



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА - Российский технологический университет»

РТУ МИРЭА

Институт Информационных Технологий
Кафедра прикладной математики

ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 7

«Ансамблевое обучение»

по дисциплине

«Технологии и инструментарий анализа больших данных»

Выполнил студент группы

Лазарев А. В.

ИБО-03-21

Принял преподаватель кафедры прикладной
математики

Тетерин Н.Н.

Практическая работа выполнена

« __ » _____ 2024 г.

«Зачтено»

« __ » _____ 2024 г.

Москва 2024

СОДЕРЖАНИЕ

1 РЕШЕНИЕ ЗАДАЧ	3
1.1 Задача №1	3
1.2 Задача №2	3
1.3 Задача №3	4
1.4 Задача №4	4

1 РЕШЕНИЕ ЗАДАЧ

1.1 Задача №1

Решение программы представлено на Рисунке 1.1.

```
[ ] url = 'https://raw.githubusercontent.com/InspectorJelly/BigDataMirea/refs/heads/main/datasets/data_Statistics.csv'
data = pd.read_csv(url)
```

Рисунок 1.1 – Программа

1.2 Задача №2

Решение и результат программы представлены на Рисунке 1.2.

```
numerical_features = [
    'Placed', 'Eliminations', 'Assists', 'Revives', 'Accuracy', 'Hits',
    'Head Shots', 'Distance Traveled', 'Materials Gathered',
    'Materials Used', 'Damage Taken', 'Damage to Players',
    'Damage to Structures'
]
X = data[numerical_features]

data['Target'] = (data['Eliminations'] > 3).astype(int)
y = data['Target']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

base_model = DecisionTreeClassifier(max_depth=5, random_state=42)

# Настройка баггинг-классификатора
bagging_model = BaggingClassifier(
    estimator=base_model, # Базовая модель
    n_estimators=50, # Количество базовых моделей
    max_samples=0.8, # Доля выборки для каждой модели
    max_features=1.0, # Доля признаков для каждой модели
    random_state=42,
    n_jobs=-1 # Использование всех ядер процессора
)

starttime = time.time()
bagging_model.fit(X_train, y_train)
time_bagging = time.time() - starttime

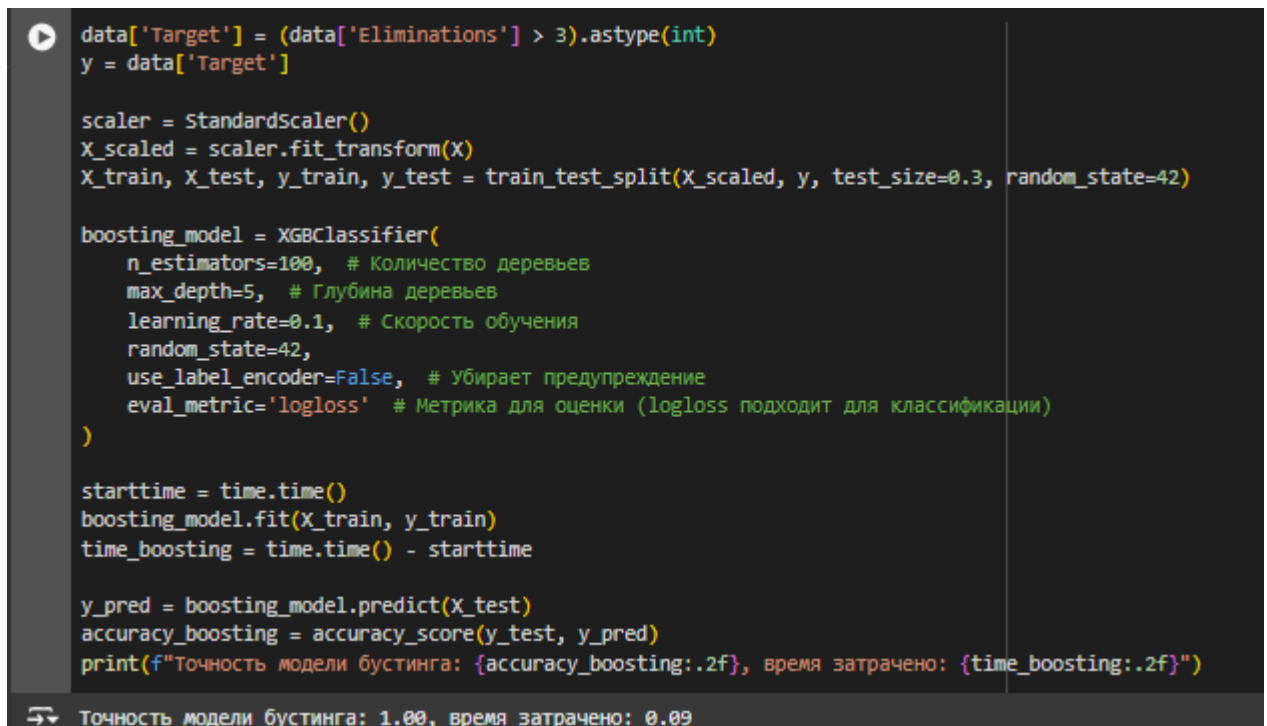
y_pred = bagging_model.predict(X_test)
accuracy_bagging = accuracy_score(y_test, y_pred)
print(f"Точность баггинг-классификатора: {accuracy_bagging:.2f}, время затрачено: {time_bagging:.2f}")
```

Точность баггинг-классификатора: 1.00, время затрачено: 0.17

Рисунок 1.2 – Программа и результат ее выполнения

1.3 Задача №3

Решение и результат программы представлены на Рисунке 1.3.



```
data['Target'] = (data['Eliminations'] > 3).astype(int)
y = data['Target']

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, random_state=42)

boosting_model = XGBClassifier(
    n_estimators=100, # Количество деревьев
    max_depth=5, # Глубина деревьев
    learning_rate=0.1, # Скорость обучения
    random_state=42,
    use_label_encoder=False, # Убирает предупреждение
    eval_metric='logloss' # Метрика для оценки (logloss подходит для классификации)
)

starttime = time.time()
boosting_model.fit(X_train, y_train)
time_boosting = time.time() - starttime

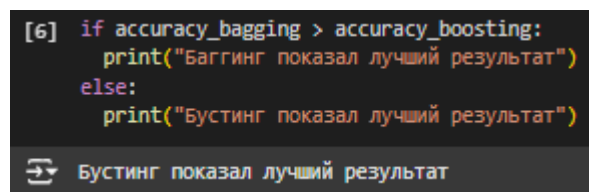
y_pred = boosting_model.predict(X_test)
accuracy_boosting = accuracy_score(y_test, y_pred)
print(f"Точность модели бустинга: {accuracy_boosting:.2f}, время затрачено: {time_boosting:.2f}")
```

Точность модели бустинга: 1.00, время затрачено: 0.09

Рисунок 1.3 – Программа

1.4 Задача №4

Решение и результат программы представлены на Рисунках 1.4.



```
[6] if accuracy_bagging > accuracy_boosting:
    print("Баггинг показал лучший результат")
else:
    print("Бустинг показал лучший результат")
```

Бустинг показал лучший результат

Рисунок 1.4 – Программа