APPLICATION OF GRADIENT BOOSTING AND RANDOM FOREST METHODS FOR ESG ESTIMATION-CRITERIA FOR ISSUING BANK LOANS

Korchinsky A.
Postgraduate student of the Faculty of Economics
Belarusian State University
Minsk, Republic of Belarus

ПРИМЕНЕНИЕ МЕТОДОВ ГРАДИЕНТНОГО БУСТИНГА И СЛУЧАЙНОГО ЛЕСА ДЛЯ ОЦЕНКИ ESG-КРИТЕРИЕВ ПРИ ВЫДАЧЕ БАНКОВСКИХ КРЕДИТОВ

Корчинский А.С.
аспирант экономического факультета
Белорусский государственный университет
г. Минск, Республика Беларусь
https://doi.org/10.5281/zenodo.18081608

Abstract

With the growing importance of sustainable development and the introduction of ESG criteria into financial practice, banks are faced with the need to integrate new approaches to assessing borrowers' creditworthiness. The article discusses the use of machine learning methods — gradient boosting and random forest — to analyze companies' ESG indicators when making loan decisions. The proposed approach makes it possible to take into account not only traditional financial metrics, but also environmental, social and managerial factors, which increases the accuracy of risk forecasting and contributes to the formation of a more responsible credit policy. The results of the study demonstrate the potential of using ensemble algorithms to build interpretable models capable of identifying key ESG determinants that affect the probability of default and the stability of the borrower.

Аннотация

В условиях роста значимости устойчивого развития и внедрения ESG-критериев в финансовую практику банки сталкиваются с необходимостью интеграции новых подходов к оценке кредитоспособности заемщиков. В статье рассматривается применение методов машинного обучения — градиентного бустинга и случайного леса — для анализа ESG-показателей компаний при принятии решений о выдаче кредитов. Предлагаемый подход позволяет учитывать не только традиционные финансовые метрики, но и экологические, социальные и управленческие факторы, что повышает точность прогнозирования рисков и способствует формированию более ответственной кредитной политики. Результаты исследования демонстрируют потенциал использования ансамблевых алгоритмов для построения интерпретируемых моделей, способных выявлять ключевые ESG-детерминанты, влияющие на вероятность дефолта и устойчивость заемщика.

Keywords: ESG; loans; banks; sustainable financing; financial metrics; risk; gradient boosting; random forest; interpretability of models; sustainable development.

Ключевые слова: ESG; кредиты; банки; устойчивое финансирование; финансовые метрики; риск; градиентный бустинг; случайный лес; интерпретируемость моделей; устойчивое развитие.

Понятие ESG возникло в начале 2000-х годов, когда Организация Объединённых Наций инициировала программу «Принципы ответственного инвестирования» (PRI). В 2004 году в докладе «Who Cares Wins» впервые было предложено систематически учитывать экологические, социальные и управленческие факторы при принятии инвестиционных решений. С этого момента ESG стало международным стандартом устойчивого финансирования, а банки начали интегрировать его в практику кредитования.

Методы машинного обучения, используемые для анализа ESG-данных, появились немного раньше. Алгоритм градиентного бустинга был предложен Джеромом Фридманом в 1999 году в Стэнфорде как развитие идей AdaBoost. Он рассматривает построение ансамбля моделей как задачу оптимизации функции потерь и позволяет достигать высокой точности прогнозов. Алгоритм случайного леса разработал Лео Брейман в 2001 году в Университете Калифорнии, Беркли. Его идея заключалась в построении множества деревьев решений на случайных подвыборках данных и признаков, что делает модель устойчивой к переобучению и шуму.

Современные банки сталкиваются с задачей интеграции ESG-критериев в процесс оценки кредитоспособности. Традиционные статистические методы не всегда позволяют учесть сложные взаимосвязи между финансовыми показателями и нефинансовыми факторами. Здесь на помощь приходят ансамблевые алгоритмы машинного обучения. Случайный лес и градиентный бустинг способны работать с разнородными данными, выявлять скрытые закономерности и формировать интерпретируемые модели, которые помогают кредитным аналитикам принимать более взвешенные решения.

Для оценки кредитоспособности заемщиков с учётом ESG-критериев предлагается использовать ансамблевые методы машинного обучения — градиентный бустинг и случайный лес. Эти алгоритмы позволяют работать с разнородными данными, включающими как количественные финансовые показатели (выручка, долговая нагрузка, ликвидность), так и качественные ESG-метрики (экологическая политика компании, социальные инициативы, структура корпоративного управления).

Подготовка данных включает несколько этапов:

1) Сбор информации из финансовой отчётности и специализированных ESG-рейтинговых агентств;

2) Очистка и нормализация данных для устранения пропусков и несоответствий;

3) Кодирование категориальных признаков, таких как наличие экологической сертификации или участие в социальных программах;

4) Формирование обучающей выборки, где целевая переменная отражает факт одобрения или отказа в кредите [1, p.17].

Обособленная разность методов очень растяжима.

Случайный лес строит множество деревьев решений, каждое из которых обучается на случайной подвыборке признаков и данных. Итоговое решение формируется путём агрегирования результатов всех деревьев. Такой подход снижает риск переобучения и делает модель устойчивой к шуму, что особенно важно при работе с ESG-данными, где часть показателей может быть субъективной или неполной.

Градиентный бустинг обучает последовательность деревьев, каждое из которых исправляет ошибки предыдущего. Алгоритм минимизирует функцию потерь, приближая её градиент. Это позволяет достигать высокой точности прогнозов и выявлять сложные нелинейные зависимости между финансовыми и ESG-факторами.

Существует ряд метрик, которые специалисты во многих банках используют, как показатель при дополнительном анализе заемщике, такие как метрики оценки качества. Данные метрики используются для проверки эффективности моделей:

a) ROC-AUC — оценка качества бинарной классификации;

b) F1-score — баланс между точностью и полнотой;

c) Accuracy — общая точность модели [2, p.3];

d) Интерпретируемость (SHAP-значения) — анализ вклада каждого признака в итоговое решение.

Применение ансамблевых алгоритмов позволяет банкам не только повысить точность прогнозирования кредитных рисков, но и выявить ключевые ESG-факторы, влияющие на вероятность дефолта. Например, наличие прозрачной системы корпоративного управления может снижать риск невозврата кредита, а высокая углеродная нагрузка — наоборот, увеличивать его.

На сегодняшний день градиентный бустинг (gradient boosting machine) является одним из основных production-решений при работе с табличными, неоднородными данными, поскольку обладает высокой производительностью и точностью [3, p.5].

Алгоритм работает следующим образом:

a) Строится базовое дерево решений, которое делает первичный прогноз вероятности дефолта;

b) Вычисляются ошибки прогноза, и на их основе формируется новое дерево, корректирующее результат;

c) Процесс повторяется многократно, каждое дерево добавляет уточнение, приближая модель к оптимальному решению;

d) Итоговый прогноз формируется как сумма всех деревьев, взвешенных по коэффициентам обучения.

Для оценки качества модели применяются метрики ROC-AUC, F1-score и анализ интерпретируемости с помощью SHAP-значений, которые позволяют определить вклад каждого ESG-фактора в итоговое решение. Например, модель может показать, что высокий уровень выбросов CO_2 увеличивает вероятность отказа в кредите, а наличие независимого совета директоров снижает риски [4, p.5].

Алгоритм Random Forest был предложен Лео Брейманом в 2001 году и стал одним из самых популярных методов машинного обучения для задач классификации и регрессии. Его ключевая идея заключается в построении множества деревьев решений, каждое из которых обучается на случайной подвыборке данных и признаков. Итоговое решение формируется путём агрегирования результатов всех деревьев — например, голосованием в задаче классификации или усреднением в задаче регрессии. Такой подход снижает риск переобучения и делает модель устойчивой к шуму. В контексте банковского кредитования с учётом ESG-критериев Random Forest позволяет анализировать широкий спектр признаков, включая:

1) Финансовые показатели: ликвидность, долговая нагрузка, рентабельность;

2) Экологические факторы: углеродный след компании, наличие экологических сертификатов;

3) Социальные факторы: участие в социальных инициативах, соблюдение трудовых прав;

4) Управленческие факторы: прозрачность корпоративного управления, независимость совета директоров [5, p.7].

Особенность Random Forest заключается в том, что он хорошо работает с данными, где признаки могут быть скоррелированы или содержать шум. Для ESG-оценки это особенно важно, так как часть показателей может быть субъективной или неполной. Модель способна выявлять значимые факторы даже в условиях высокой вариативности данных.

Пример применения Random Forest в кредитном процессе:

a) Формируется обучающая выборка с финансовыми и ESG-показателями компаний;

b) Строится множество деревьев решений, каждое из которых обучается на случайной подвыборке признаков;

c) Итоговое решение формируется путём агрегирования результатов всех деревьев.

Анализ важности признаков позволяет определить, какие ESG-факторы наиболее сильно влияют на вероятность дефолта.

Для оценки качества модели применяются метрики ROC-AUC, F1-score и анализ важности признаков. Например, Random Forest может показать, что наличие экологической сертификации снижает вероятность дефолта, а высокая углеродная нагрузка увеличивает риски.

Таким образом, Random Forest (далее RF) является надёжным инструментом для интеграции ESG-критериев в кредитный процесс. Он обеспечивает устойчивость модели к шуму, позволяет выявлять ключевые факторы устойчивости и помогает банкам принимать более взвешенные решения при выдаче кредитов [6, p.3].

Стоит подчеркнуть, что данные методы отличаются подходами в обучении. Random Forest строит множество деревьев решений параллельно, каждое на случайной подвыборке данных и признаков. Итоговое решение формируется путём агрегирования результатов всех деревьев [7, p.10].

Gradient Boosting Machine (далее GBM) строит деревья последовательно, каждое новое дерево исправляет ошибки предыдущего, минимизируя функцию потерь.

RF лучше справляется с шумными и неполными данными, что важно для ESG-метрик, где часть показателей может быть субъективной или неполной. GBM более чувствителен к качеству данных, но способен выявлять сложные нелинейные зависимости между финансовыми и ESG-факторами [8, p.9].

Важную роль играет интерпретируемость. RF предоставляет простую оценку важности признаков (feature importance), позволяя понять, какие ESG-факторы влияют на решение. GBM требует более сложных инструментов интерпретации (например, SHAP-значения), но даёт более детализированное объяснение вклада каждого признака.

В банковской сфере уже есть примеры использования данных методах, однако, неполностью доказана их значимость при техническом анализе. RF больше подходит для быстрых пилотных проектов и ситуаций, где важна устойчивость модели к шуму. GBM же предпочтителен для задач, где требуется максимальная точность и возможность выявления сложных взаимосвязей между ESG-факторами и кредитным риском [10, p.1].

Оба метода дополняют друг друга: RF обеспечивает надёжность и устойчивость, а GBM — высокую точность и гибкость. В банковской практике целесообразно использовать их совместно: RF — для первичной оценки и отбора признаков, GBM — для построения точных скоринговых моделей, учитывающих ESG-критерии (табл. 1).

Таблица 1

Таблица сравнения GBM и RF для ESG-кредитования

Критерий	Gradient Boosting Machine (GBM)	Random Forest (RF)
Принцип работы	Последовательное построение деревьев, каждое исправляет ошибки предыдущего	Параллельное построение множества деревьев на случайных подвыборках
Точность прогнозов	Обычно выше при правильной настройке гиперпараметров	Стабильная, но может уступать GBM
Устойчивость к шуму	Чувствителен к качеству данных	Высокая устойчивость к шумным и неполным данным
Интерпретируемость	Требует сложных методов (SHAP, LIME), но даёт детализированные объяснения	Простая оценка важности признаков, легко понять вклад ESG-факторов
Скорость обучения	Медленнее, особенно на больших выборках	Быстрее, хорошо масштабируется
Применимость к ESG	Выявляет сложные нелинейные зависимости между финансовыми и ESG-факторами	Надёжно работает при высокой вариативности и неполноте ESG-данных
Риск переобучения	Высокий при неправильной настройке	Низкий благодаря случайности и усреднению
Практическая роль	Подходит для построения точных скоринговых моделей	Эффективен для первичной оценки и отбора признаков

Как и в случае с Adaboost, градиентный бустинг добавляет базовые модели в ансамбль последовательно, однако вместо обучения моделей с учётом весов на основе ошибок предшественников, в данном случае модели обучаются на остаточных ошибках (residual errors), допущенных предыдущими моделями.

Существует два принципа работы GBM, для регрессии и для классификации:

Алгоритм строится следующим образом:

1) первоначальному прогнозу присваивается среднее значение y_train для всех образцов;

2) рассчитываются остатки модели на основе антиградиента функции потерь;

3) регрессионное дерево обучается на X_train и остатках, далее делается прогноз на X_train;

4) полученный прогноз добавляется к первоначальному и шаги 2-4 повторяются для каждого дерева;

5) после обучения всех моделей снова создаётся первоначальный прогноз из шага 1;

6) далее делаются прогнозы для X_test на обученных деревьях и добавляются к первоначальному;

7) полученная сумма и будет конечным прогнозом.

Когда мы изучаем классическое машинное обучение, неизбежно сталкиваемся с парадоксом: материалов по теме много, но интуитивно понятных объяснений того, почему ансамбли настолько эффективны, — удивительно мало. Попробуем восполнить этот пробел и разложить концепцию ансамблей по полочкам.

Ансамбли представляют собой особую парадигму машинного обучения. В её основе лежит идея объединения нескольких моделей, каждая из которых по отдельности может быть «слабой» в решении задачи. Однако вместе они формируют систему, способную выдавать значительно более точные и устойчивые результаты.

Главная мысль проста: множество слабых алгоритмов, действующих сообща, превращаются в один сильный. Это похоже на коллективное решение, где каждый участник вносит свой вклад, а итоговое решение оказывается более надёжным, чем мнение одного эксперта [9, p.4].

Чтобы понять силу ансамблей, нужно вспомнить, из чего складывается ошибка модели. В машинном обучении принято выделять три компонента:

1. Смещение (bias) — систематическая ошибка, возникающая из-за упрощённых предположений модели;
2. Разброс (variance) — чувствительность модели к конкретным данным, из-за чего результаты могут сильно колебаться;
3. Шум — неизбежная часть данных, которую невозможно устранить полностью.

$$Error = bias^2 + variance + noise \quad (1)$$

Ансамбли помогают сбалансировать эти составляющие. Объединяя разные модели, мы снижаем разброс, уменьшаем влияние смещения и получаем более устойчивый результат. В итоге система оказывается менее подверженной случайным ошибкам и лучше справляется с задачами, чем отдельные алгоритмы (рис.1).

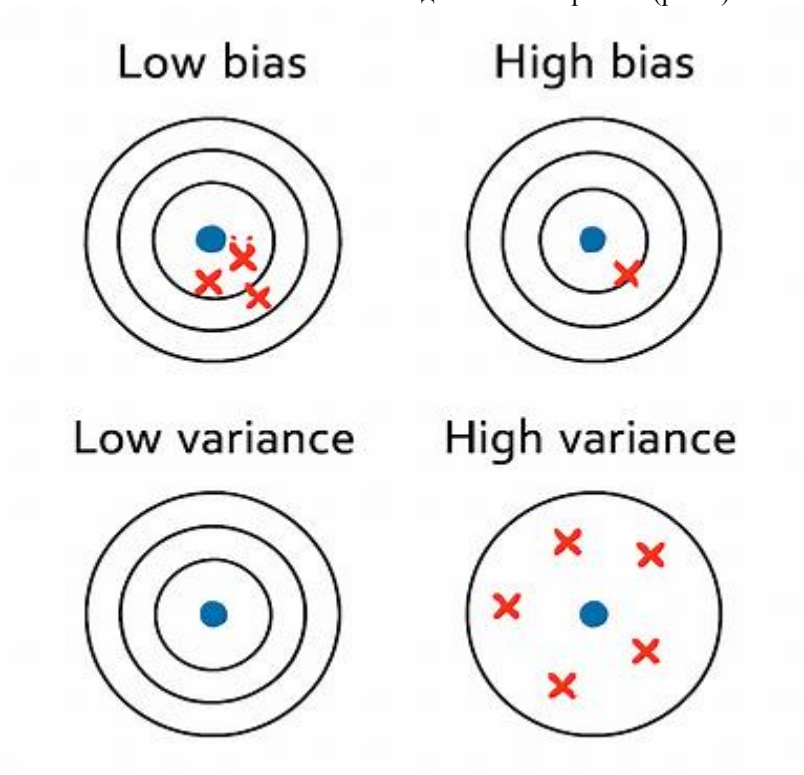

Рисунок 1

Когда мы смотрим на «мишени» для смещения и разброса, мы фактически визуализируем два аспекта ошибки модели: где в среднем бьём относительно цели (смещение) и насколько далеко разбросаны отдельные попадания (разброс). Это помогает понять характер ошибок и выбрать стратегии улучшения модели.

Разброс показывает, насколько прогнозы модели колеблются. Высокий разброс — признак переобучения: модель слишком чувствительна к данным и выдаёт разные ответы в зависимости от выборки. Низкий разброс означает устойчивость: результаты стабильны, но при высоком смещении они могут быть стабильно неверными. В идеале мы стремимся к низкому смещению и низкому разбросу — тогда модель и точна, и надёжна. Именно ансамбли помогают приблизиться к этой цели: Случайный Лес снижает разброс, а Градиентный Бустинг уменьшает смещение.

Именно в этом и заключается суть градиентного бустинга. Вместо того чтобы напрямую минимизировать сложную функцию потерь, алгоритм на каждом шаге строит новое дерево, которое пытается предсказать градиент этой функции — то есть направление, в котором ансамбль ошибается.

Проще говоря, каждое последующее дерево добавляется для исправления наиболее заметных промахов предыдущих моделей. Подробное объяснение математической основы заслуживает отдельного разбора, но именно идея обучения на градиенте ошибки и дала название этому методу.

Для начала анализа загрузим все необходимые нам библиотеки:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_diabetes
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, mean_absolute_percentage_error
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import GradientBoostingClassifier, GradientBoostingRegressor
from sklearn.ensemble import HistGradientBoostingClassifier, HistGradientBoostingRegressor
from mlxtend.plotting import plot_decision_regions (2)
```

Каждый импорт в твоём коде выполняет свою роль и нужен для определённых задач. Библиотеки NumPy и Pandas обеспечивают работу с данными: первая отвечает за быстрые численные операции с массивами, а вторая — за удобное хранение и обработку табличных структур. Matplotlib используется для построения графиков и визуализации результатов. Встроенный датасет load_diabetes из sklearn.datasets служит примером для регрессионных задач, а LabelEncoder помогает преобразовать категориальные признаки в числовые значения.

Для подготовки данных применяется train_test_split, который делит выборку на обучающую и тестовую части. Метрики accuracy_score и mean_absolute_percentage_error позволяют оценить качество моделей: первая — для классификации, вторая — для регрессии. Среди моделей у тебя есть DecisionTreeRegressor — базовое дерево решений для регрессии, а также ансамблевые методы: GradientBoostingClassifier и GradientBoostingRegressor для классического градиентного бустинга, и их более современный вариант — HistGradientBoostingClassifier и HistGradientBoostingRegressor, оптимизированные для больших данных. Наконец, функция plot_decision_regions из библиотеки MLxtend используется для наглядной визуализации областей решений классификаторов, показывая, как модель разделяет классы на плоскости.

Попробуем выполнить на языке python:

```
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeRegressor
class SimpleGBMClassifier:
def __init__(self, learning_rate=0.1, n_estimators=100, max_depth=3, random_state=0):
self.learning_rate = learning_rate
self.n_estimators = n_estimators
self.max_depth = max_depth
self.random_state = random_state
def _softmax(self, logits):
exp_logits = np.exp(logits - np.max(logits, axis=1, keepdims=True))
return exp_logits / np.sum(exp_logits, axis=1, keepdims=True)
def fit(self, X, y):
self.classes_ = np.unique(y)
self.K = len(self.classes_)
self.trees = [[] for _ in range(self.K)]
y_onehot = pd.get_dummies(y).values
predictions = np.zeros_like(y_onehot)
for _ in range(self.n_estimators):
probs = self._softmax(predictions)
for k in range(self.K):
residuals = y_onehot[:, k] - probs[:, k]
tree = DecisionTreeRegressor(max_depth=self.max_depth, random_state=self.random_state)
tree.fit(X, residuals)
self.trees[k].append(tree)
predictions[:, k] += self.learning_rate * tree.predict(X)
def predict(self, X):
scores = np.zeros((X.shape[0], self.K))

for k in range(self.K):
for tree in self.trees[k]:
scores[:, k] += self.learning_rate * tree.predict(X)
return self.classes_[np.argmax(scores, axis=1)] (3)
```

Данный код реализует класс GBMClassifier, который представляет собой классификатор на основе градиентного бустинга.

В классе GBMClassifier определены следующие методы:

1) _softmax — функция для преобразования прогнозов в вероятности с помощью softmax-функции;
2) _compute_gammas — функция для вычисления коэффициентов \gamma, которые регулируют степень вклада каждого нового дерева в общую модель;
3) fit — метод для обучения модели на данных;
4) predict — метод для прогнозирования классов объектов.

Особенностью данного классификатора является возможность использования концепции K-class LogitBoost, которая позволяет улучшить качество прогнозов за счёт расчёта весов для каждого дерева и использования их при вычислении остатков.

Этот код может быть полезен для решения задач классификации с использованием градиентного бустинга, особенно в случаях, когда необходимо улучшить качество прогнозов или учесть специфику данных.

```
def decision_boundary_plot(X, y, X_train, y_train, clf, feature_indexes, title=None):
feature1_name, feature2_name = X.columns[feature_indexes]
```

```
X_feature_columns = X.values[:, feature_indexes]
X_train_feature_columns = X_train.values[:, feature_indexes]
clf.fit(X_train_feature_columns, y_train.values)

plot_decision_regions(X=X_feature_columns, y=y.values, clf=clf)
plt.xlabel(feature1_name)
plt.ylabel(feature2_name)
plt.title(title) (4)
```

Функция decision_boundary_plot предназначена для визуализации решающих границ классификатора. Она принимает на вход данные (X и y), обучающую выборку (X_train и y_train), классификатор (clf) и индексы признаков, которые будут использоваться для построения графика.

Функция извлекает имена признаков, соответствующие указанным индексам, затем подгоняет классификатор к обучающей выборке и использует функцию plot_decision_regions для построения графика решающих границ. Оси графика помечаются именами признаков, а заголовок графика задаётся параметром title.

Этот код полезен для визуализации работы классификатора и анализа его решений в пространстве признаков.

Для задач прогнозирования кредитоспособности предприятий по ESG-факторам банки используют специализированные коммерческие наборы данных, такие как S&P Global ESG Scores, LSEG ESG Company Data или SIX ESG Data Hub. Эти датасеты содержат показатели устойчивого развития (экология, социальная ответственность, корпоративное управление), которые можно связать с финансовыми метриками для оценки риска и кредитоспособности.

Список литературы:

1. https://link.springer.com/article/10.1007/s10462-020-09896-5
2. https://arxiv.org/abs/2305.17094
3. https://en.wikipedia.org/wiki/Gradient_boosting
4. https://arxiv.org/pdf/1511.05741
5. https://journals.sagepub.com/doi/pdf/10.1177/1536867X20909688
6. https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/
7. https://machinelearningmastery.com/how-to-decide-between-random-forests-and-gradient-boosting/
8. https://www.geeksforgeeks.org/machine-learning/gradient-boosting-vs-random-forest/
9. https://www.ibm.com/think/topics/gradient-boosting
10. https://serokell.io/blog/random-forest-classification