Изображение выглядит как текст

Автоматически созданное описание

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра анализа данных и искусственного интеллекта**

Допустить к защите

заведующий кафедрой

д-р тех. наук, доцент

\_\_\_\_\_\_\_\_\_\_\_ А. В. Коваленко

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Руководитель ООП

д-р физ.-мат. наук, профессор

\_\_\_\_\_\_\_\_\_\_\_\_ М. Х. Уртенов

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**(МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)**

**Разработка современных прогностических систем в банковской сфере**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И. А. Бордонос

(подпись)

Направление подготовки 01.04.02 Прикладная математика и информатика

Направленность (профиль) Математическое и информационное обеспечение экономической деятельности

Научный руководитель

д-р тех. наук, профессор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Халафян

(подпись)

Нормоконтролер

канд. физ.-мат наук, доцент *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* Г.В. Калайдина

(подпись)

Краснодар

2022

# РЕФЕРАТ

Магистерская диссертация 55 с., 4 ч., 13 рис., 22 табл., 9 источников.

МАШИННОЕ ОБУЧЕНИЕ, АНСАМБЛИРОВАНИЕ, БЭГГИНГ, СТЭКИНГ, АДАБУСТ, ГРАДИЕНТНЫЙ БУСТИНГ, SCIKIT-LEARN, PYTHON

Объект исследования: методы ансамблирования моделей машинного обучения.

Цель работы – реализация и сравнительный анализ четырех методов ансамблирования моделей машинного обучения применительно к задаче классификации банков по их экономическим показателям.

Задачи работы:

– изучить математическая часть методов и технология их практической реализации;

– предложить способ получения оценки экономической деятельности кредитных организаций;

– реализовать методы на языке Python применительно к задаче получения приведённой выше оценки, на основе их показателей;

– провести сравнительный анализ методов;

– создать пользовательский интерфейс для взаимодействия с ними.

Итог работы – рассмотрены четыре метода ансамблирования, а именно – бэггинг, стэкинг, адабуст и градиентный бустинг, их прогностическая способность сравнена между собой. Посредством библиотек Scikit-Learn, и PyQt5 языка программирования Python осуществлена программная реализация методов.

СОДЕРЖАНИЕ

[Введение 5](#_Toc105914996)

[1 Теоретические основы методов ансамблирования 6](#_Toc105914997)

[1.1 Постановка задачи машинного обучения с учителем 6](#_Toc105914998)

[1.2 Ансамбли машинного обучения 8](#_Toc105914999)

[1.2.1 Бэггинг 10](#_Toc105915000)

[1.2.1.1 Математическое обоснование бэггинга 12](#_Toc105915001)

[1.2.2 Стэкинг 14](#_Toc105915002)

[1.2.2.1 Математическое обоснование стэкинга 16](#_Toc105915003)

[1.2.3 Бустинг 19](#_Toc105915004)

[1.2.3.1 Адабуст 20](#_Toc105915005)

[1.2.3.2 Градиентный бустинг 22](#_Toc105915006)

[2 Постановка задачи и создание входного набора данных 28](#_Toc105915007)

[2.1 Начальные данные и первичная обработка 28](#_Toc105915008)

[2.2 Использование банка консолидированного лидера 30](#_Toc105915009)

[2.3 Неразмеченные данные 32](#_Toc105915010)

[2.4 Кластеризация 34](#_Toc105915011)

[3 Построение моделей 40](#_Toc105915012)

[3.1.1 Реализация методов ансамблирования в Scikit-learn 41](#_Toc105915013)

[3.2 Построенные модели 43](#_Toc105915014)

[3.3 Сравнительный анализ моделей 49](#_Toc105915015)

[4 Пользовательский интерфейс 51](#_Toc105915016)

[Заключение 53](#_Toc105915017)

[Список использованных истчоников 54](#_Toc105915018)

[Приложение А Дополнительные статистические данные 55](#_Toc105915019)

# ВВЕДЕНИЕ

При экономическом взаимодействии в банковской сфере одну из важнейших ролей занимают финансовые показатели подобных организаций. Именно эти показатели во время выбора предложений от кредитных организаций, если такие предложения идентичны, играют решающую роль. Кроме того, они необходимы в оценке целесообразности выдвигаемых предложений.

Одним из вариантов получения такой оценки является применение метрического подхода и разбиение банков на группы однородности по их экономическим показателям, а затем применение систем машинного обучения с учителем для классификации.

Передовым направлением в рамках создания систем машинного обучения является ансамблирование различных методов машинного обучения. Под этим подразумевается объединение нескольких слабых моделей в одну сильную.

В рамках данной работы было рассмотрено четыре основных метода ансамблирования – бэггинг, стэкинг, адабуст и градиентный бустинг.

Цель работы – реализация и сравнительный анализ четырех методов ансамблирования моделей машинного обучения применительно к задаче классификации банков по их экономическим показателям. Для достижения цели были поставлены и решены следующие задачи:

– изучена математическая часть методов и технология их практической реализации;

– предложен и реализован способ получения оценки экономической деятельности кредитных организаций;

– методы реализованы на языке Python применительно к задаче получения приведённой выше оценки, на основе их показателей;

– проведен сравнительный анализ методов;

– построен пользовательский интерфейс для взаимодействия с ними.

# 1 Теоретические основы методов ансамблирования

# 1.1 Постановка задачи машинного обучения с учителем

Пусть задано пространство размерности – пространство входных векторов, элементы которого будем обозначать как . Кроме того, одноразмерное пространство – пространство выходов. Его элементы обозначим как . Предположим, что существует некоторое преобразование не известное в общем виде. Пусть есть конечный набор входных векторов , для которых известны значения функции . Иначе говоря, справедливо выражение.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Назовем набор пар обучающей выборкой, наблюдением из этого набора, а – соответствующее ему значение.

Будем искать приближение функции . Для построения такого приближения используется обучающая выборка . С учетом этого, справедливо представление

|  |  |  |
| --- | --- | --- |
|  |  |  |

будем называть моделью машинного обучения, построенной на обучающей выборке .

Для оценки точности приближения используются разные метрики ошибок. Зачастую, для дополнительной оценки точности моделей из обучающей выборки извлекают тестовый набор.

Существует две разновидности данной задачи – классификация и регрессия.

Задача регрессии подразумевает пространство , а следовательно, и возможные ответы всей системы вещественно-значными. Для таких задач точность множества некоторого множества выражается как сумма квадратов разностей между полученными моделью и заданными

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

где

– наблюдение из ;

– соответствующее ему значение;

– элементарная ошибка для множества

Тогда, для всего множества ошибка примет вид

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

Считается, что для модели чем меньше ошибка на обучающей выборке, тем выше ее обобщающая способность. Соответственно, для нее решается задача минимизации по функции .

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

Стоит также отметить, что метрика ошибки (1), (2) не является единственной, но принцип (3) – общий для всех задач регрессии и классификации.

Если векторы пространство представляет из себя конечное множество заранее известных значений, то говорят, что решается задача классификации.

В этом случае метрика точности выражает вероятность модели не ошибиться при определении выходного значения на множестве . Если в этом множестве , то точность может быть записана как может быть записана как

|  |  |  |
| --- | --- | --- |
|  |  |  |

где

– количество элементов для которых .

Соответственно, вероятность ошибки для наблюдения из множества представима следующим образом

|  |  |  |
| --- | --- | --- |
|  |  |  |

Обычно, когда говорят о точности моделей подразумевают либо точность на обучающей выборке, либо точность на тестовой выборке, либо на их объединении.

Зачастую, аналогом функции ошибок выступает функция потерь. Ее записывают как , либо как ,.

# 1.2 Ансамбли машинного обучения

Одним из наиболее проблемных вопросов в использовании машинного обучения для классификации, является вопрос переобучения и недообучения моделей. Существует такое явление, когда модель обучена настолько, что не может классифицировать наблюдения, не учувствовавшие в обучении. Иначе говоря, в процессе длительного обучения (например, при большом количестве эпох обучения, если речь идет о нейронной сети) модель теряет обобщающую способность. В таких случаях говорят, что модель переобучена. С обратной стороны, если не обладает обобщающей способностью даже на обучающем наборе данных, а соответственно и на тестовом, то говорят, что модель недообучена.

Одним из способов решения обоих проблем является ансамблирование методов машинного обучения.

Ансамблирование методов машинного обучения это способ организации нескольких моделей машинного обучения в мета-модель с целью повышения точности итоговой классификации или регрессии. Элементы мета-модели называют простыми, или слабыми моделями, а мета-модель – ансамблем

С точки зрения, указанных ранее вопросов, ансамбли используют заранее определенное множество слабых, или недообученных моделей. Результат, должен улучшаться за счет консолидации результатов всех моделей. Предполагается, что модели в общем являются независимыми друг от друга. А следовательно, независимыми будут и их оценки. Из этого следует, что вероятности ошибок на оценках также будут независимыми, а следовательно, если используется моделей, справедлива формула

|  |  |  |
| --- | --- | --- |
|  | , |  |

где  
 – общая вероятность ошибки наблюдения ;

– входное наблюдение;

– вероятность ошибки наблюдения на модели.

По однородности простых моделей, ансамбли делят на:

– гомогенные;

– гетерогенные.

Гомогенные ансамбли состоят из моделей одинаковой структуры. В таких случаях независимость слабых моделей достигается за счет разбиения обучающей выборки на подмножества определенного размера причем, в рамках таких подмножеств одно и тоже наблюдение может повторяться. Для гетерогенных моделей независимость достигается за счет разных слабых моделей, в частности если модели используют разные методы машинного обучения.

По типу построения ансамбли разделяют на:

– бэггинг;

– стэкинг;

– бустинг.

# 1.2.1 Бэггинг

Бэггинг (англ. bagging – bootstrap aggerating) подразумевает использование гомогенных слабых моделей.

Предположим, что используется обучающий набор для которого задана простая модель . Допустим, что представлен набор обучающих подвыборок , каждая из которых содержит независимых наблюдений. Эти наблюдения используются, чтобы улучшить предсказательную способность модели . Это делается посредством создания набора . Не теряя общности, будем считать, что таких наборов .

Если решается задача регрессии, то заменяется средним

|  |  |  |
| --- | --- | --- |
|  |  |  |

В случае, если решается задача классификации, то итоговый берется т. н. методом голосования, то есть берется мода от полученного набора результатов. Схематичное изображение данного метода представлено на рисунке 1.

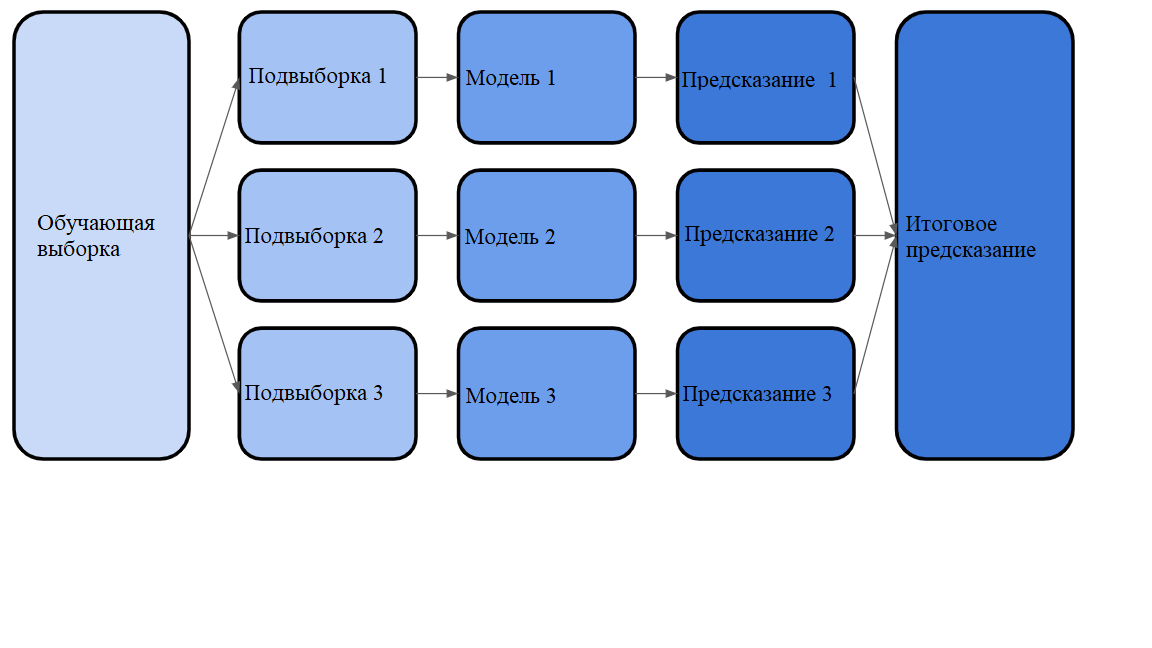


Рисунок 1– Схематичное изображение системы, использующей бэггинг

Подобный метод обработки входного набора в англоязычной получил наименование bootstrap aggregating, или сокращенно bagging [1].

Следует отметить, что распределение результатов для наборов должно соответствовать таковому для набора . Кроме, того наблюдения в наборы могут попадать с повторениями. Возможный вариант задания таких наборов представлен на рисунке 2.



Рисунок 2 – Возможный вриант распределения подвыборок

# 1.2.1.1 Математическое обоснование бэггинга

Пусть – простая модель, действующая на классах . Обозначим

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

Интерпретация представления (4) следующая: по всем независимым поднаборам наблюдений , предсказывает класс со входным вектором с относительной частотой . Иначе говоря, – это апостериорная вероятность предсказания . Пусть – вероятность того, что входной вектор должен вернуть значение , то есть – это априорная вероятность выхода . Тогда вероятность того, что при заданном , итоговое значение будет предсказано корректно выглядит следующим образом

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

Из представления (5) получим вероятность правильности итоговой классификации

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

где  
 – плотность распределения .

Заметим, что для (5) справедливо неравенство

|  |  |  |
| --- | --- | --- |
|  |  |  |

причем, равенство достигается только тогда, когда справедливо условие

|  |  |  |
| --- | --- | --- |
|  |  |  |

Предиктор (Байесовский классификатор) может быть подставлен в выражение (6), для получения наиболее высокой вероятности успешной классификации

|  |  |  |
| --- | --- | --- |
|  |  |  |

Назовем модель корректной на входном векторе , если выполняется условие

|  |  |  |
| --- | --- | --- |
|  |  |  |

Иначе говоря, если вход приводит к выходу гораздо чаще, чем к другим выходам, то и предсказывает на чаще, чем другие классы. Отметим, что корректная модель не всегда может быть точной. Рассмотрим задачу классификации с двумя классами и . Пусть , и . Очевидно, что является корректным классификатором. Тогда вероятность правильной классификации, согласно формуле (5), составит .

Рассмотрим агрегированную модель . Для него вероятность правильной классификации примет вид

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

где  
 – индикаторная функция.

Если корректна на , то выражение (7) принимает значение . Пусть это множество таких , на которых является корректной моделью, тогда итоговая вероятность правильной классификации для примет вид

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |

При этом, даже если корректна на , ее классификация может быть далека от оптимальной, но для она будет хорошей. Если модель корректна на большинстве вводимых , то ее агрегация приведет его к очень хорошим результатам [1].

# 1.2.2 Стэкинг

Стэкинг (англ. stacking – stacked generalizing) разновидность методов ассемблирования, работающая на гетерогенных простых моделях.

В рамках стэкинга, для определения итоговой классификации используется выходная модель, принимающая, выходы простых моделей. Таким образом, обучение такой мета модели состоит из двух этапов – обучения уровня 0 – простых моделей и обучения уровня 1 – выходной модели. В принципе, уровней с простыми моделями может быть и больше, однако, чаще всего используются 2 указанных уровня. Общая схема работы представлена на рисунке 3.

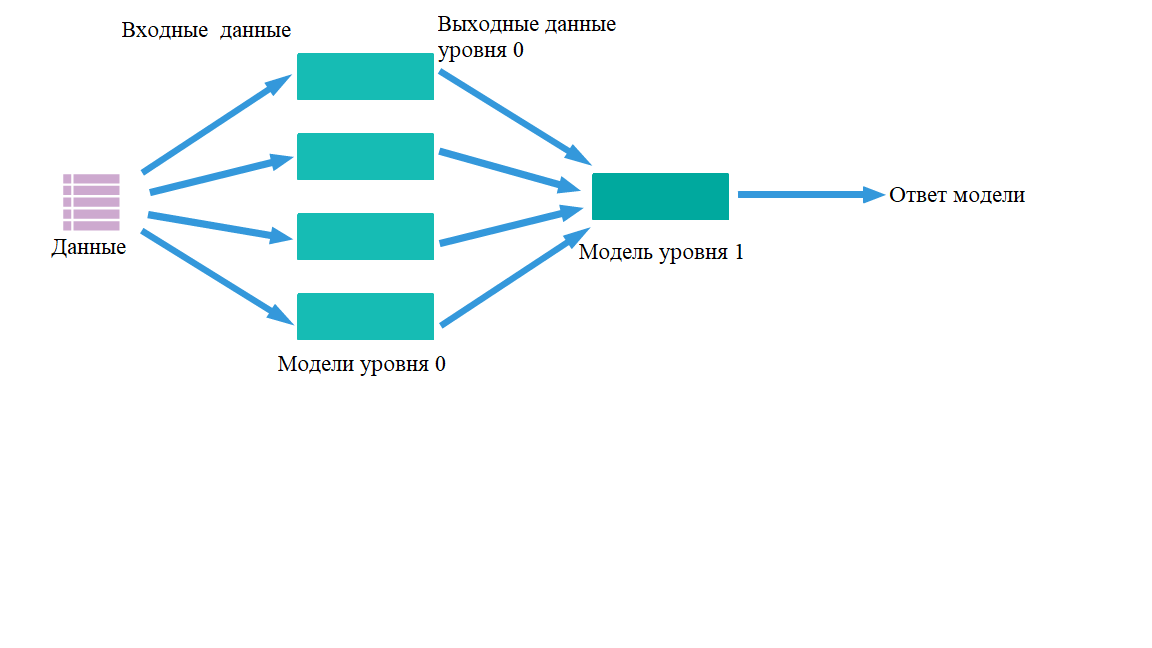


Рисунок 3 – Общая схема работы стэкинга

Существует два варианта организации стэкинга – блэндинг (англ. blending) и классический стэкинг.

Для блэндинга обучающая выборка делится на две части – одну для уровня-0 и одну для уровня-1. В этом случае, выборку для уровня-1 также еще называют валидационной. После обучения уровня-0 данные из валидационной выборки обрабатываются на нем. Полученные ответы уровня-0 становятся входными данными для уровня-1, который обучается на значениях и соответствующих классификациях валидационной выборки.

Алгоритм блэндинга можно представить следующим образом:

1) Вычленение из обучающей выборки валидационной выборки;

2) Обучение уровня-0;

3) Вычисление значений для валидационной выборки на уровня-0;

4) Использование результатов, полученных на третьем этапе для обучения уровня 1;

5) Строится итоговая модель, в которой выходы подаются на уровень-0, а выходом является ответ уровня-1.

Классический стэкинг отличается от блэндинга механизмом разбиения обучающей выборки. Здесь не происходит четкого вычленения валидационной выборки. Вместо этого используется k-кратная кросс валидация. Обучающая выборка делится на k наборов или фолдов. В таком виде обучаются все фолды за исключением последнего, который имеет ту же функцию, что и валидацнионный набор для блэндинга. Еще одним важным отличием является то, что базовые модели обучаются на целом наборе, а затем, предсказывают результат для тестовой выборки и его передают на уровень-1.

Таким образом, алгоритм обучения при классическом стэкинге можно представить следующим способом:

1) Общая выборка делится на обучающую и тестовую;

2) Обучающая выборка разбивается на фолды;

3) Простая модель обучается на k-1 фолдах, для k фолда рассчитывается выход;

4) Процесс повторяется до тех пор, пока каждый фолд таким образом не будет рассчитан;

5) Простая модель обучается на всей тренировочной выборке, а затем вычисляется ее данные для тестовой выборки;

6) Шаги 3–5 повторяются для каждой простой модели;

7) Предсказания для тестовой выборки, полученные ранее используются в качестве входов уровня-1;

8) Строится итоговая модель, в которой выходы подаются на уровень-0, а выходом является ответ уровня-1.

# 1.2.2.1 Математическое обоснование стэкинга

Представим обучающий набор , как совокупность пар и

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

где

– размерность пространства входных векторов;

– объем выборки.

Следует отметить, что в данном контексте задача рассматривается регрессии как более общая, поскольку конечный набор входных данных для классификации является подмножеством для .

Генерализатор это разновидность простой модели, которая отражает пространство входов на пространство выходов . Для любого заданного существует счетно-бесконечное множество функций . При этом, каждому возможному соответствует свой . Так для принимает три аргумента. Здесь и – обучающая выборка размерности , – это так называемый вопрос, входной вектор пространства на основе которого требуется получить ответ . Тогда, будет иметь на входе пять аргументов , – семь и так далее.

Часто, набор не задается напрямую, например в случае использования алгоритма обратного распространения. В иных случаях, когда для генрализаторов задается поверхность решений, существует возможность записи напрямую. Будем говорить, что генерализатор с аргументами , обучен на -элементном наборе и ему задают вопрос , на основании чего должен быть получен ответ . В случае, если вопрос равен одному из , а ответ равен соответствующему для любого , иначе говоря, если справедливо утверждение

|  |  |  |
| --- | --- | --- |
|  |  |  |

то говорят, что генерализатор реплицирует обучающую выборку (8).

С учетом, сказанного справедлива запись

|  |  |  |
| --- | --- | --- |
|  |  |  |

то есть вполне можно говорить о генерализаторе , как простой модели.

Обозначим , как множество входных векторов с присоединенными к ним векторам ответов . Не теряя общности, будем считать, целевой вектор – однокомпонентный. Если это не так, то исходная задача разбивается на подзадачи, таким образом, чтобы размерность целевого вектора была 1. Тогда вектора множества будут принадлежать множеству .

Сперва определим разбиений множества , таких, что делят его на два не пересекающихся набора. Обозначим такие разбиения как , где , . Назовем их набором разбиений. К таким разбиениям, например, можно относить m-кратную кросс-валидацию, такой набор разбиений называется кросс-валидационным набором разбиений, или КВНР. Для простоты будем считать, что , тогда для любого , содержит единственный элемент из , а – все остальные. При этом, , а, следовательно, совокупность всех полностью покрывает .

КВНР используется для задания отношения генреализатора на обучающем наборе и оценки ошибки на нем. Здесь эта оценка рассчитывается как ошибка на тестовом наборе . Тогда ошибка на всем КВНР может быть получена как усредненное значение по всем . Получим функционал ошибки

|  |  |  |
| --- | --- | --- |
|  |  |  |

где

– входные компоненты ;

– выходной компонент .

Назовем пространство на котором определено множество пространством уровня 0. Все генерализаторы, обученные на в этом пространстве, назовем простыми моделями уровня 0, а само множество обучающей выборкой уровня 0.

Для каждого из разбиения будем рассматривать ответов простых моделей уровня 0 , обученных на на полученный набор ответов уровня 0, совместно с определенным выходным компонентом образуют пространство . Такое пространством назовем пространством уровня 1. Полученные точек в этом пространстве редуцированной обучающей выборкой, или обучающей выборкой уровня 1.

На основе вышесказанного ставится задача получения модели, реплицирующей на основе уровня 1. Что бы это сделать, обучающую выборку пропускают через модели уровня 0 и получают обучающую выборку пространства уровня 1. На основе полученных данных обучается простые модель уровня 1. Обычно, такая модель одна и ее ответ является ответом всей совокупности, называемой мета-моделью. Такой способ организации получил название stacking generalization, или сокращенно – stacking [2].

# 1.2.3 Бустинг

Методы бустинга относятся к гомогенным ансамблевым моделям. Отличие данного класса методов от представленных выше заключается в том, что модели в нем обучаются последовательно, причем для каждого последующего метода в цепочке дополнительно используются данные предыдущего. В этом смысле он отличается от стэкинга тем, что результаты простых моделей не заменяют собой входные данные, а дополняют их.

Под бустингом подразумевается два вида алгоритмов ансамблирования:

– Адабуст;

– Градиентный бустинг.

# 1.2.3.1 Адабуст

Адабуст (англ. AdaBoost – Adaptive Boosting) стал первым практически реализованным алгоритмом бустинга. Он используется для решения задач классификации. Схематично работа метода представлена на рисунке 4.

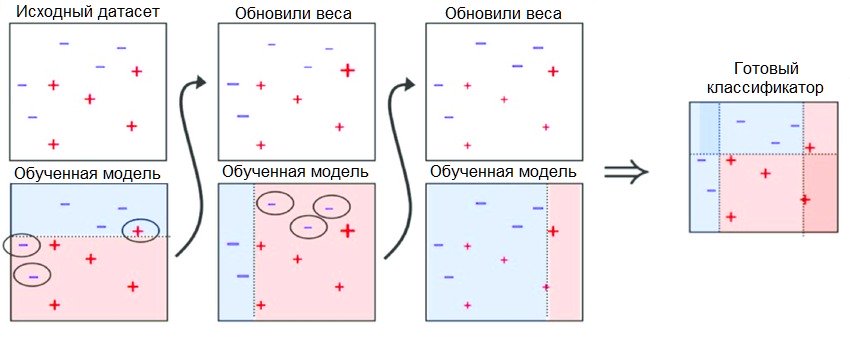


Рисунок 4 – Схематичное изображение работы метода адабуст

Пусть имеется обучающая выборка . Будем рассматривать задачу бинарной классификации такую, что . Множественная классификация будет рассмотрена отдельно. Алгоритм вызывает простую модель циклически на протяжении обходов. Идея метода заключается в добавлении к элементам обучающей выборки весовых коэффициентов, причем, эти коэффициенты должны образовывать функцию распределения на обучающей выборки. Обозначаем – вес -го наблюдения обучающей выборки на -ом обходе. Для этой функции должны выполняться условия

|  |  |  |
| --- | --- | --- |
|  |  | (9) |
|  |  |

Изначально, веса задаются одинаково для всех наблюдений, но в процессе для ошибочно классифицированных наблюдений, веса увеличиваются, таким образом, чтобы модель фокусировалась на тех пример, которые были определены неправильно.

Цель слабой модели на каждом обходе состоит в том, чтобы определить слабую гипотезу , основываясь на текущей функции . Ошибку в этом случае можно представить как

|  |  |  |
| --- | --- | --- |
|  |  | (10) |

Заметим, что вероятность ошибки здесь идет с поправкой на распределение .

Возможны два случая. Если используемая простая модель поддерживает использование взвешенных входных данных (например, метод опорных векторов, или деревья решений), тогда вектору назначаются веса, и он непосредственно участвует в обходах. В противном случае, на основании собирается новая подвыборка из тех наблюдений, на которых были ошибки, и уже она используется для обучения на следующих обходах.

После построения гипотезы , для нее определяется параметр

|  |  |  |
| --- | --- | --- |
|  |  |  |

На основе этого параметра строится распределение для следующего обхода

|  |  |  |
| --- | --- | --- |
|  |  |  |

где

– параметр, подбираемый таким образом, чтобы соответствовало условиям (9), (10).

Заметим, что в случае исключения наблюдения из рассмотрения, .

Итоговая классификация модели определяется посредством взвешенного голосования [3]

|  |  |  |
| --- | --- | --- |
|  |  |  |

Обозначим . Положим, что абсолютно случайное предсказание имеет вероятность ошибки 0.5. Тогда показывает, насколько гипотеза лучше, чем случайный ответ. Для оценки ошибки общей гипотезы используется выражение [3]

|  |  |  |
| --- | --- | --- |
|  |  | (11) |

Из выражения (11) следует, что если каждая последующая гипотеза чуть лучше случайного выбора (иначе говоря, если ), то ошибка уменьшается экспоненциально быстро.

Существуют разные методы адаптации алгоритма Адабуст к многоклассовым задачам. Одним таким способом является разбиение изначальной задачи на набор бинарных классификаций, решающих принадлежности наблюдению к одному классу отдельно.

# 1.2.3.2 Градиентный бустинг

Метод градиентного бустинга является продолжением развития адабуста. Однако, между методами есть существенное отличие. Оно заключается в том, что для усиления модели на ошибках адабуст использует веса, а градиентный бустинг – метод наискорейшего спуска. Кроме того, этот метод ансамблирования отличен тем, что способен использовать только деревья решений в качестве простой модели.

Как и в случае адабуста рассматривается бинарная классификация . Пусть дана обучающая выборка . Решается задача минимизации (3) . Будем искать такое приближение функции , которое минимизирует функцию потерь на пространстве :

|  |  |  |
| --- | --- | --- |
|  |  | (12) |

В качестве функции потерь для задачи классификации предполагается

|  |  |  |
| --- | --- | --- |
|  |  | (13) |

Главная проблема, возникающая при решении задачи (12) – то, что пространство функций, в котором ведется поиск – бесконечномерное. Способом решения этой проблемы является ограничение таким образом, чтобы она была частью класса параметризованных функций , где конечный набор параметров, позволяющий однозначно определить .

|  |  |  |
| --- | --- | --- |
|  |  | (14) |

Функция – простая функция с входным вектором и набором параметров . В частности, в качестве такой функции могут выступать простые модели машинного обучения, например деревья классификации и регрессии. [4]

Использование в качестве итоговой модели позволяет заменить задачу функциональной оптимизации, задачей параметрической оптимизации:

|  |  |  |
| --- | --- | --- |
|  |  | (15) |
|  |  | (16) |

С учетом (15) и (16) представляется в виде:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Для решения задачи (15), (16) используются численные методы оптимизации. Это подразумевает представление решения в форме

|  |  |  |
| --- | --- | --- |
|  |  | (17) |

где

– некое начальное приближение;

– последовательные улучшения (шаги, или бусты).

Способы получения каждого шага зависят от метода оптимизации.

Рассмотрим применение алгоритма наискорейшего спуска для решения данной задачи. В этом случае способ получения набора улучшений следующий. На каждом шаге вычисляется текущий градиент :

|  |  |  |
| --- | --- | --- |
|  |  | (18) |
|  |  | (19) |

Тогда, шаг определяется как:

|  |  |  |
| --- | --- | --- |
|  |  | (20) |
|  |  | (21) |

Обратный градиент также называется направлением спуска, а значение – длиной.

Подобный подход можно адаптировать для непараметрических функций. В этом случае будем считать, что сама функция на каждом входном векторе является параметром. Тогда минимизируемая функция (16) примет вид:

|  |  |  |
| --- | --- | --- |
|  |  |  |

В пространстве функции есть бесконечное множество таких параметров, однако поскольку обучающая выборка – конечное множество, то и предоставить она может конечное множество векторов . Это значит, что может быть рассмотрен только конечный набор . Решение (16), тогда представимо в виде:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Данное представление аналогично представлению (17) с той разницей, что . Наискорейший спуск (18), (19), (20), (21) в этом случае записывается как:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  | (22) |
|  |  | (23) |

В случае интегрируемости интегрирование в (21) может быть вынесено за знак дифференцирования:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Поскольку для и известен только конечный набор данных , то интеграл не может быть посчитан на тех , которые не входят в . Одним из вариантов обхода данного ограничения является выражение искомой функции в параметризированной форме (14) и проведение параметрической оптимизации, приведённой ранее:

|  |  |  |
| --- | --- | --- |
|  |  |  |

В случаях, когда это невозможно, применяют подход, названый бустингом:

|  |  |  |
| --- | --- | --- |
|  |  | (24) |
|  |  | (25) |

Пусть для некоторой функции потерь и/или простой модели решения для (24) получить тяжело. Для заданной , функция в (25) может считаться наилучшим шагом для получения приближения , при условии, что направление шага является членом класса параметризуемых функций . В этом случае, можно говорить об , как о направлении спуска. В данном случае, (23) примет вид:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

Однако, обратный градиент известен только на наборе входов и не может быть распространен на все . Способом решения данной проблемы является выбор такого члена класса параметризованных функций , который даст вектор , максимально параллельный направлению . Его можно получить, решая

|  |  |  |
| --- | --- | --- |
|  |  |  |

Получаемый таким образом замещает градиент как направление спуска [4]. В этом случае (22), (23) принимает вид:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

# 2 Постановка задачи и создание входного набора данных

При экономическом взаимодействии в банковской сфере одну из важнейших ролей занимают финансовые показатели подобных организаций. Именно эти показатели во время выбора предложений от кредитных организаций являются решающим фактором, если такие предложения идентичны. Кроме того, они необходимы в оценке целесообразности выдвигаемых предложений.

Одним из способов упрощения получения оценки деятельности финансового предприятия является присвоение ему группы, или класса на основе финансовых показателей.

В данной главе представлен одни из возможных способов разбиения банков на группы на основе их показателей. Кроме того, будут построены обучающая и тестовая выборки для создания моделей классификации на основе методов, представленных в главе 1 данной работы.

# 2.1 Начальные данные и первичная обработка

В качестве основы для выборок используются данные по финансовым показателям банков, представленные на портале banki.ru [5]. Условно, эти показатели можно разделить на две категории: показатели, выраженные в денежном эквиваленте и процентные данные.

Всего денежных показателей 5:

– Капитализация по форме 123 – итоговые данные расчета формы 0409123 "Расчет собственных средств (капитала) (Базель III)", выражающие собственный капитал кредитной организации;

– Активы-нетто – совокупность активов банка, включающая в себя денежные средства на счетах в кассе кредитной организации, средства на корреспондентский сетах в других организациях, а также выданные межбанковские кредиты, в том числе и в ЦБ;

– Вклады физических лиц – совокупность привлеченных депозитов физических лиц;

– Вложения в ценные бумаги – объем инвестиций кредитной организации в различные ценные бумаги по типу акций, облигаций, ваучеров, векселей и т. п.;

– Кредитный портфель – объем выданных кредитной организации заемных средств как физическим, так и юридическим лицам.

Все такие данные дополняются процентными изменениями за предыдущий год. Это делается для отражения динамики показателей в модели.

Для всех денежных показателей также берутся данные из Банка России [6] по общему объему рынка. Эти данные представлены в таблице А.1 приложения А. Для построения моделей используются не денежные выражения, а их отношения к объему рынка

|  |  |  |
| --- | --- | --- |
|  |  |  |

где

– используемое значение показателя;

– значение показателя, полученное с сайта banki.ru;

– объем рынка по данным центрального банка.

Следует отметить, что данные показателей полученные на banki.ru приводятся в тысячах рублей, а объемы рынка – в миллиардах, поэтому для получения объем рынка следует умножать на 1000000.

Данное преобразование позволяет изменить семантику данных. Если ранее показатель представлял объем денежных средств, выраженный в рублях того периода, для которого он брался, то теперь он представляет собой долю банка в рынке по данному показателю. Это позволяет унифицировать финансовые данные разных лет, что в свою очередь дает возможность классифицировать наблюдения будущих периодов.

В рамках категории процентных данных используется 4 показателя, непосредственно получаемых с сайта:

– Уровень просроченной задолженности в кредитном портфеле – процентное отношение объема просроченных обязательств заемщиков к кредитному портфелю;

– Уровень резервирования в кредитном портфеле – процентное отношение объема средств, выделяемых под резервы для кредитов к объему кредитного портфеля;

– Рентабельность капитала – отношение чистой прибыли финансовой организации к ее капитализации по форме 123;

– Рентабельность активов-нетто – отношение чистой прибыли к сумме активов-нетто для данной организации.

Сюда же входят и изменения денежных показателей, представленных выше:

– Изменение капитализации;

– Изменение активов-нетто;

– Изменение вкладов физических лиц;

– Изменение вложений в ценные бумаги;

– Изменение кредитного портфеля.

Данные показатели переводятся из процентных в относительные делением на 100.

К сожалению, не для всех банков имеется полный набор приведенных показателей. Если для какого, то организации набор не полный, то такая организация выключается из рассмотрения.

# 2.2 Использование банка консолидированного лидера

Для расширения данных будет использоваться метрический подход [7]. Суть подхода заключается в представлении наблюдений как точек в многомерном пространстве. На их основе определяется так называемы банк консолидированный лидер (БКЛ), показатели с наилучшими из представленных показателей по всей выборке. В рамках этой методики предполагается, что расстояние от БКЛ, в пространстве на основе которого он строился, выражает различие между банком-наблюдением и БКЛ. Иначе говоря, чем выше это расстояние, тем хуже идут дела у банка.

Перед построением БКЛ вводятся еще 5 показателей на основе изменений:

– Относительное изменение капитализации;

– Относительное изменение активов-нетто;

– Относительное изменение вкладов физических лиц;

– Относительное изменение вложений в ценные бумаги;

– Относительное изменение кредитного портфеля.

Они представляют из себя произведение соответствующих показателей изменений и долей рынков. Их использование обусловлено эффектом низкой базы, когда для банков с относительно малыми показателями, изменения этих показателей имеют очень большие значения. Это приводило к слишком большим расстояниям для всех банков. Полученные же показатели отвечают изменению доли на рынке для данного банка. Таким образом, в определении БКЛ и определении расстояния до него изначальные изменения не используются.

Для всех показателей, кроме уровня просроченной задолженности в кредитном портфеле и уровня резервирования в кредитном портфеле используется принцип, что чем больше, тем лучше. Для двух отдельных показателей считается, что чем меньше, чем лучше.

С учетом вышеперечисленного определяется БКЛ. В таблице 1 представлены его характеристики.

Таблица 1 – Показатели БКЛ

|  |  |
| --- | --- |
| Капитализация | 0,4093 |
| Активы-нетто | 0,3211 |
| Вклады физических лиц | 0,3211 |
| Вложения в ценные бумаги | 0,3514 |
| Кредитный портфель | 0,359 |
| Уровень просроченной задолженности | 0 |
| Уровень резервирования | 0 |
| Рентабельность капитала | 0,675 |
| Рентабельность активов-нетто | 0,3514 |
| Отн. изм. капитализации | 0,0359 |
| Отн. изм. активов-нетто | 0,0351 |
| Отн. изм. вкладов ФЛ | 0,0145 |
| Отн. изм. вложений в ценные бумаги | 0,1055 |
| Отн. изм. кредитного портфеля | 0,0634 |

В качестве метрики расстояния в строимой модели используется евклидово расстояние

|  |  |  |
| --- | --- | --- |
|  |  |  |

где

– показатель банка, для которого оценивается расстояние;

*–* соответствующий ему показатель БКЛ.

Таким образом, получается финальный показатель – расстояние от БКЛ.

# 2.3 Неразмеченные данные

С учетом приведенных выше преобразований входное наблюдение состоит из 20 показателей:

– Капитализация по форме 123;

– Активы-нетто;

– Вклады физических лиц;

– Вложения в ценные бумаги;

– Кредитный портфель;

– Уровень резервирования в кредитном портфеле;

– Уровень просроченной задолженности в кредитном портфеле;

– Рентабельность капитала;

– Рентабельность активов-нетто;

– Относительное изменение капитализации;

– Относительное изменение активов-нетто;

– Относительное изменение вкладов физических лиц;

– Относительное изменение вложений в ценные бумаги;

– Относительное изменение кредитного портфеля;

– Расстояние от БКЛ.

Приведенным выше способом были получено два набора данных. Первый – данные по состоянию на начало 2022 года, изменения при этом берутся с начала 2021 года. Второй, соответственно данные на начало 2021 года с изменениями за период 2020 года.

Первый набор содержит 329 наблюдений и выступит в качестве обучающей выборки. Фрагмент этой выборки представлен на рисунке 5

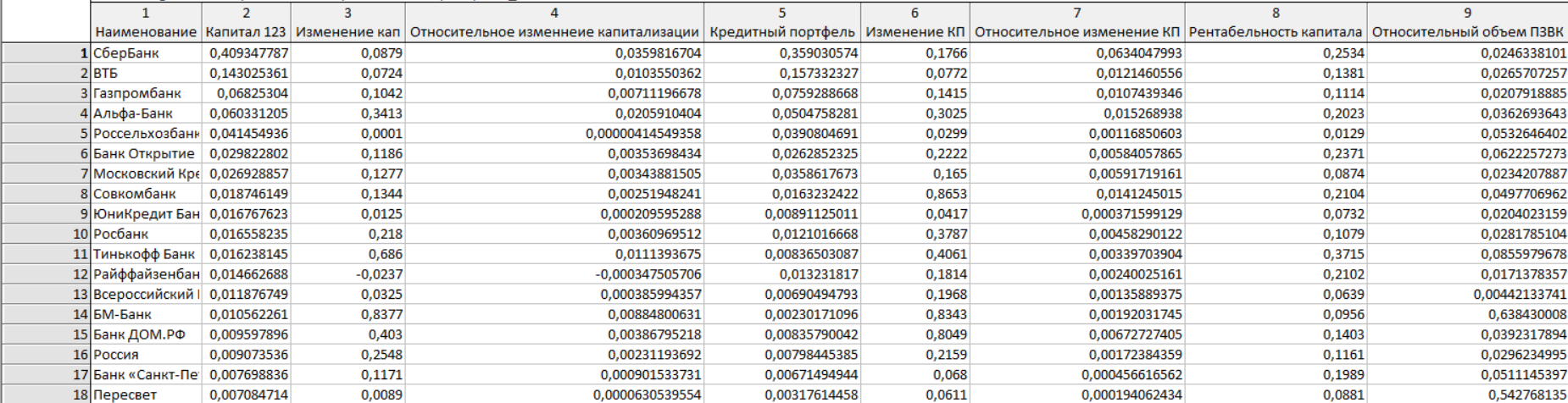


Рисунок 5 – Фрамгмент обучающей выбокри

Вторая выборка состоит из 379 наблюдений и станет тестовой.

В таблице А.2 приложения А представлены статистические данные по группам. Из нее видно, что в целом для разных лет средние и стандартные отклонения не сильно различаются.

Для дальнейшего доказательства однородности полученных данных воспользуемся t-критерием для независимых выборок. Анализ будет проводиться по всем показателям, кроме расстояния от БКЛ, так как тот сочетает в себе все остальные. Результаты анализа представлены на рисунке 6.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 6 – T-критерий для независимых выборок

Из таблицы, представленной на рисунке 6 видно, что единственным показателем, для которого разница в средних между группами статистически значима, является уровень просроченной задолженности в кредитном портфеле. В целом, несмотря на этот показатель, между годами сохраняется однородность.

# 2.4 Кластеризация

Рассматриваемые методы ансамблирования машинного обучения требуют, чтобы входные данные были размечены. Для такой цели хорошо подходит методика кластеризации.

Кластеризация – разновидность машинного обучения без учителя, при котором неразмеченные данные распределяются по группам однородности, называемым кластерами. Предполагается, что банки внутри одного кластера обладают сходными экономическим характеристиками.

Для решения задачи кластеризации будет использоваться надстройка «Обобщённые методы кластерного анализа» модуля «Добыча данных» пакета прикладных программ STATISTICA. Выбор средства несколькими факторами. Во-первых, оно предоставляет возможность быстрой загрузки и выгрузки данных в формате Excel. Во-вторых, пакет предоставляет средства быстрой оценки и настройки строящихся моделей.

Для кластеризации будет использован объединенный набор данных, состоящий из обучающей и тестовой выборки. Следует отметить, что в полученном наборе есть четыре наблюдения, которые за счет величины своих показателей находятся достаточно далеко от других банков, чтобы образовывать свои кластеры. Это банки «Сбербанк» и «ВТБ» в обеих выборках (в наборе это наблюдения 1, 2, 330 и 332). В последствии они будут приписаны к наиболее «успешному» кластеру.

В качестве входных переменных для кластеризации было выбрано 10 показателей:

– Капитализация по форме 123;

– Активы-нетто;

– Вклады физических лиц;

– Вложения в ценные бумаги;

– Кредитный портфель;

– Уровень резервирования в кредитном портфеле;

– Уровень просроченной задолженности в кредитном портфеле;

– Рентабельность капитала;

– Рентабельность активов-нетто;

– Расстояние от БКЛ.

На рисунках 7 и 8 представлены настройки модели кластеризации.

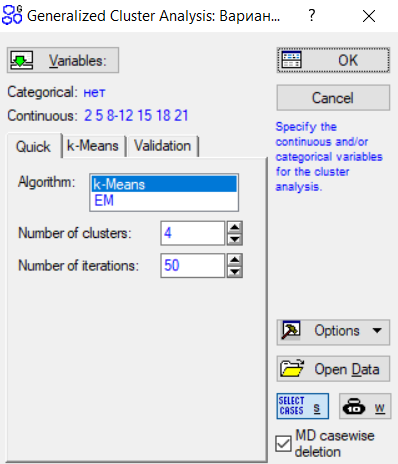


Рисунок 7 – Вкладка «Quick» модуля кластеризации

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8 – Вкладка «k-means» модуля кластеризации

Из рисунков видно, что в качестве алгоритма кластеризации используется метод -средних.

Этот метод подразумевает разделение выборки на кластеров. Итеративно для каждого кластера рассчитываются центры, а затем наблюдения перераспределяются к ближайшим к ним кластерам, согласно метрикам расстояния. В данном случае используется нормализованная метрика Чебышева

|  |  |  |
| --- | --- | --- |
|  |  |  |

Было произведено разбиение на 4 кластера. Их характеристики приведены на рисунке 9.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 9 – Характерисктики кластеров

На рисунке видно, что согласно расстоянию от БКЛ наиболее успешным является кластер номер 1. К нему и будут приписаны исключённые ранее «Сбербанк» и «ВТБ». Следом, идут кластеры 2, 4 и 3 соответственно.

К сожалению, далеко не всегда исследуемые объекты могут образовывать структуры однородности. В таких случаях нельзя говорить о корректности кластеризации. Для оценки качества полученного разбиения используется дисперсионный анализ. На рисунке 10 представлены результаты дисперсионного анализа для полученного ранее кластерного разбиения.

Изображение выглядит как текст, стол

Автоматически созданное описание

Рисунок 10 – Результаты дисперсионного анализа

По значениям внутригрупповой (Between SS) и межгрупповой (Within SS) можно судить о вкладах показателей в кластеризацию. Так, чем меньше внутригрупповая и чем больше межгрупповая дисперсии, тем больше вклад. Из представленного рисунка можно сделать вывод, что наибольший вклад в кластеризацию оказывают показатели капитализации и вкладов ФЛ. В это же время наименьший вклад вносят рентабельность капитала и рентабельность активов-нетто. Кроме того, о влиянии также говорит критерий Фишера (F) и уровень статистической значимости (p). Для построенной кластеризации все показатели, то есть имеют , а значит их вклад в модель статистически значим. Таким образом, можно говорить, что для построенная модель с 4 группами однородности имеют математически обоснованную кластерную структуру.

На рисунке 11 представлены расстояния между кластерами.

Изображение выглядит как текст, внутренний

Автоматически созданное описание

Рисунок 11 – Расстояния между кластерами

Данные на рисунке 11 отвечают на различия между кластерами. Из рисунка видно, что наибольшие расстояния были получен между кластерами 1 и 3 и 1 и 4, а наименьшее – между кластерами 2 и 4.

Чтобы обеспечить отражение экономической оценки в нумерации кластеров следует изменить их названия. Кластер 3 будет обозначен как кластер 4, а кластер 4 – как кластер 3. Таким образом получим обучающую и тестовую выборки, представленные в таблице 2

Таблица 2 – Состав обучающей и тестовой выборок

|  |  |  |  |
| --- | --- | --- | --- |
| Кластер | Обучающая | Тестовая | Всего |
| 1 | 5 | 5 | 10 |
| 2 | 13 | 13 | 26 |
| 3 | 282 | 329 | 611 |
| 4 | 29 | 32 | 61 |
| Всего | 329 | 379 | 708 |

Для наглядности кластер 1 будет обозначаться как «Высоко-успешный», кластер 2 – как «Успешный», кластер 3 – как «Основной», кластер 4 – как «Мусорный».

# 3 Построение моделей

Полученные ранее выборки будут использованы для обучения и анализа четырех моделей ансамблирования, рассмотренных в главе 1.

3.1 Scikit-learn

В качестве основанного инструмента для создания моделей был выбран фреймворк Scikit-learn версии 1.1.1 для скриптового языка программирования Python версии 3.10. Выбор обусловлен простотой разработки языка Python, реализацией указанных методов ансамблирования в библиотеках Scikit-learn, а также простых моделей для них. Кроме того, в языке Python есть удобные средства интеграции пользовательского интерфейса.

Scikit-learn – простое средство предиктивного анализа данных, построенное на библиотеках NumPy, SciPy и matplotlib. Одним из главных достоинств данного инструмента, с точки зрения разработчика, является модель распространения Open Source, а также BSD лицензирование, что позволяет свободно использовать фреймворк в некоммерческих целях.

Scikit-learn используется в таких проектах, как Spotify, Télécom ParisTech, Booking.com, Change.org. В частности, инструментарий фреймворка используется таким крупным банком, как J.P. Morgan, где широко используется в структурах банка для решения задач классификации, предиктивной аналитик и прочих вопросов, связанных с машинным обучением. [8]

Приведенные выше доводы позволяют считать фреймворк Scikit-learn подходящим средством, для решения поставленной задачи.

# 3.1.1 Реализация методов ансамблирования в Scikit-learn

В рамках Scikit-learn реализованы следующие методы ансамблирования: бэггинг, стэкинг, адабуст, градиентный бустинг, голосование, случайный лес. Указанные методы могут решать как задачи классификации, так и задачи регрессии. Рассмотрим реализацию методов бэггинга, стэкинга, адабуста и градиентного бустинга применительно к задаче классификации. Для всех – определены методы fit и predict, отвечающие, соответственно, за обучение моделей и получение ответ от них.

Класс BaggingClassifier соответствует методу бэггинга, применительно к задаче классификации. Он может реализовывать как блэндинг, так и, собственно, бэггинг. При инициализации метод может принимать следующие параметры:

– base\_estimator – объект Scikit-learn, выражающий простую модель по умолчанию в качестве простой модели выступает дерево решений;

– n\_estimators – количество простых моделей в ансамбле. По умолчанию – 10;

– bootstrap – булево значение, отвечающее за тип модели. True – бэггинг, False – блэндинг. По умолчанию – True;

– random\_state – начальное генератора случайных чисел. Если не задано значение, то выбирается системное время.

Класс StackingClassifier реализует метод стэкинга. Отличительной чертой является то, что модель может иметь только двухуровневую структуру, причем на выходном уровне моет быть только одна простая модель. Основными параметрами, при инициализации являются:

– estimators – список моделей уровня-0. Не имеет значения по умолчанию, поэтому должен быть задан при инициализации;

– final\_estimator – выходная модель уровня-1. Как и для моделей уровня-0, должна быть задана при инициализации;

– cv – вид кросс-валидации. Может принимать целочисленное значение, либо быть объектом, который может быть создавать кросс-вариационные под выборки. При целочисленном значении используется n-кратная кросс-валидация, где n – значение параметра. По умолчанию оно равно 5.

– passthrough – «пропускание» входных данных. Если False – то для уровня-1 используются только выходные данные уровня-0, если True – то на для уровня-1 также используются входные данные уровня-0.

Класс AdaBoostClassifier реализует алгоритм адабуста. Главная особенность данного алгоритма в рамках фреймворка – ограничения по видам простых моделей. Здесь не поддерживаются те простые модели, для которых нет взвешивания входных наблюдений. В качестве таких моделей могут выступать деревья решений, наивный байесовский классификатор и метод опорных векторов. Для решения задачи используется алгоритм SAMME, или – SAMME.R. Данный класс при инициализации использует следующие параметры:

– base\_estimator – простая модель для строящегося ансамбля. По умолчанию используются деревья классификации с максимальной глубиной 1;

– n\_estimators – максимальное количество простых моделей, которым ограничивается ансамбль. По умолчанию равно 50;

– learning\_rate – значение веса, присваиваемое простым моделям на каждой итерации. Более высокие значения усиливают влияние отдельных простых моделей. По умолчанию равно 1;

– algorithm – вид алгоритма бустинга, которое используется в модели. Может быть одним из алгоритмов, представленных выше. По умолчанию используется SAMME.R;

– random\_state – начальное значение генератора случайных чисел. Как и в случае бэггинга, по умолчанию используется системное время.

Класс GradientBoostingClassifier является реализацией алгоритма градиентного бустинга. В отличии от адабуста здесь в качестве простой модели используется только деревья решений. Параметрами для инициализации данного класса выступают:

– loss – функция потерь, используемая в модели. По умолчанию используется функция «log\_loss» (13);

– n\_estimators – максимальное количество бустов, которое может быть использовано в обучении модели. По умолчанию равно 100;

– min\_sample\_split – минимальное количество наблюдений, необходимое для порождения ветви. По умолчанию равно 2;

– min\_sample\_leaf – минимальное количество наблюдений, необходимое для образования листа. По умолчанию равно 1;

– max\_depth – максимальная глубина дерева решения. По умолчанию равно 3. [9]

# 3.2 Построенные модели

На основе представленных выше классов было построено четыре ансамбля классификации.

Модель бэггинга. В таблице 3 представлены характеристики модели бэггинга, применительно к поставленной задаче классификации.

Таблица 3 – Характеристики модели бэггинга

|  |  |
| --- | --- |
| Параметр | Значение |
| n\_estimators | 6 |
| bootstrap | True |
| random\_state | 1 |

В качестве простой модели используется многослойный персептрон с одним скрытым слоем, 6 нейронами в скрытом слое. В качестве функции активации выступает гиперболический тангенс.

В таблицах 4 и 5 представлены результаты этой модели на обучающей и тестовой выборках.

Таблица 4 – Результаты бэггинга на обучающей выборке

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Всего | Правильно | Неправильно | Правильно, % | Неправильно, % |
| Высоко-успешный | 4 | 4 | 0 | 100 | 0 |
| Успешный | 13 | 12 | 1 | 92,3 | 7,7 |
| Основной | 283 | 282 | 1 | 100 | 0 |
| Мусорный | 29 | 29 | 0 | 100 | 0 |
| Всего | 329 | 327 | 2 | 99,39 | 0,61 |

Таблица 5 – Результаты бэггинга на тестовой выборке

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Всего | Правильно | Неправильно | Правильно, % | Неправильно, % |
| Высоко-успешный | 4 | 4 | 0 | 100 | 0 |
| Успешный | 8 | 6 | 2 | 75 | 25 |
| Основной | 331 | 324 | 7 | 97,89 | 2,11 |
| Мусорный | 36 | 32 | 4 | 88,89 | 13,11 |
| Всего | 379 | 366 | 13 | 96,57 | 3,43 |

Из приведённых таблиц можно сделать вывод, что бэггинг достаточно хорошо определяет основной и высоко-успешный рейтинг, но с успешным и мусорным она справляется чуть хуже.

На таблицах 6 и 7 приведены матрицы несоответствий для данной модели на обучающей и тестовой выборках. Строки выражают предсказание модели, а столбцы – истинные значения наблюдений.

Таблица 6 – Матрица несоответствии бэггинга на обучающей выборке

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | В.-успешный | Успешный | Основной | Мусорный |
| В.-успешный | 4 | 0 | 0 | 0 |
| Успешный | 1 | 12 | 0 | 0 |
| Основной | 0 | 1 | 282 | 0 |
| Мусорный | 0 | 0 | 0 | 29 |

Таблица 7 – Матрица несоответствий бэггинга на тестовой выборке

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | В.-успешный | Успешный | Основной | Мусорный |
| В.-успешный | 4 | 0 | 0 | 0 |
| Успешный | 1 | 6 | 1 | 0 |
| Основной | 0 | 7 | 324 | 0 |
| Мусорный | 0 | 0 | 6 | 32 |

По таблицам 6 и 7 можно сделать вывод, что большая часть ошибок, совершаемых представленной моделью бэггинга являются пессимистичные прогнозы. Например, относительно часто успешные банки определяются как основные, а основные как мусорные.

Модель стэкинга. В качестве единственной простой модели на уровне-0 здесь используется многослойный персептрон с одним скрытым слоем, 6 нейронами в скрытом слое и логистической функцией активации. На выходном уровне также применяется многослойный персептрон с одним скрытым слоем, но с 2 нейронами и гиперболическим тангенсом в качестве функции активации. Прочие характеристики модели представлены в таблице 8.

Таблица 8 – Характеристики модели стэкинга

|  |  |
| --- | --- |
| Параметр | Значение |
| cv | 5 |
| passthrough | False |

В таблицах 9 и 10 представлены результаты работы модели.

Таблица 9 – Результаты работы стэкинга на обучающей выборке

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Всего | Правильно | Неправильно | Правильно, % | Неправильно, % |
| Высоко-успешный | 5 | 5 | 0 | 100 | 0 |
| Успешный | 13 | 13 | 0 | 100 | 0 |
| Основной | 282 | 283 | 0 | 100 | 0 |
| Мусорный | 29 | 29 | 0 | 100 | 0 |
| Всего | 329 | 329 | 0 | 100 | 0 |

Таблица 10 – Результаты работы стэкинга на тестовой выборке

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Всего | Правильно | Неправильно | Правильно, % | Неправильно, % |
| Высоко-успешный | 6 | 5 | 1 | 83,33 | 16,67 |
| Успешный | 14 | 12 | 2 | 85,71 | 14,29 |
| Основной | 323 | 321 | 2 | 99,38 | 0,62 |
| Мусорный | 36 | 31 | 5 | 86,11 | 13,89 |
| Всего | 379 | 369 | 10 | 96,57 | 3,43 |

Несмотря на то, что стэкинг показал на обучающей выборке абсолютную точность, для тестовой ситуация неоднозначная. С одной стороны, великолепно себя показала при определении банков основного кластера, с другой – для всех остальных кластеров точность была существенно ниже.

На таблице 11 приведена матрица несоответствий для тестовой выборки. В то же время, для обучающей выборки нет смысла приводить матрицу несоответствий в силу абсолютной точности на ней.

Таблица 11 – Матрица несоответствий стэкинга на тестовой выборке

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | В.-успешный | Успешный | Основной | Мусорный |
| В.-успешный | 5 | 0 | 1 | 0 |
| Успешный | 0 | 12 | 2 | 0 |
| Основной | 0 | 1 | 321 | 1 |
| Мусорный | 0 | 0 | 5 | 31 |

Из матрицы видно, что модель довольно часто определяет банки из основного кластера как мусорные. С другой, стороны, имеются случаи оптимистичного предсказания, когда наблюдения основного кластера были определены в более высокие кластеры.

Модель адабуста. Слабым учеником здесь выступает дерево решений с максимальной глубиной 4. Характеристики модели представлены в таблице 12

Таблица 12 – Характеристики модели адабуста

|  |  |
| --- | --- |
| Параметр | Значение |
| n\_estimators | 50 |
| learning\_rate | 1 |
| algorithm | SAMME.R |
| random\_state | 2 |

Результаты этой модели на обоих выборках представлены в таблицах 13 и 14.

Таблица 13 – Результаты адабуста на обучающей выборке

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Всего | Правильно | Неправильно | Правильно, % | Неправильно, % |
| Высоко-успешный | 5 | 5 | 0 | 100 | 0 |
| Успешный | 13 | 13 | 0 | 100 | 0 |
| Основной | 282 | 283 | 0 | 100 | 0 |
| Мусорный | 29 | 29 | 0 | 100 | 0 |
| Всего | 329 | 329 | 0 | 100 | 0 |

Таблица 14 – Результаты адабуста на тестовой выборке

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Всего | Правильно | Неправильно | Правильно, % | Неправильно, % |
| Высоко-успешный | 5 | 5 | 0 | 100 | 0 |
| Успешный | 15 | 12 | 3 | 80 | 20 |
| Основной | 323 | 322 | 1 | 99,69 | 0,31 |
| Мусорный | 36 | 32 | 4 | 88,88 | 11,12 |
| Всего | 379 | 369 | 8 | 97,88 | 2,12 |

Эта модель также показала абсолютную точность на обучающей выборке, что позволяет не рассматривать матрицу несоответствий на ней. На тестовой выборке она смогла безошибочно определить высоко-успешные банки. Кроме того, она показала высокую точность при определении банков из основного массива.

На таблице 15 представлена матрица несоответствий модели адабуста на тестовой выборке

Таблица 15 – Матрица несоответствий адабуста на тестовой выборке

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | В.-успешный | Успешный | Основной | Мусорный |
| В.-успешный | 5 | 0 | 0 | 0 |
| Успешный | 0 | 12 | 3 | 0 |
| Основной | 0 | 1 | 322 | 0 |
| Мусорный | 0 | 0 | 4 | 32 |

На основе данных в таблице 15 можно сделать вывод, что адабуст может классифицировать банки основного кластера как успешные, чуть реже наоборот. Также часть банков основной группы определяется как мусорные.

Модель градиентного бустинга. Характеристики этой модели приведены в таблице 16.

Таблица 16 – Характеристики модели градиентного бустинга

|  |  |
| --- | --- |
| Параметр | Значение |
| loss | log\_loss |
| n\_estimators | 100 |
| min\_sample\_split | 2 |
| min\_sample\_leaf | 1 |
| max\_depth | 3 |
| random\_state | 3 |

Для модели градиентного бустинга были получены результаты, представленные в таблицах 17 и 18.

Таблица 18 – Результаты градиентного бустинга на обучающей выборке

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Всего | Правильно | Неправильно | Правильно, % | Неправильно, % |
| Высоко-успешный | 5 | 5 | 0 | 100 | 0 |
| Успешный | 13 | 13 | 0 | 100 | 0 |
| Основной | 282 | 283 | 0 | 100 | 0 |
| Мусорный | 29 | 29 | 0 | 100 | 0 |
| Всего | 329 | 329 | 0 | 100 | 0 |

Таблица 19 – Результаты градиентного бустинга для тестовой выборки

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Всего | Правильно | Неправильно | Правильно, % | Неправильно, % |
| Высоко-успешный | 5 | 5 | 0 | 100 | 0 |
| Успешный | 14 | 12 | 2 | 85,71 | 14,29 |
| Основной | 323 | 322 | 1 | 99,69 | 0,31 |
| Мусорный | 36 | 32 | 4 | 88,88 | 11,12 |
| Всего | 379 | 369 | 7 | 98,15 | 1,85 |

На основе данных таблиц 18 и 19 можно сделать вывод о высокой прогностической способности градиентного бустинга. Обучающая выборка была определена безошибочно, что снова позволяет не рассматривать матрицу несоответствий для нее. Слабым местом выступили банки, относящиеся к успешному и мусорному кластерам. В то же время модель безошибочно определила высоко-успешные банки и с одной ошибкой – основные банки.

В таблице 20 представлена матрица несоответствий градиентного бустинга на тестовой выборке.

Таблица 20 – Матрица несоответствий градиентного бустинга на тестовой выборке

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | В.-успешный | Успешный | Основной | Мусорный |
| В.-успешный | 5 | 0 | 0 | 0 |
| Успешный | 0 | 12 | 2 | 0 |
| Основной | 0 | 1 | 323 | 0 |
| Мусорный | 0 | 0 | 4 | 32 |

Из таблицы можно сделать вывод, что основная ошибка для данной модели – определение основных банков как мусорных. В то же время было два случая оптимистичной оценки основных банков, а также один случай пессимистичной оценки успешного банка.

# 3.3 Сравнительный анализ моделей

Представленные методы в целом показали достаточно хорошую обобщающую способность. Для всех методов ошибка на тестовом наборе не превысила 5%. Самым результативным оказался градиентный бустинг с точностью в 98,15 %, самым слабым – бэггинг, имеющий точность 96,57 %. Между ними стэкинг и адабуст, имеющие точности 97,36 % и 97,89 % соответственно.

Для кластера высоко-успешных банков безошибочно справились адабуст и градиентный бустинг. В это же время бэггинг определил один из высоко-успешных банков как просто успешный, а стэкинг ошибочно отнес к этой группе банк из основного кластера.

Для кластера успешных банков лучше всего справились методы стэкинга и градиентного бустинга дважды ошибочно определив банки основного кластера как успешные, а также определив один успешный банк как основной. Адабуст чуть хуже справился с тремя определениями основных банков в этот кластер. Хуже всего на этом кластере отработал бэггинг, который определил в этот кластер один высоко-успешный и один основной банк, а также определив 7 успешных банков как основные.

Для кластера основных банков лучше всего справился градиентный бустинг, ошибочно определив в этот кластер один успешный банк, а также определив 6 основных банков в другие кластеры. Следом идет адабуст, который в отличии от градиентного бустинга определил 7 основных банков в другие кластеры. Метод стэкинга оказался чуть хуже адабуста тем, что ошибочно внес два успешных банка в основной кластер. Хуже всего справился бэггинг, поскольку определил 7 успешных банков как основные, но «потеряв» при этом 5 основных банков.

Для кластера мусорных банков одинаково хорошо справились методы адабуста, бэггинга и градиентного бустинга, ошибочно определив 4 банка из основного кластера в этот. Хуже справился стэкинг, определив в этот кластер 5 банков из основного и определив один мусорный банк как основной.

# 4 Пользовательский интерфейс

Для построенных была создана графическая оболочка с использованием фреймворка PyQt5. В качестве средства создания интерфейса был выбран QtDesigner. На рисунке 12 представлен графический интерфейс готовой программы.

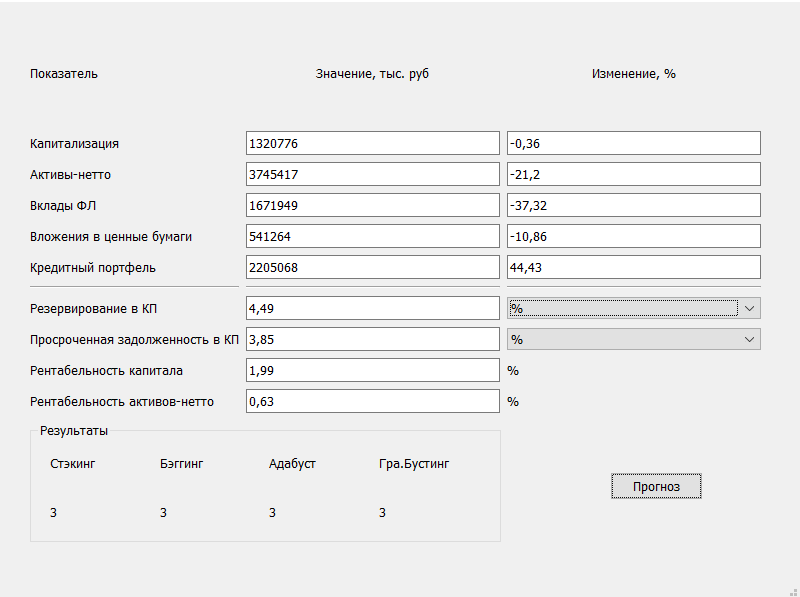


Рисунок 12 – Графический интерфейс готовой программы

В окне, представленном на рисунке 12 вводятся показатели банка в соответствующих единицах измерения. Для показателей резервирования в кредитном портфеле и просроченной задолженности в кредитном портфеле реализована возможность выбора единицы измерений. Возможны варианты процентов и тысяч рублей.

При нажатии кнопки «Прогноз» данные обрабатываются схожим образом, что был представлен в главе 2. Сперва денежные показатели делятся на соответствующие значения Центробанка, сохраненные в файле «CBR\_data.json». В зависимости от выбора единицы измерения вычисляются уровень резервирования в кредитном портфеле и уровень просроченной задолженности в кредитном портфеле. Для полученных данных вычисляется расстояние от БКЛ, представленного в файле «BCL.json». Обработанные данные затем направляются на прогнозирование в реализованные модели. Результат этих моделей затем выводится в область «Результаты», представленную на рисунке 13.

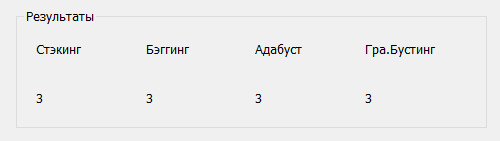


Рисунок 13 – Область результатов для прогнозируемого банка

# ЗАКЛЮЧЕНИЕ

В данной работе рассмотрены следующие методы ансамблирования машинного обучения: бэггинг, стэкинг, адабуст, а также градиентный бустинг. Были изучены их теоретические основы, их практическая реализация в фреймворке Scikit-learn. Пакет Scikit-learn предоставляет достаточный набор инструментов для создания и настройке указанных методов ансамблирования.

Реализован способ оценки экономического состояния банков с использованием модификации метрического подхода и применением кластеризации.

Созданы адекватные модели классификации на основе указанных методов ансамблирования для определения принадлежности банка к кластеру на основе его показателей. Модели реализованы с помощью инструментов языка Python

Оценены обобщающие способности этих методов. Эти оценки были сравнены между собой.

Для полученных методов реализован графический интерфейс для взаимодействия с ними.

Таким образом все поставленные задачи данной работы были решены полностью.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Breiman L. Bagging predictors // L. Breiman // Machine Learning – 1996 – № 24. – P. 123-140

2. Wolpert D. H. Stacked generalization // D. H. Wolpert //Neural networks: the official journal of the International Neural Network Society – 1992 – vol. 5 – №2. – P. 241-259

3. Freund Y.// A short introduction to boosting // Y. Freund, R. E. Schapire// Journal of Japanese Society for Artificial Intelligence – 1999 – vol. 14 – №5. – P.771-780

4. Lecture R. // Greedy function approximation: a gradient boosting machine // R. Lecture // The Annals of Statistics – 2001 – vol.29 – №5. – P. 1189-1232

5. Банки.ру: информационный портал: рейтинги банков // (Рус.) . – URL:. http://www.banki.ru/banks/?order=fin\_rating [16 Мая 2022]

6. ЦБР: Статистические показатели банковского сектора // (Рус.) . – URL:. https://cbr.ru/statistics/bank\_sector/review/ [16 Мая 2022]

7. Халафян А. А. // Составление и оценка согласованности банковских рейтингов средствами компьютерного анализа // А. А. Халафян., И. В. Шевченко // Финансы и кредит. – 2017. – Т.23, №28. – с.1655–1677

8. Кто использует Scikit-learn // (Engl.). – URL:. https://scikit-learn.org/stable/testimonials/testimonials.html [11 June 2022]

9. Электронное руководство модуля Scikit-learn.Ensemble // (Engl.). – URL:. https://scikit-learn.org/stable/modules/classes.html#module-sklearn.ensemble

# ПРИЛОЖЕНИЕ А

# Дополнительные статистические даные

Таблица А.1 – Данные центрального банка по используемым показателям

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Дата | 1.01.2018 | 1.01.2019 | 1.01.2020 | 1.01.2201 | 1.01.2022 |
| Капитализация | 9397,30 | 10269,29 | 10981,13 | 11413,14 | 12599 |
| Активы -нетто | 77961,13 | 86231,98 | 88796,19 | 103841,75 | 120310,47 |
| Вклады ФЛ | 25986,74 | 28459,45 | 30 411,88 | 32834,23 | 34694,76 |
| Вложения в Цен. Бум. | 10563,89 | 11484,17 | 12011,72 | 16150,93 | 17289,31 |
| Кредитный портфель | 45993,63 | 52912,37 | 56654,42 | 64803,83 | 74949,0 |

Таблица А.2 – Статистические показатели неразмеченных обучающей и тестовой выборок

|  |  |  |  |
| --- | --- | --- | --- |
|  | Обучающая выборка | Тестовая выборка | Всего |
| Капитализация среднее | 0,0029 | 0,0033 | 0,0031 |
| Капитализация ст. откл. | 0,0232 | 0,0246 | 0,0239 |
| Капитализация относительное изменение среднее | 0,0001 | 0,0004 | 0,0003 |
| Капитализация отн. изм. ст. откл. | 0,0011 | 0,0025 | 0,0019 |
| Кредитный портфель среднее | 0,0025 | 0,0029 | 0,0027 |
| Кредитный портфель ст. откл. | 0,0208 | 0,0224 | 0,0226 |
| Кредитный портфель отн. изм. среднее | 0,0004 | 0,0005 | 0,0004 |
| Кредитный портфель отн. изм. ст. откл. | 0,0035 | 0,0038 | 0,0037 |
| Рентабельность капитала среднее | 0,0712 | 0,0547 | 0,0635 |
| Рентабельность капитала ст. откл. | 0,2117 | 0,227 | 0,2189 |
| Рентабельность активов-нетто среднее | 0,0137 | 0,0108 | 0,0124 |
| Рентабельность активов-нетто ст. откл. | 0,0588 | 0,0504 | 0,055 |
| Уровень просроченной задолженности среднее | 0,1123 | 0,1086 | 0,1106 |
| Уровень просроченной задолженности ст. откл. | 0,199 | 0,1955 | 0,1973 |
| Уровень резервирования среднее | 0,187 | 0,1494 | 0,1696 |
| Уровень резервирования ст. откл. | 0,2177 | 0,1994 | 0,21 |
| Вложения в цен. бум. среднее | 0,0024 | 0,0027 | 0,0025 |
| Вложения в цен. бум. ст. откл. | 0,0182 | 0,0198 | 0,0190 |

Продолжение таблицы А.2

|  |  |  |  |
| --- | --- | --- | --- |
|  | Обучающая выборка | Тестовая выборка | Всего |
| Вложения в цен. бум. отн. изм. среднее | 0,0014 | 0,0004 | 0,0009 |
| Вложения в цен. бум. отн. изм. ст. откл. | 0,0151 | 0,0059 | 0,0117 |
| Вклады ФЛ среднее | 0,0026 | 0,003 | 0,0027 |
| Вклады ФЛ ст. откл. | 0,0246 | 0,026 | 0,0252 |
| Вклады ФЛ отн. изм. среднее | 0,0003 | 0,0002 | 0,0002 |
| Вклады ФЛ отн. изм. ст. откл. | 0,0028 | 0,0015 | 0,0023 |
| Активы-нетто среднее | 0,0027 | 0,003 | 0,0028 |
| Активы-нетто ст. откл. | 0,02 | 0,0207 | 0,0203 |
| Активы-нетто отн. изм. среднее | 0,0005 | 0,0004 | 0,0005 |
| Активы-нетто отн. изм. ст. откл. | 0,0041 | 0,0026 | 0,0035 |
| Расстояние от БКЛ среднее | 1,2551 | 1,1961 | 1,2277 |
| Расстояние от БКЛ ст. откл. | 0,1258 | 0,2664 | 0,2055 |
| Количество наблюдений | 379 | 329 | 708 |

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание