Лабораторная работа № 5

Линейные модели, SVM и деревья решений

Цель лабораторной работы: изучение линейных моделей, SVM и деревьев решений.

Выполнил: Ханмурзин Тагир ИУ5-64

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.

In [23]:
```
import numpy as np
import pandas as pd
from scipy import stats
from sklearn.datasets import load_iris
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import *
from sklearn.metrics import *
from sklearn.svm import SVR
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor

%matplotlib inline

import warnings
```

In [24]:
```
warnings.filterwarnings('ignore') # Отключаем предупреждения
```

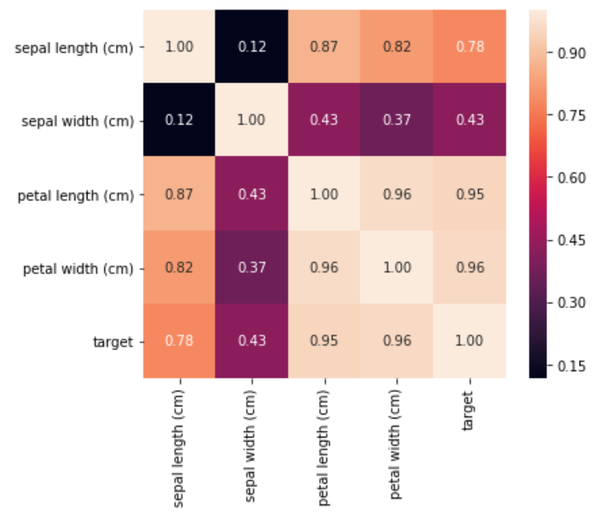
In [25]:
```
df = load_iris()
df = pd.DataFrame(data = np.c_[df['data'], df['target']], columns = df['feature_names'] + ['target'])
df.head()
```

Out[25]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0.0
1	4.9	3.0	1.4	0.2	0.0
2	4.7	3.2	1.3	0.2	0.0
3	4.6	3.1	1.5	0.2	0.0
4	5.0	3.6	1.4	0.2	0.0

In [26]:
```
fig, ax = plt.subplots(figsize=(6, 5))
sns.heatmap(df.corr(method='pearson').abs(), ax=ax, annot=True, fmt='.2f')
plt.yticks()
```

Out[26]: (array([0.5, 1.5, 2.5, 3.5, 4.5]), <a list of 5 Text yticklabel objects>)



In [27]:
```
df.loc[:, df.columns!='target'] = df.loc[:, df.columns!='target'].apply(lambda x: x/x.max(), axis=0)
x_train, x_test, y_train, y_test = train_test_split(df.loc[:, df.columns!='target'],
                                                    df['target'],
                                                    test_size= 0.33)
```

Линейная регрессия

In [28]:
```
def statistics(test_Y, target):
    print("Средняя абсолютная ошибка:", mean_absolute_error(test_Y, target))   # Средняя абсолютная ошибка
    print("Средняя квадратичная ошибка:", mean_squared_error(test_Y, target))    # Средняя квадратичная ошибка
    print("Медианная абсолютная ошибка:", median_absolute_error(test_Y, target)) # Медианная абсолютная ошибка
```

In [29]:
```
lr = LinearRegression().fit(x_train, y_train)

statistics(lr.predict(x_test), y_test)
```

```
Средняя абсолютная ошибка: 0.19393171749524554
Средняя квадратичная ошибка: 0.06217277784285951
Медианная абсолютная ошибка: 0.1681450948944981
```

SVM (метод опорных векторов)

In [30]:
```
svr = SVR().fit(x_train, y_train)

statistics(svr.predict(x_test), y_test)
```

```
Средняя абсолютная ошибка: 0.1914532804949536
Средняя квадратичная ошибка: 0.06671084414512415
Медианная абсолютная ошибка: 0.12088539780461649
```

Decision Tree

In [31]:
```
dt = DecisionTreeRegressor(max_depth=2).fit(x_train, y_train)

statistics(dt.predict(x_test), y_test)
```

```
Средняя абсолютная ошибка: 0.09052631578947368
Средняя квадратичная ошибка: 0.0765650969529086
Медианная абсолютная ошибка: 0.0
```

GridSearch

In [32]:
```
lr = GridSearchCV(LinearRegression(), {'n_jobs':range(1,10)}, cv=3).fit(x_train, y_train).best_estimator_

statistics(lr.predict(x_test), y_test)
```

```
Средняя абсолютная ошибка: 0.19393171749524554
Средняя квадратичная ошибка: 0.06217277784285951
Медианная абсолютная ошибка: 0.1681450948944981
```

In [33]:
```
svr = GridSearchCV(SVR(), {'degree':range(1,10)}, cv=3).fit(x_train, y_train).best_estimator_

statistics(svr.predict(x_test), y_test)
```

```
Средняя абсолютная ошибка: 0.1914532804949536
Средняя квадратичная ошибка: 0.06671084414512415
Медианная абсолютная ошибка: 0.12088539780461649
```

In [34]:
```
dt = GridSearchCV(DecisionTreeRegressor(), {'max_depth':range(1,10)}, cv=3).fit(x_train, y_train).best_estimator_

statistics(dt.predict(x_test), y_test)
```

```
Средняя абсолютная ошибка: 0.08
Средняя квадратичная ошибка: 0.08
Медианная абсолютная ошибка: 0.0
```