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 «Системы обработки информации и управления»

Отчет по лабораторной работе №4

«Линейные модели, SVM и деревья решений»

по дисциплине «Технологии машинного обучения»

Выполнил:
студент группы ИУ5Ц-84Б
Тихонова Д.Д.
подпись, дата

Проверил:
к.т.н., доц., Ю.Е. Гапанюк
подпись, дата

2025 г.

1. Цель лабораторной работы

Изучение линейных моделей, SVM и деревьев решений.

2. Описание задания

Выберите набор данных (датасет) для решения задачи классификации или регрессии.

В случае необходимости проведите удаление или заполнение пропусков и кодирование категориальных признаков.

С использованием метода train_test_split разделите выборку на обучающую и тестовую.

Обучите следующие модели:

одну из линейных моделей (линейную или полиномиальную регрессию при решении задачи регрессии, логистическую регрессию при решении задачи классификации);

SVM;

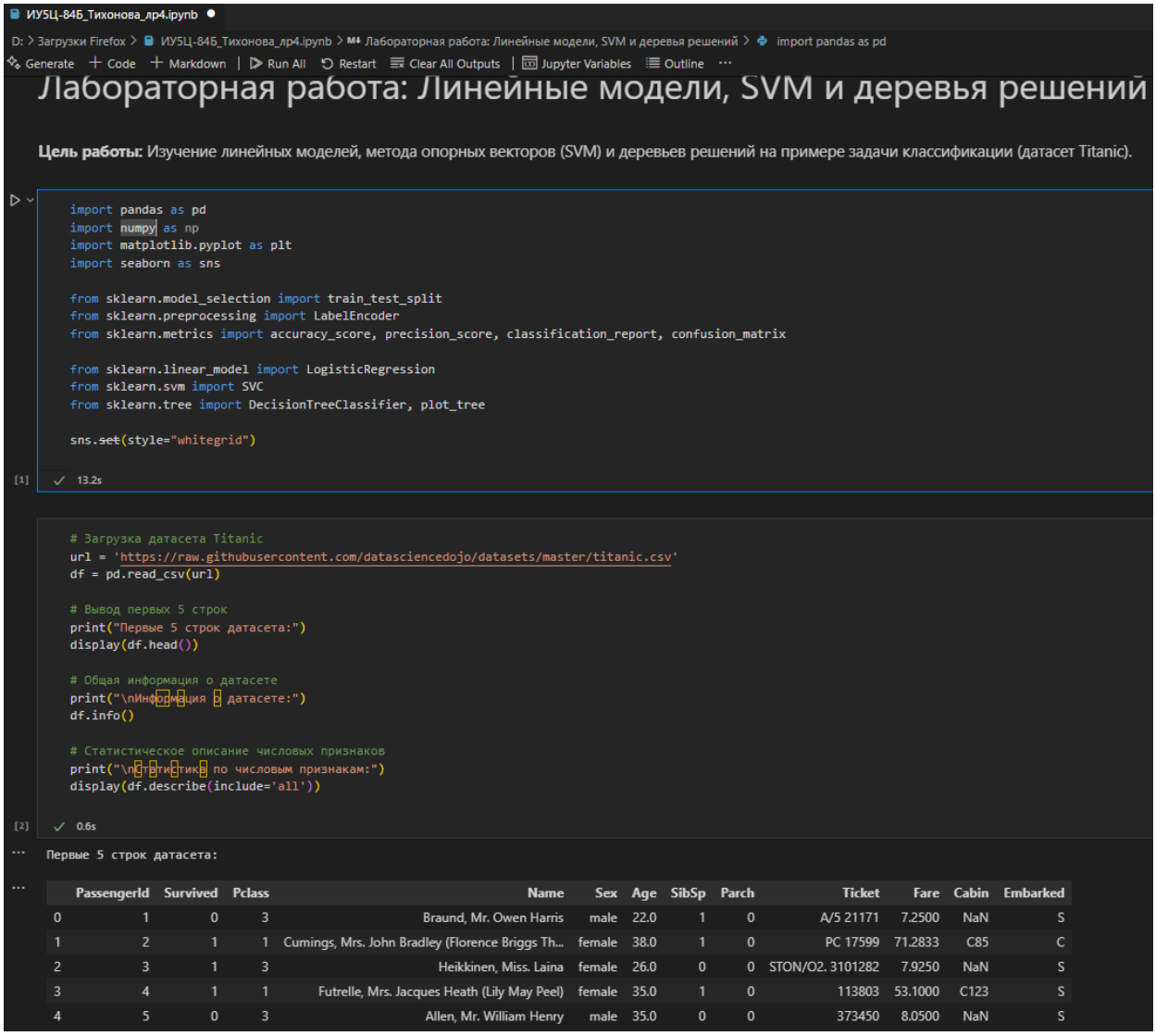
дерево решений.

Оцените качество моделей с помощью двух подходящих для задачи метрик. Сравните качество полученных моделей.

Постройте график, показывающий важность признаков в дереве решений.

Визуализируйте дерево решений или выведите правила дерева решений в текстовом виде.

3. Ход работы



	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

ИУ5Ц-84Б_Тихонова_лр4.ipynb

D: > Загрузки Firefox > ИУ5Ц-84Б_Тихонова_лр4.ipynb > Лабораторная работа: Линейные модели, SVM и деревья решений > import pandas as pd

Generate | Code | Markdown | Run All | Restart | Clear All Outputs | Jupyter Variables | Outline | ...

```
Информация о датасете:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB

Статистика по числовым признакам:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
count	891.000000	891.000000	891.000000	891	891	714.000000	891.000000	891.000000	891	891.000000	204	889
unique	NaN	NaN	NaN	891	2	NaN	NaN	NaN	681	NaN	147	3
top	NaN	NaN	NaN	Dooley, Mr. Patrick	male	NaN	NaN	NaN	347082	NaN	G6	S
freq	NaN	NaN	NaN	1	577	NaN	NaN	NaN	7	NaN	4	644
mean	446.000000	0.383838	2.308642	NaN	NaN	29.699118	0.523008	0.381594	NaN	32.204208	NaN	NaN
std	257.353842	0.486592	0.836071	NaN	NaN	14.526497	1.102743	0.806057	NaN	49.693429	NaN	NaN
min	1.000000	0.000000	1.000000	NaN	NaN	0.420000	0.000000	0.000000	NaN	0.000000	NaN	NaN
25%	223.500000	0.000000	2.000000	NaN	NaN	20.125000	0.000000	0.000000	NaN	7.910400	NaN	NaN
50%	446.000000	0.000000	3.000000	NaN	NaN	28.000000	0.000000	0.000000	NaN	14.454200	NaN	NaN
75%	668.500000	1.000000	3.000000	NaN	NaN	38.000000	1.000000	0.000000	NaN	31.000000	NaN	NaN
max	891.000000	1.000000	3.000000	NaN	NaN	80.000000	8.000000	6.000000	NaN	512.329200	NaN	NaN

```python
# Удалим ненужные признаки и обработаем пропуски
df = df[['Survived', 'Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare']]
df.dropna(inplace=True)

# Кодирование категориального признака
df['Sex'] = LabelEncoder().fit_transform(df['Sex'])

# Проверим после обработки
print("Данные после предобработки:")
display(df.head())
```

[3]

ИУ5Ц-84Б_Тихонова_лр4.ipynb

D: > Загрузки Firefox > ИУ5Ц-84Б_Тихонова_лр4.ipynb > Лабораторная работа: Линейные модели, SVM и деревья решений > import pandas as pd

Generate | + Code | + Markdown | Run All | Restart | Clear All Outputs | Jupyter Variables | Outline | ...

[3] ✓ 0.0s

```
Данные после предобработки:
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare
0	0	3	1	22.0	1	0	7.2500
1	1	1	0	38.0	1	0	71.2833
2	1	3	0	26.0	0	0	7.9250
3	1	1	0	35.0	1	0	53.1000
4	0	3	1	35.0	0	0	8.0500

```python
# Разделим данные на признаки и целевую переменную
X = df.drop('Survived', axis=1)
y = df['Survived']

# Разделим на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print(f"Размер обучающей выборки: {X_train.shape}")
print(f"Размер тестовой выборки: {X_test.shape}")
```

[4] ✓ 0.0s

```
Размер обучающей выборки: (571, 6)
Размер тестовой выборки: (143, 6)
```

```python
# Логистическая регрессия
log_reg = LogisticRegression(max_iter=1000)
log_reg.fit(X_train, y_train)
y_pred_log = log_reg.predict(X_test)

# SVM
svm_model = SVC(kernel='linear')
svm_model.fit(X_train, y_train)
y_pred_svm = svm_model.predict(X_test)

# Дерево решений
tree_model = DecisionTreeClassifier(max_depth=4, random_state=42)
tree_model.fit(X_train, y_train)
y_pred_tree = tree_model.predict(X_test)
```

[5] ✓ 6.5s

```python
def evaluate_model(name, y_true, y_pred):
    print(f"\n📊 {name}")
    print(f"Accuracy: {accuracy_score(y_true, y_pred):.2f}")
```

[6]

ИУ5Ц-84Б_Тихонова_лр4.ipynb

D: > Загрузки Firefox > ИУ5Ц-84Б_Тихонова_лр4.ipynb > Лабораторная работа: Линейные модели, SVM и деревья решений > import pandas as pd

Generate | Code | Markdown | Run All | Restart | Clear All Outputs | Jupyter Variables | Outline | ...

```python
def evaluate_model(name, y_true, y_pred):
    print(f"\n📊 {name}")
    print(f"Accuracy: {accuracy_score(y_true, y_pred):.2f}")
    print(f"Precision: {precision_score(y_true, y_pred):.2f}")
    print("Confusion Matrix:")
    print(confusion_matrix(y_true, y_pred))
    print("Classification Report:")
    print(classification_report(y_true, y_pred))

evaluate_model("Логистическая регрессия", y_test, y_pred_log)
evaluate_model("SVM", y_test, y_pred_svm)
evaluate_model("Дерево решений", y_test, y_pred_tree)
```

[6] ✓ 0.0s

```
📊 Логистическая регрессия
Accuracy: 0.75
Precision: 0.69
Confusion Matrix:
[[71 16]
 [20 36]]
Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.82      0.80        87
           1       0.69      0.64      0.67        56

    accuracy                           0.75       143
   macro avg       0.74      0.73      0.73       143
weighted avg       0.75      0.75      0.75       143


📊 SVM
Accuracy: 0.73
Precision: 0.67
Confusion Matrix:
[[69 18]
 [20 36]]
Classification Report:
...
    accuracy                           0.76       143
   macro avg       0.74      0.74      0.74       143
weighted avg       0.76      0.76      0.76       143
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```python
# Важность признаков
importances = tree_model.feature_importances_
features = X.columns
plt.figure(figsize=(10, 6))
```

[7]

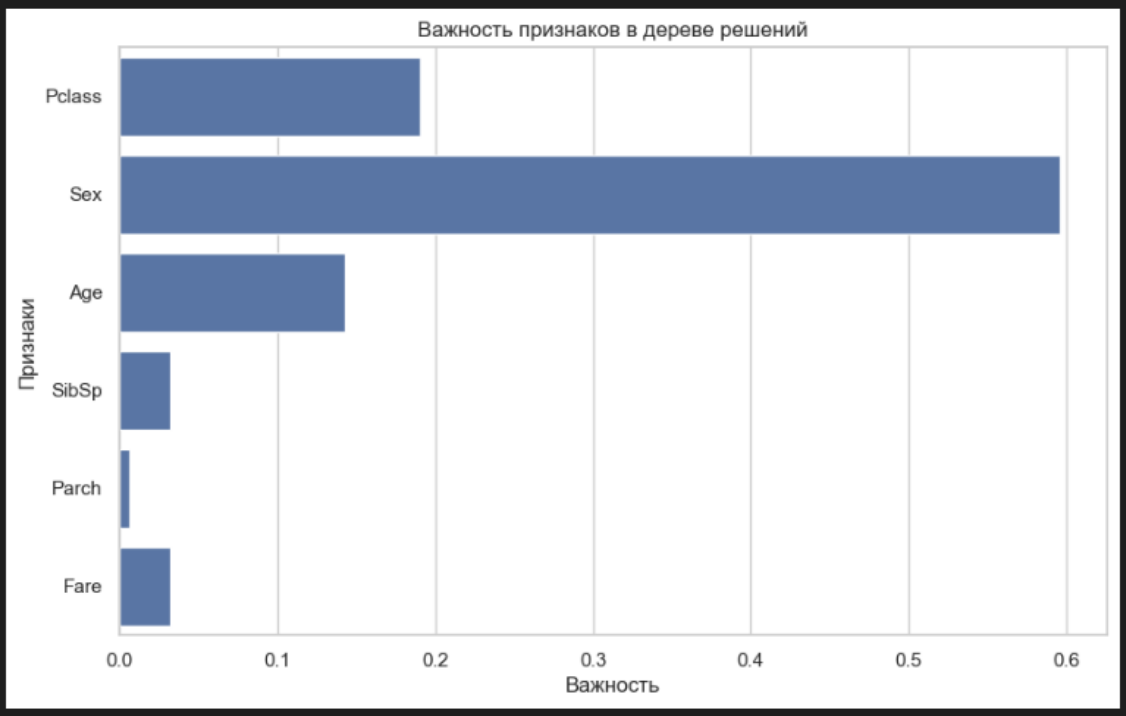
ИУ5Ц-84Б_Тихонова_лр4.ipynb

D: > Загрузки Firefox > ИУ5Ц-84Б_Тихонова_лр4.ipynb > Лабораторная работа: Линейные модели, SVM и деревья решений > import pandas as pd

Generate | Code | Markdown | Run All | Restart | Clear All Outputs | Jupyter Variables | Outline | ...

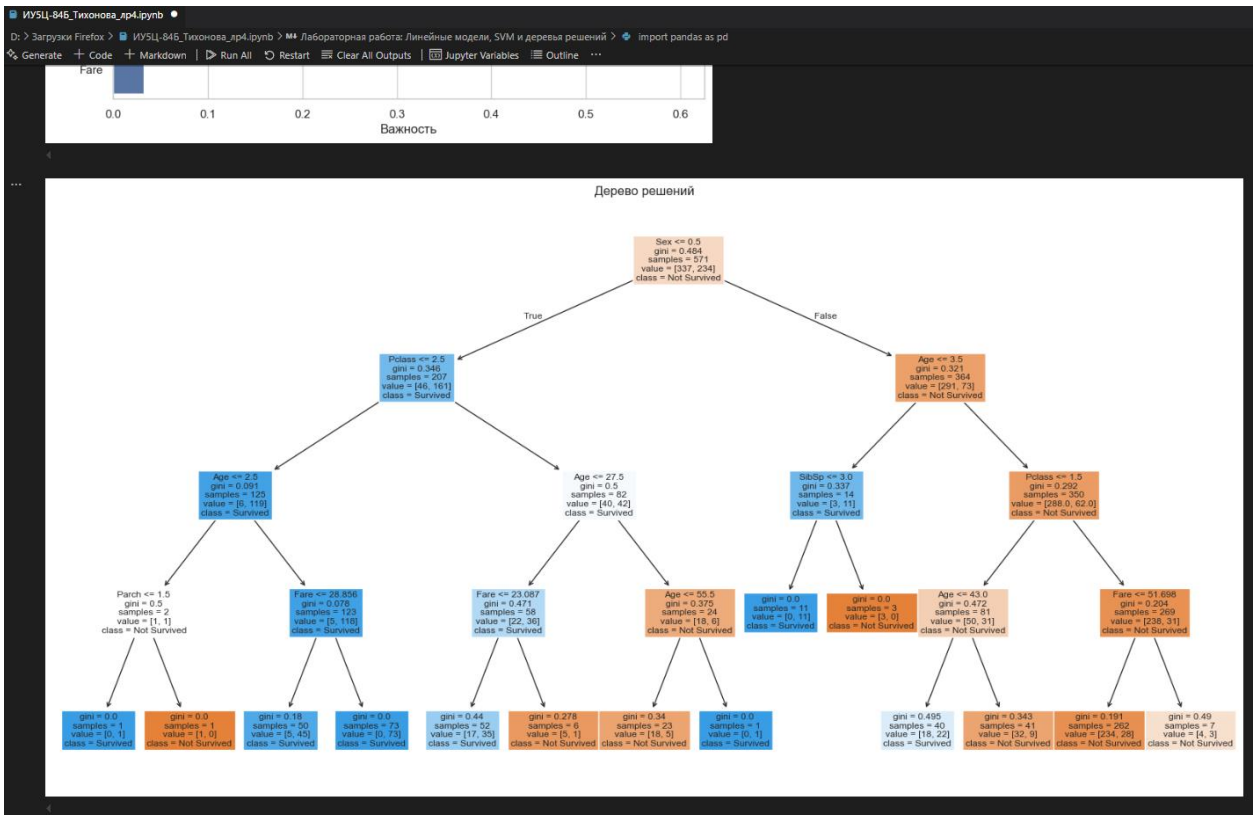
```
features = X.columns
plt.figure(figsize=(10, 6))
sns.barplot(x=importances, y=features)
plt.title('Важность признаков в дереве решений')
plt.xlabel('Важность')
plt.ylabel('Признаки')
plt.show()

# Визуализация дерева решений
plt.figure(figsize=(20, 10))
plot_tree(tree_model, feature_names=features, class_names=['Not Survived', 'Survived'], filled=True)
plt.title('Дерево решений')
plt.show()
```

[7] ✓ 1.0s



Дерево решений

Sex <= 0.5
gini = 0.484
samples = 571
value = [337, 234]
class = Not Survived


ИУ5Ц-84Б_Тихонова_лр4.ipynb
D: > Загрузки Firefox > ИУ5Ц-84Б_Тихонова_лр4.ipynb > Лабораторная работа: Линейные модели, SVM и деревья решений > import pandas as pd
Generate
Code
Markdown
Run All
Restart
Clear All Outputs
Jupyter Variables
Outline
Fare
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Важность
Дерево решений
Sex <= 0.5
gini = 0.484
samples = 571
value = [337, 234]
class = Not Survived
True
False
Pclass <= 2.5
gini = 0.346
samples = 207
value = [46, 161]
class = Survived
Age <= 3.5
gini = 0.321
samples = 364
value = [291, 73]
class = Not Survived
Age <= 2.5
gini = 0.091
samples = 125
value = [6, 119]
class = Survived
Age <= 27.5
gini = 0.5
samples = 82
value = [40, 42]
class = Survived
SibSp <= 3.0
gini = 0.337
samples = 14
value = [3, 11]
class = Survived
Pclass <= 1.5
gini = 0.292
samples = 350
value = [288.0, 62.0]
class = Not Survived
Parch <= 1.5
gini = 0.5
samples = 2
value = [1, 1]
class = Not Survived
Fare <= 28.856
gini = 0.078
samples = 123
value = [5, 118]
class = Survived
Fare <= 23.087
gini = 0.471
samples = 58
value = [22, 36]
class = Survived
Age <= 55.5
gini = 0.375
samples = 24
value = [18, 6]
class = Not Survived
gini = 0.0
samples = 11
value = [0, 11]
class = Survived
gini = 0.0
samples = 3
value = [3, 0]
class = Not Survived
Age <= 43.0
gini = 0.472
samples = 81
value = [50, 31]
class = Not Survived
Fare <= 51.696
gini = 0.204
samples = 269
value = [238, 31]
class = Not Survived
gini = 0.0
samples = 1
value = [0, 1]
class = Survived
gini = 0.0
samples = 1
value = [1, 0]
class = Not Survived
gini = 0.18
samples = 50
value = [5, 45]
class = Survived
gini = 0.0
samples = 73
value = [0, 73]
class = Survived
gini = 0.44
samples = 52
value = [17, 35]
class = Survived
gini = 0.278
samples = 6
value = [5, 1]
class = Not Survived
gini = 0.34
samples = 23
value = [18, 5]
class = Not Survived
gini = 0.0
samples = 1
value = [0, 1]
class = Survived
gini = 0.495
samples = 40
value = [18, 22]
class = Survived
gini = 0.343
samples = 41
value = [32, 9]
class = Not Survived
gini = 0.191
samples = 262
value = [234, 28]
class = Not Survived
gini = 0.49
samples = 7
value = [4, 3]
class = Not Survived