

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

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

Курс «Технологии машинного обучения»  
Отчет по рубежному контролю №2  
«Методы построения моделей машинного обучения»  
Вариант №2

Выполнил:  
студент группы ИУ5-62Б  
Балабанов  
Алексей  
Олегович

Подпись: \_\_\_\_\_

Дата: \_\_\_\_\_

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Юрий  
Евгеньевич

Подпись: \_\_\_\_\_

Дата: \_\_\_\_\_

Москва, 2023 г.

# Выполнение работы

Загрузим выданный датасет вин используя команду `load_wine()`.

```
In 1 1 from sklearn.datasets import load_wine
      2 from sklearn.svm import SVC
      3 from sklearn.ensemble import RandomForestClassifier
      4 from sklearn.model_selection import train_test_split
      5 from sklearn.metrics import accuracy_score, f1_score, confusion_matrix
      6
      7 wine = load_wine()
      8 X = wine.data
      9 y = wine.target
      Executed in 3s, 9 May at 22:22:36
```

Масштабируем его с помощью `StandardScaler`.

```
In 2 1 from sklearn.preprocessing import StandardScaler
      2
      3 scaler = StandardScaler()
      4 X = scaler.fit_transform(X)
      Executed in 64ms, 9 May at 22:22:36
```

Разделим его на обучающую и тестовую выборку.

```
In 3 1 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
      Executed in 47ms, 9 May at 22:22:36
```

Создадим и обучим модели SVM и Random Forest.

```
In 4 1 svc_model = SVC()
      2 svc_model.fit(X_train, y_train)
      3
      4 rf_model = RandomForestClassifier(n_estimators=100)
      5 rf_model.fit(X_train, y_train)
      Executed in 327ms, 9 May at 22:22:37
```

Оценим производительность разными методами

```
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)

print("SVC Model Results:")
print("Accuracy: ", svc_accuracy)
print("F1 Score: ", svc_f1)
print("Matrix: ", svc_matrix)

print("Random Forest Model Results:")
print("Accuracy: ", rf_accuracy)
print("F1 Score: ", rf_f1)
print("Matrix: ", rf_matrix)
```

### Получившийся результат

```
SVC Model Results:  
Accuracy: 0.9444444444444444  
F1 Score: 0.9451735389235388  
Matrix: [[18 1 0]  
[ 0 18 0]  
[ 0 2 15]]  
Random Forest Model Results:  
Accuracy: 0.9814814814814815  
F1 Score: 0.9814543481210146  
Matrix: [[19 0 0]  
[ 1 17 0]]
```

Обе модели показали высокие результаты, но модель случайного леса показала более высокие значения точности (0.98) и F1-меры (0.98). Кроме того, матрица ошибок также показывает, что модель случайного леса имеет меньше ложноотрицательных и ложноположительных результатов, что свидетельствует о ее лучшей производительности в сравнении с моделью SVM.

