



**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 №5  
«Ансамбли моделей машинного обучения. Часть 1»  
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 ансамблей моделей машинного обучения.

## **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 ансамблевые модели:

две модели группы бэггинга (бэггинг или случайный лес или сверхслучайные деревья);

AdaBoost;

градиентный бустинг.

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

### 3. Ход работы

ИУСЦ-84Б\_Тихонова\_лр4.ipynbИУСЦ-84Б\_Тихонова\_ЛР5\_часть1.ipynb

D: > ТМО > ЛР5 > ИУСЦ-84Б\_Тихонова\_ЛР5\_часть1.ipynb > \*\* Лабораторная работа: Ансамбли моделей машинного обучения (Часть 1)

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

Лабораторная работа: Ансамбли моделей машинного обучения (Часть 1)

Цель: Изучение ансамблевых методов машинного обучения: бэггинг, случайный лес, AdaBoost и градиентный бустинг.

import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
from sklearn.model\_selection import train\_test\_split  
from sklearn.preprocessing import LabelEncoder  
from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix  
  
from sklearn.ensemble import BaggingClassifier, RandomForestClassifier, ExtraTreesClassifier, AdaBoostClassifier, GradientBoostingClassifier  
  
sns.set(style="whitegrid")

[1] ✓ 35s

# Используем снова Titanic для классификации  
url = 'https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv'  
df = pd.read\_csv(url)  
  
# Первые строки  
print("Первые 5 строк датасета:")  
display(df.head())  
  
# Информация  
print("\nИнформация о датасете:")  
df.info()  
  
# Статистика  
print("\nСтатистика по числовым признакам:")  
display(df.describe(include='all'))

[2] ✓ 0.6s

Первые 5 строк датасета:

	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	Heikinen, 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-845\_Тихонова\_лр4.ipynbИУ5Ц-845\_Тихонова\_ЛР5\_часть1.ipynb

D: > ТМО > ЛР5 > ИУ5Ц-845\_Тихонова\_ЛР5\_часть1.ipynb > М\* Лабораторная работа: Ансамбли моделей машинного обучения (Часть 1)

GenerateCodeMarkdownRun AllRestartClear All OutputsJupyter VariablesOutline

4503Allen, Mr. William Henrymale35.0003734508.0500NaN

...

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

# Предобработка  
df = df[['Survived', 'Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare']]  
df.dropna(inplace=True)  
  
# Кодирование пола  
df['Sex'] = LabelEncoder().fit\_transform(df['Sex'])  
  
# Разделение признаков и целевой переменной  
X = df.drop('Survived', axis=1)

```
ИУ5Ц-84Б_Тихонова_лр4.ipynb • ИУ5Ц-84Б_Тихонова_ЛР5_часть1.ipynb •
D: > ТМО > ЛР5 > ИУ5Ц-84Б_Тихонова_ЛР5_часть1.ipynb > М* Лабораторная работа: Ансамбли моделей машинного обучения (Часть 1)
Generate + Code + Markdown ▶ Run All ⌂ Restart ≡ Clear All Outputs | Jupyter Variables ≡ Outline ...

# Предобработка
df = df[['Survived', 'Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare']]
df.dropna(inplace=True)

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

# Разделение признаков и целевой переменной
X = df.drop('Survived', axis=1)
y = df['Survived']

# Разделение 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}")

[3] ✓ 0.0s
...
Размер обучающей выборки: (571, 6)
Размер тестовой выборки: (143, 6)

▶ # Бэггинг с базовым классификатором (дерево решений)
bagging_model = BaggingClassifier(n_estimators=100, random_state=42)
bagging_model.fit(X_train, y_train)

# Случайный лес
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Сверхслучайные деревья (Extra Trees)
et_model = ExtraTreesClassifier(n_estimators=100, random_state=42)
et_model.fit(X_train, y_train)

# AdaBoost
ada_model = AdaBoostClassifier(n_estimators=100, random_state=42)
ada_model.fit(X_train, y_train)

# Градиентный бустинг
gb_model = GradientBoostingClassifier(n_estimators=100, random_state=42)
gb_model.fit(X_train, y_train)

[4] ✓ 1.2s
...
▼ GradientBoostingClassifier ⓘ ?
GradientBoostingClassifier(random_state=42)

def evaluate_model(name, model, X_test, y_test):
```

ИУ5Ц-845\_Тихонова\_лр4.ipynbИУ5Ц-845\_Тихонова\_ЛР5\_часть1.ipynb

D: > ТМО > ЛР5 > ИУ5Ц-845\_Тихонова\_ЛР5\_часть1.ipynb > М\* Лабораторная работа: Ансамбли моделей машинного обучения (Часть 1)

GenerateCodeMarkdownRun AllRestartClear All OutputsJupyter VariablesOutline

```
def evaluate_model(name, model, X_test, y_test):
    y_pred = model.predict(X_test)
    print(f"\n{name}")
    print(f"Accuracy: {accuracy_score(y_test, y_pred):.2f}")
    print("Confusion Matrix:")
    print(confusion_matrix(y_test, y_pred))
    print("Classification Report:")
    print(classification_report(y_test, y_pred))

evaluate_model("Бэггинг (BaggingClassifier)", bagging_model, X_test, y_test)
evaluate_model("Случайный лес (RandomForest)", rf_model, X_test, y_test)
evaluate_model("Сверхслучайные деревья (ExtraTrees)", et_model, X_test, y_test)
evaluate_model("AdaBoost", ada_model, X_test, y_test)
evaluate_model("Градиентный бустинг", gb_model, X_test, y_test)
```

[5] ✓ 0.1s

...

Бэггинг (BaggingClassifier)

Accuracy: 0.75

Confusion Matrix:

[[68 19]
 [17 39]]

Classification Report:

	precision	recall	f1-score	support
0	0.80	0.78	0.79	87
1	0.67	0.70	0.68	56
accuracy			0.75	143
macro avg	0.74	0.74	0.74	143
weighted avg	0.75	0.75	0.75	143

...

Случайный лес (RandomForest)

Accuracy: 0.80

Confusion Matrix:

[[73 14]
 [15 41]]

Classification Report:

	precision	recall	f1-score	support
...				
accuracy			0.78	143
macro avg	0.77	0.77	0.77	143
weighted avg	0.78	0.78	0.78	143

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

# Важность признаков: Случайный лес и Градиентный бустинг

ИУ5Ц-845\_Тихонова\_лр4.ipynbИУ5Ц-845\_Тихонова\_ЛР5\_часть1.ipynb

D: > ТМО > ЛР5 > ИУ5Ц-845\_Тихонова\_ЛР5\_часть1.ipynb > М\* Лабораторная работа: Ансамбли моделей машинного обучения (Часть 1)

GenerateCodeMarkdownRun AllRestartClear All OutputsJupyter VariablesOutline

```
# Важность признаков: Случайный лес и Градиентный бустинг
def plot_feature_importance(model, model_name):
    importances = model.feature_importances_
    indices = np.argsort(importances)[::-1]
    plt.figure(figsize=(8, 4))
    plt.title(f"Важность признаков ({model_name})")
    sns.barplot(x=importances[indices], y=X.columns[indices], palette="viridis")
    plt.tight_layout()
    plt.show()

plot_feature_importance(rf_model, "Random Forest")
plot_feature_importance(gb_model, "Gradient Boosting")
```

[6] ✓ 0.5s

... C:\Users\tdash\AppData\Local\Temp\ipykernel\_16464\2275057930.py:7: FutureWarning:
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=F
sns.barplot(x=importances[indices], y=X.columns[indices], palette="viridis")
...

Важность признаков (Random Forest)

Feature	Importance
Age	0.29
Fare	0.27
Sex	0.27
Pclass	0.08
SibSp	0.04
Parch	0.03

... C:\Users\tdash\AppData\Local\Temp\ipykernel\_16464\2275057930.py:7: FutureWarning:
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=F
sns.barplot(x=importances[indices], y=X.columns[indices], palette="viridis")
...

Важность признаков (Gradient Boosting)

ИУ5Ц-846\_Тихонова\_лр4.ipynbИУ5Ц-846\_Тихонова\_лр5\_часть1.ipynb

D:\ТМО > лр5 > ИУ5Ц-846\_Тихонова\_лр5\_часть1.ipynb > ММ Лабораторная работа: Ансамбли моделей машинного обучения (Часть 1)

GenerateCodeMarkdownRun AllRestartClear All OutputsJupyter VariablesOutline

Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'y' variable to 'hue' and set 'legend=False' for the same effect.

sns.barplot(x=importances[indices], y=X.columns[indices], palette="viridis")

...

### Важность признаков (Gradient Boosting)

Feature	Importance
Sex	0.45
Pclass	0.18
Age	0.18
Fare	0.12
SibSp	0.05
Parch	0.01

# Сравнение accuracy моделей

model\_names = ["Bagging", "RandomForest", "ExtraTrees", "AdaBoost", "GradientBoosting"]

models = [bagging\_model, rf\_model, et\_model, ada\_model, gb\_model]

accuracies = [accuracy\_score(y\_test, m.predict(X\_test)) for m in models]

plt.figure(figsize=(8, 5))

sns.barplot(x=accuracies, y=model\_names, palette="Set2")

plt.title("Сравнение Accuracy моделей")

plt.xlabel("Accuracy")

plt.xlim(0.7, 1.0)

plt.tight\_layout()

plt.show()

[7] ✓ 03s

C:\Users\tdash\AppData\Local\Temp\ipykernel\_16464\684681739.py:7: FutureWarning:

Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'y' variable to 'hue' and set 'legend=False' for the same effect.

sns.barplot(x=accuracies, y=model\_names, palette="Set2")

...

### Сравнение Accuracy моделей

Model	Accuracy
Bagging	0.85



