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

Лабораторна робота №3

Програмування ітелектуальних інформаційних систем

Виконав
студент групи ІП-11:
Панченко С. В.

Перевірила:
Баришич Л. М

Київ 2023

ЗМІСТ

1 МЕТА ЛАБОРАТОРНОЇ РОБОТИ

Класифікація даних.

2 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/

N.B.: catboost російський, тому того, хто спробує здати буст кетбустом, буде внесено в базу “Миротворець”

3 ВИКОНАННЯ

3.13вичайна модель

Для початку імпортуємо модулі. Завантажимо датафрейм та виведемо його вміст.

In [116…
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('data/car_evaluation.csv')
df
```

Out[116…

	buying	maint	doors	persons	lug_boot	safety	class
0	vhigh	vhigh	2	2	small	low	unacc
1	vhigh	vhigh	2	2	small	med	unacc
2	vhigh	vhigh	2	2	small	high	unacc
3	vhigh	vhigh	2	2	med	low	unacc
4	vhigh	vhigh	2	2	med	med	unacc
...	...	...	...	...	...	...	...
1723	low	low	5more	more	med	med	good
1724	low	low	5more	more	med	high	vgood
1725	low	low	5more	more	big	low	unacc
1726	low	low	5more	more	big	med	good
1727	low	low	5more	more	big	high	vgood

1728 rows × 7 columns

Рисунок 3.1.1 - Сутності

Розділимо дані на тестові та навчальні.

In [117…
```
x=df.drop(['class'],axis=1)
y=df['class']
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=1)
x_train.shape,x_test.shape
```

Out[117… ((1209, 6), (519, 6))

Рисунок 3.1.2 - Тестові та навчальні дані

Перетворимо іменовані значення у числові.

```
In [118…  import category_encoders as ce

          encoder = ce.OrdinalEncoder(cols=['buying', 'maint', 'doors', 'persons', 'lug_boot', 'safe
          x_train = encoder.fit_transform(x_train)
          x_test = encoder.transform(x_test)
```

```
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:28: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  elif pd.api.types.is_categorical_dtype(cols):
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:50: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  return pd.api.types.is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:50: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  return pd.api.types.is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:50: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  return pd.api.types.is_categorical_dtype(dtype)
```

Рисунок 3.1.3 - Перетворення іменованих значень у числові

Натренуємо модель.

```
In [119…  from sklearn.ensemble import RandomForestClassifier
          rfc=RandomForestClassifier(random_state=0)
          rfc.fit(x_train,y_train)

Out[119…  ▾        RandomForestClassifier
          RandomForestClassifier(random_state=0)
```

Рисунок 3.1.4 - Тренування моделі

Спрогнозуємо значення

```
In [120…  y_pred=rfc.predict(x_test)
```

Рисунок 3.1.5 - Прогнозування значень

Виведемо класифікаційний звіт.

```
In [121…  from sklearn.metrics import classification_report
          common_report = classification_report(y_test, y_pred)
          print(common_report)

              precision    recall  f1-score   support

         acc       0.70      0.77      0.73       111
        good       0.40      0.18      0.25        22
       unacc       0.94      0.97      0.96       368
       vgood       0.88      0.39      0.54        18

    accuracy                           0.87       519
   macro avg       0.73      0.58      0.62       519
weighted avg       0.87      0.87      0.86       519
```

Рисунок 3.1.6 - класифікайційний звіт

Побудуємо матрицю невідповідностей.

```
In [122…
from sklearn.metrics import confusion_matrix
def conf_mat(model, x_test, y_test):
    y_predicted = model.predict(x_test)
    cm = confusion_matrix(y_test, y_predicted)
    plt.figure(figsize = (8,5))
    sns.heatmap(cm, annot=True, fmt=".1f")
    plt.xlabel('Predicted')
conf_mat(rfc, x_test, y_test)
```

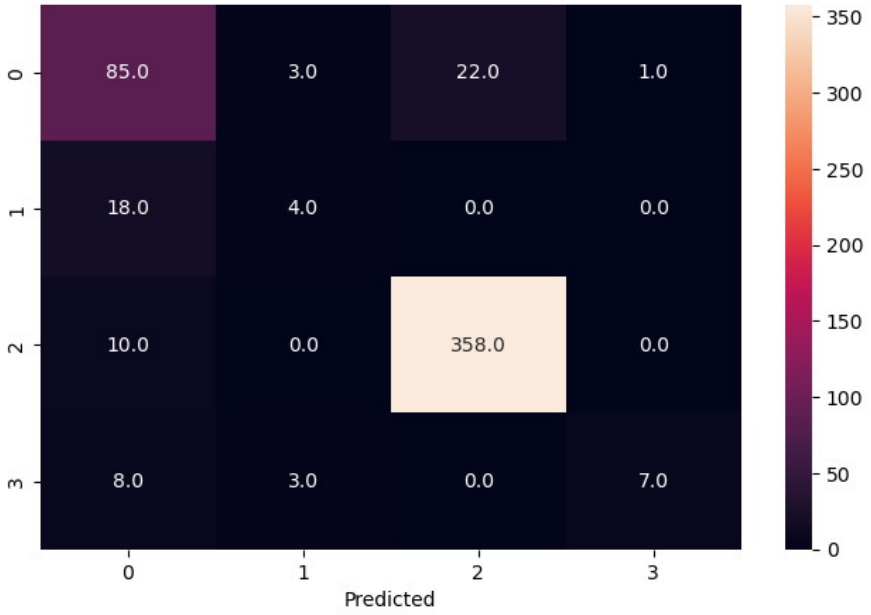

Рисунок 3.1.7 - Матриця невідповідностей

Для побудови ROC застосуємо принцип "один проти всіх".

Виведемо можливі значення класів.

```
In [123…
target = y.unique()
target
```

```
Out[123…
array(['unacc', 'acc', 'vgood', 'good'], dtype=object)
```

Рисунок 3.1.8 - Значення класів

Бінаризуємо значення.

```
In [124…
from sklearn.preprocessing import label_binarize
binarized = label_binarize(y, classes=target)
binarized[:3]
```

```
Out[124…
array([[1, 0, 0, 0],
       [1, 0, 0, 0],
       [1, 0, 0, 0]])
```

Рисунок 3.1.9 - Бінаризація

Натренуємо модель.

In [125…

```
from sklearn.multiclass import OneVsRestClassifier
from sklearn.metrics import roc_curve, auc

ovr_train_X, ovr_test_X, ovr_train_y, ovr_test_y = train_test_split(x,
    binarized, test_size=0.25, random_state=42)

encoder = ce.OrdinalEncoder(cols=['buying', 'maint', 'doors', 'persons', 'lug_boot', 'safe
ovr_train_X = encoder.fit_transform(ovr_train_X)
ovr_test_X = encoder.transform(ovr_test_X)

model = OneVsRestClassifier(RandomForestClassifier(random_state=0))\
    .fit(ovr_train_X, ovr_train_y)
```

```
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:28: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  elif pd.api.types.is_categorical_dtype(cols):
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:50: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  return pd.api.types.is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:50: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  return pd.api.types.is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:50: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  return pd.api.types.is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:50: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  return pd.api.types.is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:50: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  return pd.api.types.is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:50: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  return pd.api.types.is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:50: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  return pd.api.types.is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:50: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  return pd.api.types.is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:50: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  return pd.api.types.is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:50: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  return pd.api.types.is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:50: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  return pd.api.types.is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:50: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  return pd.api.types.is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:50: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  return pd.api.types.is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:50: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  return pd.api.types.is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:50: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  return pd.api.types.is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/category_encoders/utils.py:50: Fut
ureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use
```

Рисунок 3.1.10 - Тренування моделі

Визначення auc.

```
In [126…
prob_test_vec = model.predict_proba(ovr_test_X)

n_classes = 4
fpr = [0] * n_classes
tpr = [0] * n_classes
thresholds = [0] * n_classes
auc_score = [0] * n_classes

for i in range(n_classes):
    fpr[i], tpr[i], thresholds[i] = roc_curve(ovr_test_y[:, i],
                                              prob_test_vec[:, i])
    auc_score[i] = auc(fpr[i], tpr[i])

auc_score

Out[126…
[0.9914596273291926,
 0.9250745123498687,
 0.9801577669902912,
 0.9609112709832135]
```

Рисунок 3.1.11 - AUC Score

Виведемо усереднений AUC Score.

```
In [127…
sum(auc_score) / n_classes

Out[127…
0.9644007944131415
```

Рисунок 3.1.12 - Усереднений AUC Score

Побудуємо графік ROC.

In [128…

```
from sklearn.metrics import RocCurveDisplay
fig, ax = plt.subplots(figsize=(10, 10))
colors = ["aqua", "darkorange", "cornflowerblue", "red"]
for class_id, color in zip(range(n_classes), colors):
    RocCurveDisplay.from_predictions(
        ovr_test_y[:, class_id],
        prob_test_vec[:, class_id],
        name=f"ROC curve for {target[class_id]}",
        color=color,
        ax=ax,
    )
```

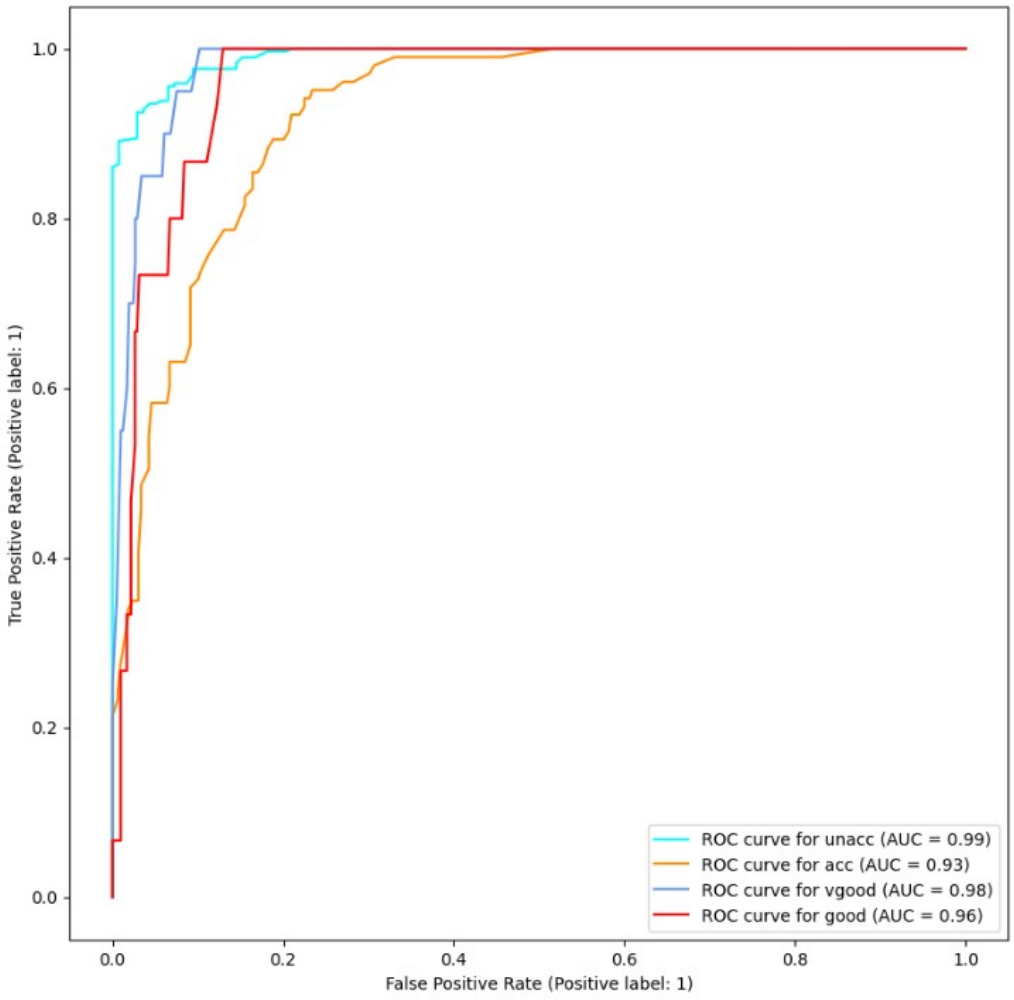

Рисунок 3.1.13 - Графік ROC

3.2 XGBoost

Натренуємо модель, знайдемо найкращі параметри.

In [129…

```
import xgboost
from sklearn.model_selection import GridSearchCV

n_estimators = [int(x) for x in np.linspace(start = 10, stop = 300, num = 60)]
params = {'n_estimators': n_estimators}
xg = xgboost.XGBRFClassifier()
xg_model = GridSearchCV(xg, param_grid=params, cv=3, n_jobs=5)
xg_model.fit(ovr_train_X, ovr_train_y)
```

```
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:335: FutureWarnin
g: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtyp
e, pd.SparseDtype)` instead.
  if is_sparse(dtype):
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:338: FutureWarnin
g: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstan
ce(dtype, CategoricalDtype) instead
  is_categorical_dtype(dtype) or is_pa_ext_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:384: FutureWarnin
g: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstan
ce(dtype, CategoricalDtype) instead
  if is_categorical_dtype(dtype):
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:359: FutureWarnin
g: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstan
ce(dtype, CategoricalDtype) instead
  return is_int or is_bool or is_float or is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:335: FutureWarnin
g: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtyp
e, pd.SparseDtype)` instead.
  if is_sparse(dtype):
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:338: FutureWarnin
g: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstan
ce(dtype, CategoricalDtype) instead
  is_categorical_dtype(dtype) or is_pa_ext_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:384: FutureWarnin
g: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstan
ce(dtype, CategoricalDtype) instead
  if is_categorical_dtype(dtype):
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:359: FutureWarnin
g: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstan
ce(dtype, CategoricalDtype) instead
  return is_int or is_bool or is_float or is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:335: FutureWarnin
g: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtyp
e, pd.SparseDtype)` instead.
  if is_sparse(dtype):
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:338: FutureWarnin
g: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstan
ce(dtype, CategoricalDtype) instead
  is_categorical_dtype(dtype) or is_pa_ext_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:384: FutureWarnin
g: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstan
ce(dtype, CategoricalDtype) instead
  if is_categorical_dtype(dtype):
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:359: FutureWarnin
g: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstan
ce(dtype, CategoricalDtype) instead
  return is_int or is_bool or is_float or is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:335: FutureWarnin
g: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtyp
e, pd.SparseDtype)` instead.
  if is_sparse(dtype):
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:338: FutureWarnin
g: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstan
ce(dtype, CategoricalDtype) instead
  is_categorical_dtype(dtype) or is_pa_ext_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:384: FutureWarnin
g: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstan
ce(dtype, CategoricalDtype) instead
  if is_categorical_dtype(dtype):
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:359: FutureWarnin
g: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstan
ce(dtype, CategoricalDtype) instead
  return is_int or is_bool or is_float or is_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:335: FutureWarnin
g: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtyp
e, pd.SparseDtype)` instead.
  if is_sparse(dtype):
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:338: FutureWarnin
g: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstan
ce(dtype, CategoricalDtype) instead
  is_categorical_dtype(dtype) or is_pa_ext_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:384: FutureWarnin
g: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstan
ce(dtype, CategoricalDtype) instead
```

Рисунок 3.2.1 - Тренування моделі XGBoost

Спрогнозуємо значення.

```
In [130… xg_pred = xg_model.predict(ovr_test_X)
```

```
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:335: FutureWarnin
g: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtyp
e, pd.SparseDtype)` instead.
  if is_sparse(dtype):
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:338: FutureWarnin
g: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstan
ce(dtype, CategoricalDtype) instead
  is_categorical_dtype(dtype) or is_pa_ext_categorical_dtype(dtype)
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:384: FutureWarnin
g: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstan
ce(dtype, CategoricalDtype) instead
  if is_categorical_dtype(dtype):
/home/sideshowbobgot/.local/lib/python3.10/site-packages/xgboost/data.py:359: FutureWarnin
g: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstan
ce(dtype, CategoricalDtype) instead
  return is_int or is_bool or is_float or is_categorical_dtype(dtype)
```

Рисунок 3.2.2 - Прогнозовані значення

Виведемо класифікаційний звіт XGBoost.

```
In [134… xg_boost_report = classification_report(ovr_test_y, xg_pred)
print(target)
print(xg_boost_report)
```

```
['unacc' 'acc' 'vgood' 'good']
              precision    recall  f1-score   support

           0       0.93      0.96      0.95       294
           1       0.65      0.64      0.65       103
           2       1.00      0.05      0.10        20
           3       0.00      0.00      0.00        15

   micro avg       0.86      0.81      0.84       432
   macro avg       0.65      0.41      0.42       432
weighted avg       0.84      0.81      0.80       432
 samples avg       0.81      0.81      0.81       432
```

```
/home/sideshowbobgot/.local/lib/python3.10/site-packages/sklearn/metrics/_classification.p
y:1469: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0
in labels with no predicted samples. Use `zero_division` parameter to control this behavio
r.
  _warn_prf(average, modifier, msg_start, len(result))
/home/sideshowbobgot/.local/lib/python3.10/site-packages/sklearn/metrics/_classification.p
y:1469: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0
in samples with no predicted labels. Use `zero_division` parameter to control this behavio
r.
  _warn_prf(average, modifier, msg_start, len(result))
```

Рисунок 3.2.3 - Класифікаційний звіт XGBoost

Виведемо перший класифікаційний звіт.

```
In [132… print(common_report)
```

```
              precision    recall  f1-score   support

         acc       0.70      0.77      0.73       111
        good       0.40      0.18      0.25        22
       unacc       0.94      0.97      0.96       368
       vgood       0.88      0.39      0.54        18

    accuracy                           0.87       519
   macro avg       0.73      0.58      0.62       519
weighted avg       0.87      0.87      0.86       519
```

Рисунок 3.2.4 - Перший звіт

Результати(перше число звичайний, другий - XGBoost): acc { 0.7 : 0.65 },

good { 0.4 : 0 }, unacc { 0.94 : 0.93 }, vgood { 0.88 : 1 }.

ДОДАТОК А ТЕКСТИ ПРОГРАМНОГО КОДУ

Тексти програмного коду

(Найменування програми (документа))

Жорсткий диск

(Вид носія даних)

(Обсяг програми (документа), арк.)

Студента групи ІП-113 курсу

Панченка С. В

```
      import numpy as np
      import pandas as pd
      import matplotlib.pyplot as plt
      import seaborn as sns
      df = pd.read_csv('data/car_evaluation.csv')
      df
      x=df.drop(['class'],axis=1)
      y=df['class']
      from sklearn.model_selection import train_test_split
      x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_st
ate=1)
      x_train.shape,x_test.shape
      import category_encoders as ce
      encoder = ce.OrdinalEncoder(cols=['buying', 'maint', 'doors', 'persons',
'lug_boot', 'safety'])
      x_train = encoder.fit_transform(x_train)
      x_test = encoder.transform(x_test)
      from sklearn.ensemble import RandomForestClassifier
      rfc=RandomForestClassifier(random_state=0)
      rfc.fit(x_train,y_train)
      y_pred=rfc.predict(x_test)
      from sklearn.metrics import classification_report
      common_report = classification_report(y_test, y_pred)
      print(common_report)
      from sklearn.metrics import confusion_matrix
      def conf_mat(model, x_test, y_test):
      y_predicted = model.predict(x_test)
      cm = confusion_matrix(y_test, y_predicted)
      plt.figure(figsize = (8,5))
      sns.heatmap(cm, annot=True, fmt=".1f")
      plt.xlabel('Predicted')
      conf_mat(rfc, x_test, y_test)
      target = y.unique()
      target
      from sklearn.preprocessing import label_binarize
      binarized = label_binarize(y, classes=target)
      binarized[:3]
      from sklearn.multiclass import OneVsRestClassifier
      from sklearn.metrics import roc_curve, auc
      ovr_train_X, ovr_test_X, ovr_train_y, ovr_test_y = train_test_split(x,
      binarized, test_size=0.25, random_state=42)
```

```
      encoder = ce.OrdinalEncoder(cols=['buying', 'maint', 'doors', 'persons',
'lug_boot', 'safety'])
      ovr_train_X = encoder.fit_transform(ovr_train_X)
      ovr_test_X = encoder.transform(ovr_test_X)
      model = OneVsRestClassifier(RandomForestClassifier(random_state=0))\
      .fit(ovr_train_X, ovr_train_y)
      prob_test_vec = model.predict_proba(ovr_test_X)
      n_classes = 4
      fpr = [0] * n_classes
      tpr = [0] * n_classes
      thresholds = [0] * n_classes
      auc_score = [0] * n_classes
      for i in range(n_classes):
      fpr[i], tpr[i], thresholds[i] = roc_curve(ovr_test_y[:, i],
      prob_test_vec[:, i])
      auc_score[i] = auc(fpr[i], tpr[i])
      auc_score
      sum(auc_score) / n_classes
      from sklearn.metrics import RocCurveDisplay
      fig, ax = plt.subplots(figsize=(10, 10))
      colors = ["aqua", "darkorange", "cornflowerblue", "red"]
      for class_id, color in zip(range(n_classes), colors):
      RocCurveDisplay.from_predictions(
      ovr_test_y[:, class_id],
      prob_test_vec[:, class_id],
      name=f"ROC curve for {target[class_id]}",
      color=color,
      ax=ax,
      )
      import xgboost
      from sklearn.model_selection import GridSearchCV
      n_estimators = [int(x) for x in np.linspace(start = 10, stop = 300, num =
60)]
      params = {'n_estimators': n_estimators}
      xg = xgboost.XGBRFClassifier()
      xg_model = GridSearchCV(xg, param_grid=params, cv=3, n_jobs=5)
      xg_model.fit(ovr_train_X, ovr_train_y)
      xg_pred = xg_model.predict(ovr_test_X)
      xg_boost_report = classification_report(ovr_test_y, xg_pred)
      print(target)
      print(xg_boost_report)
      print(common_report)
```