Практическая работа №1

Выполнил студент: Хрусталев А.А. Группа R3235

Преподаватели: Каканов М.А. Астафьев О.А.

Цель и задачи лабораторной работы

Цель: Необходимо по данным с мобильных сенсоров при помощи прикладных алгоритмов машинного обучения предсказать активность человека по шести классам движений:

- Движется по прямой
- Движется вверх (например, движение по лестнице вверх)
- Движется вниз (например, движение по лестнице вниз)
- Сидит
- Стоит
- Лежит

Задачи:

- Протестировать различные модели классификации
- Выведите отчет о классификации, сравнив предсказания (yhat) с базовой истиной (test_y), на основе которого найти самую эффективную модель
- Найти отличия между показателями precision и recall
- Дать определение показателю F1

Ход работы

DATALORE

```
# Импорт библиотек
import os
import numpy as np
import pandas as pd

# Считывание данных
def read_data(path, filename):
    return pd.read_csv(os.path.join(path, filename))

# df - массив данных
df = read_data('/data/notebook_files', 'train.csv')
df.head()

# Загружаем полный набор данных
def load_dataset(label_dict):
    train_X = read_data('/data/notebook_files', 'train.csv').values[:,:-2]
    train_y = read_data('/data/notebook_files', 'train.csv')['Activity']
    train_y = train_y.map(label_dict).values
    test_X = read_data('/data/notebook_files', 'test.csv').values[:,:-2]
    test_y = read_data('/data/notebook_files', 'test.csv')
    test_y = test_y['Activity'].map(label_dict).values
    return(train_X, train_y, test_X, test_y)
label_dict = {'WALKING':0, 'WALKING_UPSTAIRS':1, 'WALKING_DOWNSTAIRS':2, 'SITTING':3,
'STANDING':4, 'LAYING':5}
```

```
train_X, train_y, test_X, test_y = load_dataset(label_dict)

# https://tproger.ru/translations/scikit-learn-in-python/
# Выбор модели обучения

from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC

LDA_model = LinearDiscriminantAnalysis()
KNN_model = KNeighborsClassifier()
GNB_model = GaussianNB()
DTC_model = DecisionTreeClassifier()
SVC_model = SVC()

# Обучение модели
LDA_model.fit(train_X, train_y)
KNN_model.fit(train_X, train_y)
GNB_model.fit(train_X, train_y)
DTC_model.fit(train_X, train_y)
SVC_model.fit(train_X, train_y)

# Прогнозирование модели
LDA_prediction = LDA_model.predict(test_X)
KNN_prediction = KNN_model.predict(test_X)
GNB_prediction = GNB_model.predict(test_X)
DTC_prediction = DTC_model.predict(test_X)
SVC_prediction = SVC_model.predict(test_X)

# Оценка модели
from sklearn.metrics import classification_report
target_names = ['Walking', 'Walking Upstairs', 'Walking Downstairs', 'Sitting',
'Standing', 'Laying']
print("\n\n===LDA===\n")
print(classification_report(test_y, LDA_prediction, target_names=target_names))
print("\n\n===KNN===\n")
print(classification_report(test_y, KNN_prediction, target_names=target_names))
print("\n\n===GNB===\n")
print(classification_report(test_y, GNB_prediction, target_names=target_names))
print("\n\n===DTC===\n")
print(classification_report(test_y, DTC_prediction, target_names=target_names))
print("\n\n===SVC===\n")
print(classification_report(test_y, SVC_prediction, target_names=target_names))
```

Результат вывода программы

```
       ===LDA===

                    precision    recall  f1-score   support
```

```
           Walking       0.98      0.99      0.98       496
  Walking Upstairs       0.96      0.98      0.97       471
Walking Downstairs       1.00      0.96      0.98       420
           Sitting       0.95      0.88      0.92       491
          Standing       0.90      0.96      0.93       532
            Laying       1.00      1.00      1.00       537

          accuracy                           0.96      2947
         macro avg       0.96      0.96      0.96      2947
      weighted avg       0.96      0.96      0.96      2947



===KNN===

                    precision    recall  f1-score   support

           Walking       0.85      0.98      0.91       496
  Walking Upstairs       0.89      0.90      0.90       471
Walking Downstairs       0.95      0.79      0.86       420
           Sitting       0.91      0.79      0.85       491
          Standing       0.83      0.93      0.88       532
            Laying       1.00      0.99      1.00       537

          accuracy                           0.90      2947
         macro avg       0.91      0.90      0.90      2947
      weighted avg       0.91      0.90      0.90      2947



===GNB===

                    precision    recall  f1-score   support

           Walking       0.82      0.84      0.83       496
  Walking Upstairs       0.76      0.96      0.84       471
Walking Downstairs       0.83      0.61      0.70       420
           Sitting       0.58      0.75      0.65       491
          Standing       0.80      0.86      0.83       532
            Laying       0.96      0.60      0.74       537

          accuracy                           0.77      2947
         macro avg       0.79      0.77      0.77      2947
      weighted avg       0.79      0.77      0.77      2947



===DTC===

                    precision    recall  f1-score   support
```

```
           Walking       0.82      0.89      0.85       496
  Walking Upstairs       0.82      0.76      0.79       471
Walking Downstairs       0.86      0.85      0.86       420
           Sitting       0.84      0.77      0.80       491
          Standing       0.80      0.86      0.83       532
            Laying       1.00      1.00      1.00       537

          accuracy                           0.86      2947
         macro avg       0.86      0.85      0.85      2947
      weighted avg       0.86      0.86      0.86      2947



===SVC===

                    precision    recall  f1-score   support

           Walking       0.94      0.98      0.96       496
  Walking Upstairs       0.93      0.96      0.94       471
Walking Downstairs       0.99      0.91      0.95       420
           Sitting       0.94      0.89      0.91       491
          Standing       0.91      0.95      0.93       532
            Laying       1.00      1.00      1.00       537

          accuracy                           0.95      2947
         macro avg       0.95      0.95      0.95      2947
      weighted avg       0.95      0.95      0.95      2947
```

По показателю "f1-score" видно, что LDA - наиболее эффективная модель для данной задачи.

Вывод

В ходе лабораторной работы обучены пять моделей классификации и выявлена самая эффективная (LDA), изучены параметры по которым можно оценить успешность модели (precision=TP/(TP+FP), recall=TP/(TP+FN), F1=w_avg(precision, recall)).