



**Министерство науки и высшего образования  
Российской Федерации Федеральное государственное  
бюджетное образовательное учреждение высшего  
образования «Московский государственный  
технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управления»**

**Курс «Технологии машинного обучения»**

**Отчёт по рубежному контролю №2**

**«Методы построения моделей машинного обучения»**

**Вариант №1**

**Выполнила:  
студентка группы ИУ5-62Б  
Андреева А.А.**

**Преподаватель:  
Гапанюк Ю. Е.**

**2023 г.**

**Задание.** Для заданного набора данных - iris постройте модели классификации или регрессии (в зависимости от конкретной задачи, рассматриваемой в наборе данных). Для построения моделей используйте методы Метод опорных векторов и случайный лес. Оцените качество моделей на основе подходящих метрик качества (не менее двух метрик).

## Выполнение работы

Загружаем датасет.

```
: from sklearn.svm import SVC
  from sklearn.model_selection import train_test_split
  from sklearn.metrics import accuracy_score, f1_score, confusion_matrix
  from sklearn.ensemble import RandomForestClassifier
  from sklearn.preprocessing import StandardScaler
```

```
: from sklearn.datasets import load_iris
  iris = load_iris()
  X = iris.data
  y = iris.target
```

Масштабируем его с помощью StandardScaler.

```
scaler = StandardScaler()
X = scaler.fit_transform(X)
```

Разделим его на обучающую и тестовую выборку.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

Создадим и обучим модели SVM и Random Forest.

```
: svc_model = SVC()
  svc_model.fit(X_train, y_train)

  rf_model = RandomForestClassifier(n_estimators=100)
  rf_model.fit(X_train, y_train)
```

Была произведена оценка производительности с помощью методов `accuracy_score`, `f1_score`, `confusion_matrix`.

`Confusion_matrix` - это таблица, которая показывает, насколько часто классификатор ошибается. Выводится матрица размером  $n \times n$ , где  $n$  - количество классов. В каждой ячейке  $(i, j)$  матрицы указывается количество примеров класса  $i$ , которые были помечены как класс  $j$ . Эта метрика позволяет проанализировать, какие типы ошибок допускает модель

`F1_score` - это гармоническое среднее между точностью и полнотой. Она используется для оценки результатов бинарной классификации, а также в многоклассовой классификации, когда интересует среднее значение показателя F1.

Ассурасу\_score показывает, какая доля из всех предсказаний была правильной.

```
def evaluate_model(model, X_test, y_test):
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred, average='weighted')
    matrix = confusion_matrix(y_test, y_pred)
    return accuracy, f1, matrix

svc_accuracy, svc_f1, svc_matrix = evaluate_model(svc_model, X_test, y_test)
rf_accuracy, rf_f1, rf_matrix = evaluate_model(rf_model, X_test, y_test)
```

## Результаты:

```
print("random forest model")
print("accuracy:", rf_accuracy)
print("f1 score:", rf_f1)
print("matrix:")
print(rf_matrix)
```

```
random forest model
accuracy: 0.9111111111111111
f1 score: 0.9111111111111111
matrix:
[[11  0  0]
 [ 0 15  1]
 [ 0  3 15]]
```

```
print("svc model")
print("accuracy:", svc_accuracy)
print("f1 score:", svc_f1)
print("matrix:")
print(svc_matrix)
```

```
svc model
accuracy: 0.9555555555555556
f1 score: 0.9555555555555556
matrix:
[[11  0  0]
 [ 0 16  0]
 [ 0  2 16]]
```

## Вывод:

Обе модели показали высокие результаты, но SVC модель показала более высокие значения ассурасу – 0.95 и f1 score – 0.95. Матрица ошибок также показала, что модель SVC имеет меньше ложноотрицательных и ложноположительных результатов, что свидетельствует о ее лучшей производительности по сравнению с моделью случайного леса.