

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

Курс «Технологии машинного обучения»

Отчёт по лабораторной работе №4

Выполнил:
Каятский П. Е.
группа ИУ5-64Б

Проверил:
Гапанюк Ю.Е.

Дата: 07.04.25

Дата:

Подпись:

Подпись:

Москва, 2025 г.

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

Задание:

1. Выберите набор данных (датасет) для решения задачи классификации или регрессии.
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 следующие модели:
 - одну из линейных моделей (линейную или полиномиальную регрессию при решении задачи регрессии, логистическую регрессию при решении задачи классификации);
 - SVM;
 - дерево решений.
5. Оцените качество моделей с помощью двух подходящих для задачи метрик. Сравните качество полученных моделей.
6. Постройте график, показывающий важность признаков в дереве решений.
7. Визуализируйте дерево решений или выведите правила дерева решений в текстовом виде.

Ход выполнения:

Линейные модели, SVM и деревья решений

```
'data_module': 'sklearn.datasets.data'}
```

1. проверим наличие пропусков

```
[5] 1 X.isnull().sum()
```

Executed at 2025.04.03 12:23:45 in 24ms

```
worst radius      0
worst texture     0
worst perimeter   0
worst area        0
worst smoothness  0
worst compactness 0
worst concavity   0
worst concave points 0
worst symmetry    0
worst fractal dimension 0
dtype: int64
```

2. масштабирование признаков

```
[6] 1 scaler = StandardScaler()
```

```
2 X_scaled = scaler.fit_transform(X)
```

Executed at 2025.04.03 12:23:48 in 18ms

3. Разделение на обучающую и тестовую выборки

```
[7] 1 X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, random_state=42)
```

Executed at 2025.04.03 12:23:50 in 9ms

Обучение моделей

- логистическая регрессия
- SVM
- дерево решений

1. логистическая регрессия

```
[8] 1 log_reg = LogisticRegression(max_iter=1000)
```

```
2 log_reg.fit(X_train, y_train)
```

```
3 y_pred_log = log_reg.predict(X_test)
```

Executed at 2025.04.03 12:23:52 in 100ms

2. SVM

```
[9] 1 svm = SVC(kernel='linear')
    2 svm.fit(X_train, y_train)
    3 y_pred_svm = svm.predict(X_test)
```

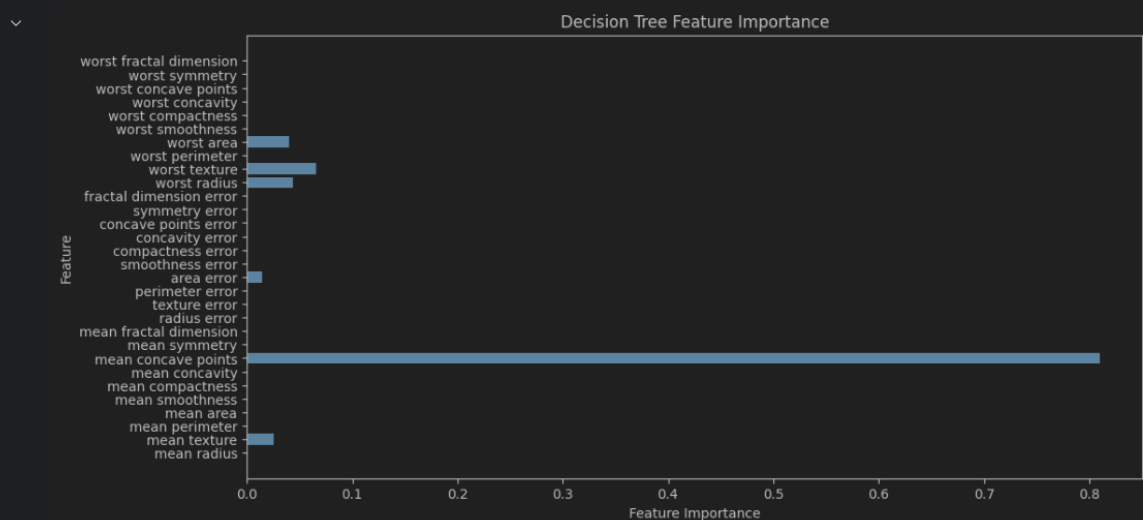
Executed at 2025.04.03 12:23:53 in 28ms

3. дерево решений

```
[10] 1 tree = DecisionTreeClassifier(max_depth=3, random_state=42)
    2 tree.fit(X_train, y_train)
    3 y_pred_tree = tree.predict(X_test)
```

Executed at 2025.04.03 12:23:55 in 46ms

```
[16] 1 plt.figure(figsize=(12, 6))
    2 plt.barh(data.feature_names, tree.feature_importances_)
    3 plt.xlabel("Feature Importance")
    4 plt.ylabel("Feature")
    5 plt.title("Decision Tree Feature Importance")
    6 plt.show()
```



Построение графика для визуализации важности признаков в д² 36 решений

```
[15] 1 from sklearn.tree import plot_tree
2
3 plt.figure(figsize=(20, 10))
4 plot_tree(tree, feature_names=data.feature_names, class_names=data.target_names, filled=True)
5 plt.show()
```

