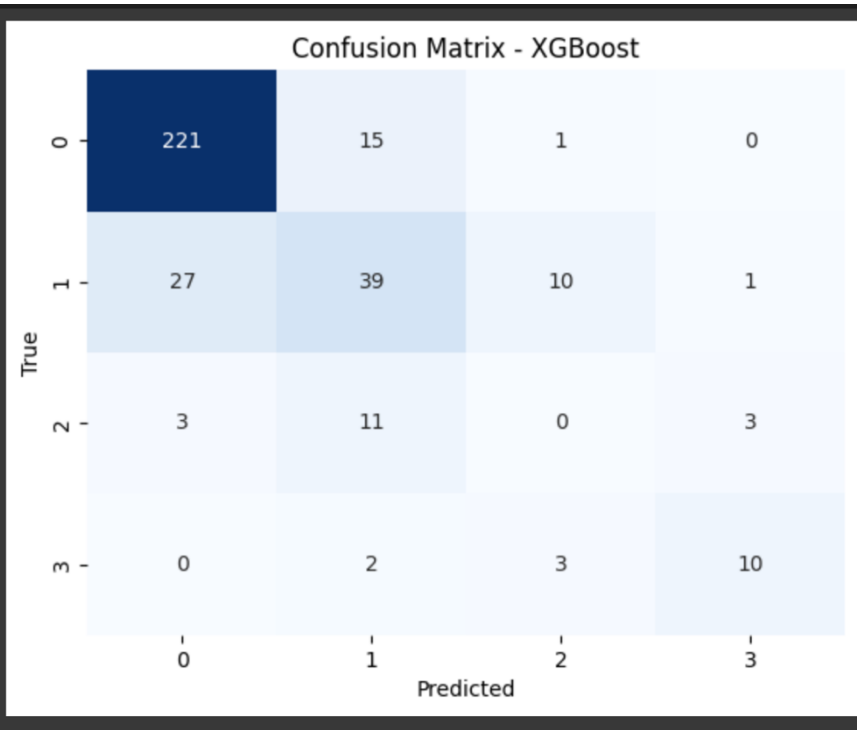
```
[ ] y_pred_xgb = clf_xgb.predict(X_test)
```

```
[ ] accuracy_xgb = accuracy_score(y_test, y_pred_xgb)
    print(f'Accuracy Score - XGBoost: {accuracy_xgb:.2f}')
```

```
Accuracy Score - XGBoost: 0.78
```

Connfusionn matrix

```
# Confusion Matrix
cm_xgb = confusion_matrix(y_test, y_pred_xgb)
sns.heatmap(cm_xgb, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix - XGBoost')
plt.show()
```



AUC

```
[ ] # AUC Score
    auc_score_xgb = roc_auc_score(y_test, clf_xgb.predict_proba(X_test), multi_class='ovr')
    print(f'AUC Score - XGBoost: {auc_score_xgb}')
```

```
AUC Score - XGBoost: 0.8796777855296329
```

Classification Report

```
# Classification Report
class_report_xgb = classification_report(y_test, y_pred_xgb)
print('Classification Report - XGBoost:\n', class_report_xgb)
```

```
Classification Report - XGBoost:
              precision    recall  f1-score   support

           0       0.88      0.93      0.91       237
           1       0.58      0.51      0.54        77
           2       0.00      0.00      0.00        17
           3       0.71      0.67      0.69        15

    accuracy                           0.78       346
   macro avg       0.54      0.53      0.53       346
weighted avg       0.76      0.78      0.77       346
```

Висновок:

У цій лабораторній роботі був побудований та натренований Random Forest класифікатор на наборі даних для оцінки автомобільних покупок. Датасет був енкодований за допомогою OrdinalEncoder з бібліотеки category_encoders для того, щоб підготувати дані для моделі. Після цього був використаний RandomForestClassifier з sklearn для тренування моделі та отримання результатів.

Confusion matrix, AUC Score та Classification Report були використані для оцінки ефективності моделі. Матриця плутанини вказує на точність прогнозування для кожного класу, AUC Score вимірює дискримінативну здатність моделі, а Classification Report надає деталізовану інформацію про точність, полноту та F1-меру для кожного класу.

Далі був використаний XGBoost, який є бустінговим методом машинного навчання, для порівняння результатів з Random Forest. Моделі показали схожі результати у визначенні точності, але за рахунок підвищення значень AUC Score, можна вважати XGBoost більш ефективним для даного набору даних.

У висновку, обидві моделі мають прийнятні показники ефективності, але XGBoost виявився трошки кращим за Random **Forest за показниками AUC Score, що вказує на його кращу здатність розрізняти класи.**