

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«Омский государственный технический университет»

Факультет информационных технологий и компьютерных систем
Кафедра «Автоматизированные системы обработки информации и управления»

КУРСОВОЙ ПРОЕКТ

по дисциплине Динамические языки программирования
на тему Классификация данных о клиентах банка

Пояснительная записка

Шифр проекта 043-КП-09.03.04-№20-ПЗ

Студента (ки) Сидорина Артёма Алексеевича
фамилия, имя, отчество полностью

Курс 4 Группа ПИН-202

Направление (специальность) 09.03.04 – Программная инженерия
код, наименование

Руководитель Старший преподаватель
ученая степень, звание
Кабанов.А.А
фамилия, инициалы

Выполнил (а) _____
дата, подпись студента (ки)

К защите _____
дата, подпись руководителя

Проект (работа) защищен (а) с оценкой _____

Набранные баллы: _____
в семестре на защите Итого

Омск 2023

РЕФЕРАТ

Пояснительная записка к курсовому проекту 23 с., 1 ч., 10 рис., 1 табл., 8 источн., 0 прил.

МАШИННОЕ ОБУЧЕНИЕ, КЛАССИФИКАЦИЯ, ТОЧНОСТЬ, МЕТОДЫ НАИВНОГО БАЙЕСА, САТ BOOST, МЕТОД К-БЛИЖАЙШИХ СОСЕДЕЙ, МЕТОД ОПОРНЫХ ВЕКТОРОВ, МЕТОД ЛИНЕЙНОЙ РЕГРЕССИИ, МЕТОД ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ, РЕШАЮЩИЕ ДЕРЕВЬЯ, СЛУЧАЙНЫЙ ЛЕС, ДАТАСЕТ

Объектами исследования являются методы машинного обучения для классификации данных.

Цель работы – реализация и сравнение различных методов машинного обучения для классификации данных выбранного датасета.

В процессе работы производилась разработка и реализация алгоритмов машинного обучения на языке программирования высокого уровня Python, а также сравнение методов друг с другом.

В результате работы были реализованы различные методы обучения и выбран тот, который показал себя лучше всего.

СОДЕРЖАНИЕ

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.....	3
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ	5
ВВЕДЕНИЕ.....	6
ОПИСАНИЕ ДАТАСЕТА	7
ОБРАБОТКА ДАТАСЕТА	9
ОБУЧЕНИЕ МОДЕЛЕЙ.....	11
1. Метод k-ближайших соседей.....	11
2. Метод машины опорных векторов.....	12
3. Метод линейной регрессии	14
4. Метод логистической регрессии	15
5. Методы наивного Байеса.....	16
6. Решающее дерево.....	17
7. Случайный лес.....	18
8. Метод CatBoost.....	19
СРАВНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ	21
ЗАКЛЮЧЕНИЕ	22
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	23

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В данном курсовом проекте используются следующие термины с соответствующими определениями.

Машинное обучение – область исследования, в которой компьютерные системы автоматически обучаются на основе данных и алгоритмов, без явного программирования;

Метод К-ближайших соседей – метод обучения, при котором объект присваивается к тому классу, который является наиболее распространённым среди k соседей данного элемента, классы которых уже известны;

Машина опорных векторов – алгоритм машинного обучения, строящий гиперплоскость в пространстве признаков для максимального разделения объектов разных классов;

Линейная регрессия – метод анализа данных, который предсказывает ценность неизвестных данных с помощью другого, связанного и известного значения данных;

Логистическая регрессия – метод анализа данных, который использует математику для поиска взаимосвязей между двумя факторами данных. Затем эта взаимосвязь используется для прогнозирования значения одного из этих факторов на основе другого;

Классификатор наивного Байеса – это группа алгоритмов, построенных на принципе байесовской классификации с предположением о независимости между признаками (наивное предположение);

Решающие деревья - алгоритмы машинного обучения, строящие структуру дерева для принятия решений на основе признаков объектов;

Случайный лес - ансамбль решающих деревьев, объединенных для улучшения обобщающей способности и уменьшения переобучения;

CatBoost – метод машинного обучения, который формирует прогностическую модель в виде ансамбля слабых моделей, чаще всего деревьев решений. Этот метод пошагово создает модель, что позволяет оптимизировать произвольную дифференцируемую функцию потерь;

Датасет – совокупность данных, систематизированных в определённом формате, представляющих собой базовый элемент для работы с данными во многих отраслях.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В данном курсовом проекте используются следующие сокращения с соответствующими обозначениями.

ML – Машинное обучение;

K-NN – Метод k-ближайших соседей;

SVM – Машина опорных векторов.

ВВЕДЕНИЕ

Курсовой проект по дисциплине «Динамические языки программирования», 4 курс. В проекте используется язык программирования высокого уровня Python.

Задача: реализовать различные методы машинного обучения, произвести с помощью них анализ датасета и произвести сравнение методов между собой для выбора лучшего из них, ориентируясь на точности предсказаний и времени, затраченном на поиск оптимальных параметров модели.

ОПИСАНИЕ ДАТАСЕТА

Набор данных содержит информацию о клиентах банка. Целевым столбцом является столбец Loan Status, указывающий, был ли уплачен долг во время или нет.

Датасет содержит следующие столбцы:

ID - уникальный идентификатор представителя.

Loan Amount – применяемая сумма кредита.

Funded Amount - финансируемая сумма кредита.

Funded Amount Investor - сумма кредита, одобренная инвесторами.

Term - срок кредита (в месяцах).

Batch Enrolled - номера партий представителям.

Interest Rate - процентная ставка (%) по кредиту.

Grade - оценка банком.

Sub Grade - подкласс, присвоенный банком.

Employment Duration – период.

Verification Status - Подтверждение дохода банком.

Loan Title - предоставленный титул по кредиту.

Debit to Income - отношение общего ежемесячного погашения долга представителя к ежемесячному доходу, о котором он сам сообщил, без учета ипотеки.

Delinquency - two years - количество просрочек более чем на 30 дней за последние 2 года.

Inquires - six months - общее количество запросов за последние 6 месяцев.

Open Account - номер открытой кредитной линии в кредитной линии представителя.

Public Record - количество уничижительных публичных записей.

Revolving Balance - общий возобновляемый остаток по кредиту.

Revolving Utilities - сумма кредита, используемая представителем относительно revolving_balance.

Total Accounts - общее количество кредитных линий, доступных в рамках кредитной линии представителей.

Initial List Status - уникальный статус перечисления займа - W(Ожидание), F(переадресован).

Total Received Interest - общая сумма процентов, полученных на сегодняшний день.

Total Received Late Fee - общая сумма платы за просрочку, полученная до настоящего времени .

Recoveries - списание после списания валового возмещения.

Collection Recovery Fee - списание платы за инкассацию по почте.

Collection 12 months Medical - общее количество сборов за последние 12 месяцев без учета медицинских сборов.

Application Type - указывает, является ли представитель индивидуальным или совместным.

Last week Pay - указывает, в течение какого времени (в неделях) представитель платил EMI после регистрации партии.

Accounts Delinquent - количество счетов, по которым представитель имеет просрочку платежа.

Total Collection Amount - общий текущий баланс по всем счетам.

Total Current Balance - общий текущий баланс по всем счетам.

Total Revolving Credit Limit - общий возобновляемый кредитный лимит.

Loan Status – статус кредита.

Датасет содержит в себе 33 столбца и 67 463 строки.

ОБРАБОТКА ДАТАСЕТА

Перед сравнением методов машинного обучения, необходимо подготовить данные датасета к анализу:

- Кодирование текстовых данных:

Производится кодирование текстовых значений в числовой формат с использованием LabelEncoder.

```
data['Grade Encoded'] = label_encoder.fit_transform(data['Grade'])
data['Sub Grade Encoded'] = label_encoder.fit_transform(data['Sub Grade'])
data['Employment Duration Encoded'] = label_encoder.fit_transform(data['Employment Duration'])
data['Verification Status Encoded'] = label_encoder.fit_transform(data['Verification Status'])
data['Loan Title Encoded'] = label_encoder.fit_transform(data['Loan Title'])
data['Initial List Status Encoded'] = label_encoder.fit_transform(data['Initial List Status'])

non_numeric_columns = data.select_dtypes(exclude=['number']).columns.tolist()
data = data.drop("ID", axis=1)
data = data.drop("Accounts Delinquent", axis=1)

data = data.drop(columns=non_numeric_columns)
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 67463 entries, 0 to 67462
Data columns (total 30 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Loan Amount                               67463 non-null  int64
1   Funded Amount                             67463 non-null  int64
2   Funded Amount Investor                    67463 non-null  float64
3   Term                                       67463 non-null  int64
4   Interest Rate                             67463 non-null  float64
5   Home Ownership                           67463 non-null  float64
6   Debit to Income                           67463 non-null  float64
7   Delinquency - two years                   67463 non-null  int64
8   Inquires - six months                     67463 non-null  int64
9   Open Account                              67463 non-null  int64
10  Public Record                             67463 non-null  int64
11  Revolving Balance                         67463 non-null  int64
12  Revolving Utilities                       67463 non-null  float64
13  Total Accounts                           67463 non-null  int64
14  Total Received Interest                   67463 non-null  float64
15  Total Received Late Fee                   67463 non-null  float64
16  Recoveries                               67463 non-null  float64
17  Collection Recovery Fee                   67463 non-null  float64
```

Рис.1 – Замена текстовых категориальных данных числовыми

Итоговые данные:

Обработанные обучающие данные готовы для внедрения в модели машинного обучения. Теперь датасет представляет собой сбалансированный обучающий набор данных, готовый для обучения моделей.

ОБУЧЕНИЕ МОДЕЛЕЙ

1. Метод k-ближайших соседей

Метод k-ближайших соседей представляет собой метрический алгоритм, применяемый для автоматической классификации объектов или выполнения регрессии. При использовании данного метода для классификации, каждый объект получает присвоенный ему класс, который определяется на основе наиболее часто встречающегося класса среди k ближайших соседей данного элемента. В данном контексте, параметр k представляет собой заданное количество соседей, классы которых предварительно известны.

Гиперпараметры метода k-ближайших соседей:

- k – число соседей (Определяет насколько далеко в пространстве признаков смотрит алгоритм);
- метрика расстояния (Метрика, с помощью которой определяется расстояние между соседями).

В приведённом методе были использованы следующие гиперпараметры для поиска оптимальных параметров:

- k: от 1 до 40;
- метрика расстояния: Евклидова, Манхэтанская, Чебышева, Минковски.

В результате поиска лучших гиперпараметров были получены следующие параметры:

- k – 11;
- метрика расстояния: Манхэтанская.

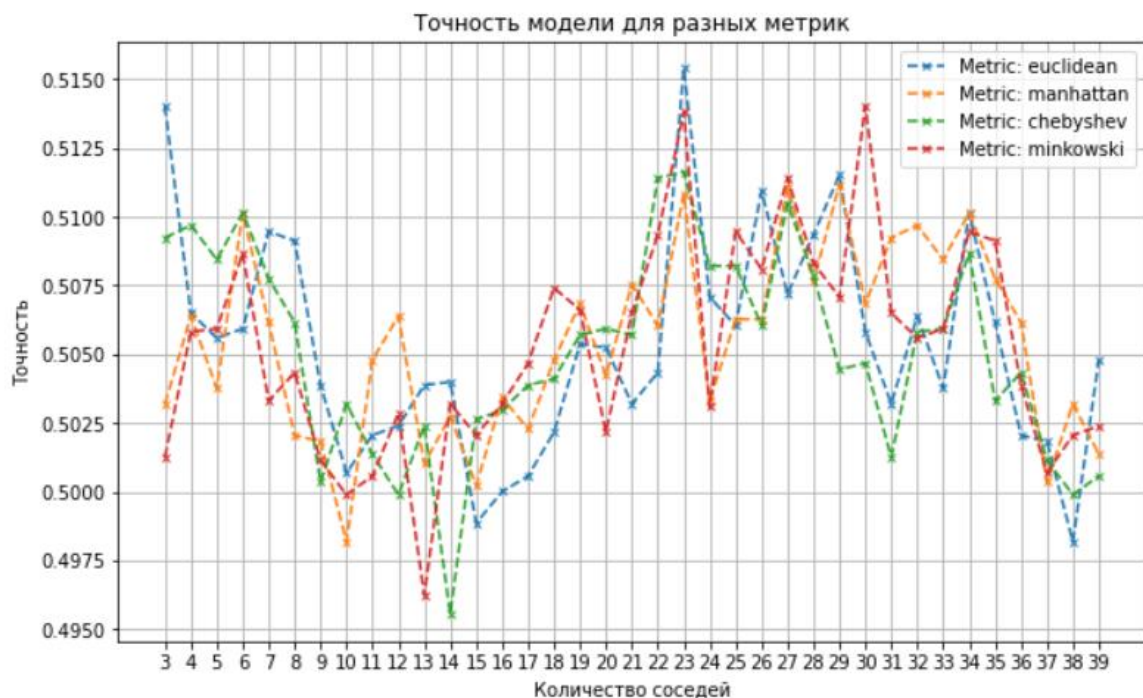


Рис.2 – График поиска лучших гиперпараметров

На основе лучших параметров была получена следующая точность модели:

```

Точность модели с 35 соседями и метрикой manhattan: 0.510153412880753
precision    recall  f1-score   support

      0       0.91      0.51      0.66      24528
      1       0.09      0.47      0.15       2458

 accuracy          0.51      26986
 macro avg       0.50      0.49      0.40      26986
 weighted avg    0.83      0.51      0.61      26986
  
```

Рис.3 – Точность на лучших параметрах

Таким образом, метод k-ближайших соседей даёт точность ~51%.

2. Метод машины опорных векторов

Метод машины опорных векторов представляет собой алгоритм машинного обучения, применяемый для решения задач классификации и регрессии. Суть метода заключается в поиске оптимальной гиперплоскости в многомерном пространстве признаков, которая наилучшим образом разделяет объекты разных классов. Основной стратегией SVM является максимизация расстояния между гиперплоскостью и объектами различных классов, что

придаёт ему эффективность даже в случаях с нелинейными разделяющими поверхностями.

Гиперпараметры метода SVM:

- Ядро (выбор ядра влияет на способность модели обрабатывать сложные нелинейные зависимости в данных);
- C – параметр регуляризации (контролирует уровень штрафа за неверную классификацию);
- параметры ядра (некоторые ядра могут иметь дополнительные параметры, такие как степень полинома и коэффициенты).

В приведённом методе были использованы следующие гиперпараметры для поиска оптимальных параметров:

- Ядро: linear, rbf, sigmoid, poly;
- C: 0.001, 0.01, 0.1, 1, 10;
- параметры ядра: 1, 2, 3, 4, 5, 6, 7.

В результате поиска лучших гиперпараметров были получены следующие параметры:

- Ядро: rbf;
- C: 10;
- параметры ядра: 1;

На основе лучших параметров была получена следующая точность модели:

Точность модели: 0.5615396017589802				
	precision	recall	f1-score	support
0	0.91	0.57	0.70	18386
1	0.09	0.43	0.15	1853
accuracy			0.56	20239
macro avg	0.50	0.50	0.43	20239
weighted avg	0.83	0.56	0.65	20239
Средняя точность перекрестной проверки: 0.4997301672962763				

Рис.4 – Точность на лучших параметрах

На основе этого можно сделать вывод, что метод машины опорных векторов на выбранном датасете даёт точность ~56.2%

3. Метод линейной регрессии

Линейная регрессия представляет собой метод, который применяется не только для классификации, но и для прогнозирования вероятности принадлежности к определённому классу.

Гиперпараметры метода линейной регрессии:

- Включение свободного члена (определяет, следует ли включать свободный член в уравнение регрессии. Если установлен False, модель проходит через начало координат).

В приведённом методе были использованы следующие гиперпараметры для поиска оптимальных:

- Включение свободного члена: true, false.

Поиск лучших параметров занял меньше минуты.

В результате были получены следующие значения гиперпараметров:

- Включение свободного члена: true.

На основе лучших параметров была получена следующая точность модели:

	precision	recall	f1-score	support
0	0.91	0.54	0.68	18386
1	0.09	0.47	0.16	1853
accuracy			0.54	20239
macro avg	0.50	0.51	0.42	20239
weighted avg	0.84	0.54	0.63	20239

Рис. 5 – Точность модели на лучших параметрах

Таким образом, для выбранного набора данных метод линейной регрессии показывает точность ~54%.

4. Метод логистической регрессии

Логистическая регрессия – метод анализа данных, который использует математику для поиска взаимосвязей между двумя факторами данных. Затем

эта взаимосвязь используется для прогнозирования значения одного из этих факторов на основе другого.

Гиперпараметры для метода логистической регрессии:

- C – обратная сила регуляризации (контролирует величину штрафа за сложность модели);
- пенальти (определяет тип регуляризации, который применяется к модели).

В приведённом методе использовались следующие гиперпараметры для поиска лучших параметров:

- C: 0.001, 0.01, 0.1, 1, 10;
- пенальти: l1, l2.

В результате были получены следующие параметры:

- C: 0.1;
- пенальти: l1.

На основе лучших параметров была получена следующая точность модели:

	precision	recall	f1-score	support
0	0.91	0.54	0.68	18386
1	0.09	0.47	0.16	1853
accuracy			0.54	20239
macro avg	0.50	0.51	0.42	20239
weighted avg	0.84	0.54	0.63	20239

Рис.6 – Точность модели на лучших параметрах

Таким образом, для выбранного набора данных метод линейной регрессии показывает точность ~54%.

5. Методы наивного Байеса

Классификатор наивного Байеса – это группа алгоритмов, построенных на принципе байесовской классификации с предположением о независимости между признаками (наивное предположение).

Основные методы наивного Байеса и их гиперпараметры:

- Наивный байес с распределением Бернулли:

Гиперпараметры:

- Параметр аддитивного сглаживания (параметр для предотвращения проблемы с нулевыми вероятностями);
- порог бинаризации (задаёт порог для бинаризации входных данных. Все значения, большие этого порога становятся равны 1, а все меньшие или равные – 0).

- наивный байес с мультиномиальным распределением:

Гиперпараметры:

- параметр аддитивного сглаживания.

- наивный байес с нормальным распределением:

Гиперпараметры:

- отсутствуют.

В приведённых методах были выбраны следующие параметры для поиска лучших гиперпараметров:

- Наивный байес с распределением Бернулли:

- Параметр аддитивного сглаживания: 0.1, 0.5, 1;
- порог бинаризации: 0, 0.1, 0.2.

- Наивный байес с мультиномиальным распределением:

- Параметр аддитивного сглаживания: 0.1, 0.5, 1.

- Наивный байес с нормальным распределением:

- Отсутствуют.

В результате были получены следующие результаты:

- Лучшая модель: наивный байес с нормальным распределением.
- Лучшие гиперпараметры: отсутствуют.

На основе лучших параметров была получена следующая точность модели:


```

Точность модели: 0.7620435792282227
      precision    recall  f1-score   support

     0       0.91      0.82      0.86      18386
     1       0.09      0.19      0.13       1853

 accuracy          0.76      20239
 macro avg          0.50      0.50      0.49      20239
 weighted avg       0.83      0.76      0.79      20239

```

Рис.7 – Точность модели с лучшими гиперпараметрами

Таким образом, для выбранного набора данных метод наивного байеса показывает точность ~76,2%.

6. Решающее дерево

Модель решающего дерева представляет собой метод машинного обучения, основанный на структуре дерева для принятия решений. Каждый узел в дереве представляет собой решение, связанное с определенным признаком, а каждый лист дерева представляет собой окончательный ответ в виде класса или значения.

Гиперпараметры для метода решающих деревьев:

- Глубина (ограничивает максимальную глубину дерева);
- минимальное количество объектов для разделения (устанавливает минимальное количество объектов, необходимых для разделения внутреннего узла);
- минимальное количество объектов в листе (определяет минимальное количество объектов, которые должны находиться в листе).

В методе решающих деревьев были использованы следующие значения параметров для поиска лучших гиперпараметров:

- Глубина: 5, 10, 15, 20;
- минимальное количество объектов для разделения: 2, 5, 10;
- минимальное количество объектов в листе: 1, 2, 4.

В результате были получены следующие значение гиперпараметров:

- Глубина: 15;

- минимальное количество объектов для разделения: 10;
- минимальное количество объектов в листе: 2.

На основе лучших параметров была получена следующая точность модели:

```
Decision Tree с лучшими параметрами - Точность на тесте: 0.46632738771678445
Classification Report для Decision Tree:
              precision    recall  f1-score   support

     0           0.91       0.46       0.61       18386
     1           0.09       0.56       0.16        1853

 accuracy                   0.47       20239
 macro avg              0.50       0.51       0.38       20239
 weighted avg           0.84       0.47       0.57       20239
```

Рис.8 – Точность модели с лучшими гиперпараметрами

Таким образом, для выбранного набора данных метод решающего дерева выдаёт точность ~46.6%.

7. Случайный лес

Случайный лес представляет собой совокупность деревьев решений, где каждое дерево обучается на случайной подвыборке данных, а затем прогнозы объединяются или усредняются для формирования окончательного результата.

Гиперпараметры для метода случайного леса:

- Количество деревьев;
- глубина (ограничивает максимальную глубину дерева);
- минимальное количество объектов для разделения (устанавливает минимальное количество объектов, необходимых для разделения внутреннего узла);
- минимальное количество объектов в листе (определяет минимальное количество объектов, которые должны находиться в листе).

В методе случайного леса были использованы следующие значения параметров для поиска лучших гиперпараметров:

- Количество деревьев: 50, 100, 150, 200;

- глубина: 5, 10, 15;
- минимальное количество объектов для разделения: 2, 5, 10;
- минимальное количество объектов в листе: 1, 2, 4.

В результате были получены следующие значения гиперпараметров:

- Количество деревьев: 200;
- глубина: 15;
- минимальное количество объектов для разделения: 10;
- минимальное количество объектов в листе: 2.

На основе лучших параметров была получена следующая точность модели:

Random Forest с лучшими параметрами - Точность на тесте: 0.5445427145609961

Classification Report для Random Forest:

	precision	recall	f1-score	support
0	0.92	0.55	0.69	18386
1	0.10	0.50	0.17	1853
accuracy			0.54	20239
macro avg	0.51	0.53	0.43	20239
weighted avg	0.84	0.54	0.64	20239

Рис.9 – Точность модели с лучшими гиперпараметрами

Таким образом, для выбранного набора данных метод случайного леса выдаёт точность ~54.4%.

8. Метод CatBoost

CatBoost — метод машинного обучения, который формирует прогностическую модель в виде ансамбля слабых моделей, чаще всего деревьев решений. Этот метод пошагово создает модель, что позволяет оптимизировать произвольную дифференцируемую функцию потерь.

Гиперпараметры для метода CatBoost:

- Количество итераций (определяет количество базовых моделей, которые будут обучены);
- скорость обучения (контролирует величину шага градиентного спуска);
- глубина (определяет максимальную глубину деревьев).

В методе CatBoost были использованы следующие значения параметров для поиска лучших гиперпараметров:

- Количество итераций: 100, 200, 300;
- скорость обучения: 0.01, 0.05, 0.1;
- глубина: 4, 6, 8.

Поиск лучших параметров занял 7 минут.

В результате были получены следующие значение гиперпараметров:

- Количество итераций: 300;
- скорость обучения: 0.1;
- глубина: 8.

На основе лучших параметров была получена следующая точность модели:

```
Точность CatBoost на тестовых данных: 0.5314985918276595
Classification Report для CatBoost:
              precision    recall  f1-score   support

     0               0.91         0.54         0.68       18386
     1               0.10         0.48         0.16        1853

   accuracy                   0.53       20239
  macro avg                   0.50         0.51         0.42       20239
 weighted avg                   0.84         0.53         0.63       20239
```

Рис.10 – Точность модели с лучшими гиперпараметрами

Таким образом, для выбранного набора данных метод CatBoost выдаёт точность ~53%.

СРАВНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

В приведенной таблице 1 показано сравнение всех используемых методом машинного обучения.

Таблица 1 – Сравнение методов машинного обучения

Название метода	Точность (%)	Время (мин.)
Метод k-ближайших соседей	51	5
Метод машины опорных векторов	56,2	15
Метод линейной регрессии	54	<1
Метод логистической регрессии	54	5
Метод наивный байесовский алгоритм Гаусса	76,2	<1
Метод решающего дерева	46,6	2
Метод случайного леса	54,4	20
Метод CatBoost	53	<1

Среди всех методов выделяется метод наивного байесовского алгоритма Гаусса с точностью 76,6%. Он показывает наивысшую точность при малом времени поиска параметров.

ЗАКЛЮЧЕНИЕ

В процессе выполнения курсового проекта был найден и подготовлен к анализу набор данных о клиентах банка. После этого, был проведён анализ данного датасета различными методами машинного обучения для решения задач классификации. Затем, методы были сравнены между собой по критериям точности анализа и времени, затраченного на оптимизацию параметров. В результате лучше всего себя показал метод наивного байесовского алгоритма Гаусса с точностью 76,6% и временем на поиск гиперпараметров менее одной минуты.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Метод k-ближайших соседей (k-nearest neighbour) – URL: <https://proglib.io/p/metod-k-blizhayshih-sosedey-k-nearest-neighbour-2021-07-19> (дата обращения 18.12.2023). – Текст: электронный.
2. Классификация данных методом k-ближайших соседей – URL: <https://loginom.ru/blog/knn> (дата обращения 18.12.2023). – Текст: электронный.
3. Наивные методы Байеса – URL: <https://scikit-learn.ru/1-9-naive-bayes> (дата обращения 21.12.2023). – Текст: электронный.
4. Краткий обзор алгоритма машинного обучения Метод Опорных Векторов (SVM) – URL: <https://habr.com/ru/articles/428503/> (дата обращения 18.12.2023). – Текст: электронный.
5. Машинное обучение для начинающих: алгоритм случайного леса (Random Forest) – URL: <https://proglib.io/p/mashinnoe-obuchenie-dlya-nachinayushchih-algoritm-sluchaynogo-lesa-random-forest-2021-08-12> (дата обращения 21.12.2023). – Текст: электронный.
6. Метод классификации CatBoost – URL: <https://neerc.ifmo.ru/wiki/index.php?title=CatBoost> (дата обращения 22.12.2023). – Текст: электронный.
7. Метод k-ближайших соседей – URL: https://ru.wikipedia.org/wiki/Метод_k-ближайших_соседей (дата обращения 18.12.2023). – Текст: электронный.
8. Классификация методом решающих деревьев. – URL: <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html> (дата обращения 21.12.2023). – Текст: электронный.