Московский государственный технический университет им. Н.Э. Баумана.

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Технологии машинного обучения»

Отчет по лабораторной работе №5

Выполнила:		Проверил:
студентка группы ИУ5-61		преподаватель каф. ИУ5
Абросимова Надежда		Гапанюк Ю.Е.
Подпись и дата:		Подпись и дата:

г. Москва, 2019 г.

Задание

1. Выберите набор данных (датасет) для решения задачи классификации или регресии.
2. В случае необходимости проведите удаление или заполнение пропусков и кодирование категориальных признаков.
3. С использованием метода train_test_split разделите выборку на обучающую и тестовую.
4. Обучите 1) одну из линейных моделей, 2) SVM и 3) дерево решений. Оцените качество моделей с помощью трех подходящих для задачи метрик. Сравните качество полученных моделей.
5. Произведите для каждой модели подбор одного гиперпараметра с использованием GridSearchCV и кросс-валидации.
6. Повторите пункт 4 для найденных оптимальных значений гиперпараметров. Сравните качество полученных моделей с качеством моделей, полученных в пункте 4.

Текст программы

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import SGDClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score, recall_score
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
%matplotlib inline
sns.set(style="ticks")

data=pd.read_csv('heart.csv', sep=",")

data.head()
```

	age	sex	cp	trestbps	chol	restecg	thalach	slope	ca	thal	target
0	63	1	3	145	233	0	150	0	0	1	1
1	37	1	2	130	250	1	187	0	0	2	1
2	41	0	1	130	204	0	172	2	0	2	1
3	56	1	1	120	236	1	178	2	0	2	1
4	57	0	0	120	354	1	163	2	0	2	1

Разделение выборки на обучающую и тестовую

```
X_train, X_test, y_train, y_test = train_test_split( data, data['target'],
test_size= 0.2, random_state= 1)
# Размер обучающей выборки
X_train.shape, y_train.shape
```

((242, 11), (242,))

```
# Размер тестовой выборки X_test.shape, y_test.shape
```

((61, 11), (61,))

```
#обучение линейной модели
sgd = SGDClassifier().fit(X_train, y_train)

target_sgd = sgd.predict(X_test)

accuracy_score(y_test, target_sgd), \
precision_score(y_test, target_sgd), \
 recall_score(y_test, target_sgd)
```
(0.6557377049180327, 0.65625, 0.6774193548387096)

```
#обучение SVC svc = SVC(gamma='auto').fit(X_train, y_train)
target_svc = svc.predict(X_test)

accuracy_score(y_test, target_svc), \
precision_score(y_test, target_svc), \
recall_score(y_test, target_svc)
```

(0.5081967213114754, 0.5081967213114754, 1.0)

```
#обучение дерева решений
tree = DecisionTreeClassifier(random_state=1, max_depth=0.75).fit(X_train,
y_train)
target_tree = tree.predict(X_test)
accuracy_score(y_test, target_tree), \
precision_score(y_test, target_tree),  \
recall_score(y_test, target_tree)
```
(0.5081967213114754, 0.5081967213114754, 1.0)

Подбор гиперпараметра

```
scores_sgd = cross_val_score(SGDClassifier(),
                             X_train, y_train, cv=2)
scores_svm_svc = cross_val_score(SVC(gamma='auto'),
                             X_train, y_train, cv=2)
scores_decision_tree = cross_val_score(DecisionTreeClassifier(), X_train,
y_train, cv=2)

parameters = {'alpha':[0.5,0.4,0.3,0.2,0.1]} clf_gs_sgd =
GridSearchCV(SGDClassifier(), parameters, cv=2, scoring='accuracy')
clf_gs_sgd.fit(X_train, y_train)
#для линейной модели
clf_gs_sgd.best_params_
Out[26]:
{'alpha': 0.1}
parameters = {'gamma':[0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0.1]}
clf_gs_svm_svc = GridSearchCV(SVC(), parameters, cv=2, scoring='accuracy')
clf_gs_svm_svc.fit(X_train, y_train)
#для SVC

clf_gs_svm_svc.best_params_
```
{'gamma': 0.9}

```
parameters = {'min_impurity_decrease':[0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0.1
]}
clf_gs_decision_tree = GridSearchCV(DecisionTreeClassifier(), parameters,
cv=2, scoring='accuracy')
clf_gs_decision_tree.fit(X_train, y_train)
#для дерева решений
clf_gs_decision_tree.best_params_
Out[30]:
{'min_impurity_decrease': 0.4}
```

Для найденных оптимальных значений:

```
sgd2 = SGDClassifier(alpha=0.1).fit(X_train, y_train)
target_sgd2= sgd2.predict(X_test)
accuracy_score(y_test, target_sgd2), \
precision_score(y_test, target_sgd2),  \
recall_score(y_test, target_sgd2)
```

```
(0.5573770491803278, 0.75, 0.1935483870967742)
```

```
svc2 = SVC(gamma=0.9).fit(X_train, y_train)
target_svc2 = svc2.predict(X_test)
accuracy_score(y_test, target_svc2), \
precision_score(y_test, target_svc2),  \
recall_score(y_test, target_svc2)
```

(0.5081967213114754, 0.5081967213114754, 1.0)

```
tree2 = DecisionTreeClassifier(random_state=1, min_impurity_decrease=0.4,
max_depth=0.75).fit(X_train, y_train)

target_tree2 = tree2.predict(X_test)
accuracy_score(y_test, target_tree2), \
precision_score(y_test, target_tree2),  \
recall_score(y_test, target_tree2)
```

(0.5081967213114754, 0.5081967213114754, 1.0)