УДК [004.032.26+ 004.852] : 336.77.067

МЕТОДЫ ОЦЕНКИ КРЕДИТНЫХ РИСКОВ

Д.В. Шичков
Магистрант БГУИР,
инженер-программист СКБ
Радиотехпроект

И.И. Фролов
кандидат технических наук,
доцент, кафедра ЭВМ,
БГУИР

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь.
ООО «СКБ Радиотехпроект», Республика Беларусь.
E-mail: dmitry_shychkov@mail.ru.

Д. В. Шичков
Окончил Белорусский государственный университет информатики и радиоэлектроники. Магистрант БГУИР. Работает в СКБ Радиотехпроект в должности инженера-программиста. Проводит научные исследования оценки кредитных рисков с помощью различных методов классификаций.

И. И. Фролов
Окончил Белорусский государственный университет информатики и радиоэлектроники. Доцент кафедры ЭВМ БГУИР. Проводит научные исследования в области машинного обучения и компьютерного зрения, участвует в проектировании и разработке систем технического зрения.

Аннотация. В данной работе исследуются классификаторы машинного обучения с учителем для прогнозирования результатов кредитования. Учитывая большое количество алгоритмов, анализ начинается с простых методов, таких как логистическая регрессия, с постепенным увеличением сложности модели до методов рандомизированных деревьев. Далее сравнивается производительность каждой модели и обсуждается наиболее подходящее для этой задачи кредитования решение классификации. Исследованы и построены следующие модели: логистическая регрессия; стохастический градиентный спуск; метод опорных векторов; градиентный спуск; рандомизированные деревья.

Ключевые слова: кредитный скоринг, кредитный риск, логистическая регрессия, стохастический градиентный спуск, метод опорных векторов, градиентный спуск, рандомизированные деревья.

Введение.

Оценка кредитных рисков или кредитный скоринг является одним из наиболее «классических» приложёний для прогнозного моделирования, чтобы предсказать, одобрить ли кредит потенциальному заёмщику, и приведёт ли это к прибыли или убыткам для кредитной организации.

Кредитный скоринг – это набор моделей принятия решений и лежащих в их основе методов, которые помогают кредиторам принять решение при предоставлении потребительского кредита. Эти методы помогают определить, кто получит кредит, в каком размере и какие операционные стратегии повысят прибыльность заемщиков для кредиторов. Использование кредитного скоринга увеличивает надежность оценки кредитоспособности человека, поскольку основано на реальных данных.

Кредитор обычно принимает два типа решений:

– следует ли предоставлять кредит новому заявителю;

– как поступить с существующими заявителями, в том числе увеличить ли их кредитные лимиты.

Процессов построения скоринговых моделей может быть множество, но главные этапы являются достаточно близкими по своей сути. Всегда необходимо производить обработку входных параметров. Согласно источнику [1] процедура построения скоринговой модели может выглядеть следующим образом:

1. Подготовка данных для построения скоринговой модели.
1.1. Анализ и предварительная обработка данных.
1.2. Определение зависимых переменных.
1.3. Определение независимых переменных.
1.4. Формирование обучающей и тестовой выборок.
1.5. Определение объема выборки.
2. Анализ и корректировка переменных для построения модели.
2.1. Корректировка распределения зависимой переменной.
2.2. Описательный анализ скоринговых переменных (для обнаружения ошибок).
2.3. Преобразование количественных переменных и порождение новых признаков.
2.4. Оценка мультиколлинеарности между количественными переменными.
2.5. Категоризация количественных переменных (биннинг).
2.6. Сегментация выборки.
2.7. Оценка взаимосвязи скоринговых переменных на вероятность дефолта.
3. Построение скоринговой карты.
3.1. Выбор модели (логистическая регрессия).
3.2. Включение независимых переменных в модель.
3.3. Выбор критерии качества модели логистической регрессии.
3.4. Перевод коэффициентов модели в скоринговую карту.

Обучающие данные (выборка) для примера кредитного скоринга являются реальными банковскими данными клиента, которые были систематизированы по очевидным причинам. Функции (характеристики в кредитном скоринге) состоят из двенадцати параметров. В данной случае целевой переменной является двоичная переменная со значеним «неудовлетворительно» или «удовлетворительно» (0 или 1) по отношению к клиенту с учетом финансового состояния в различные периоды его жизни.

Подготовка данных. При анализе, после получения данных, происходит замена редких и пропущенных значений отдельной категорией. После этого выбирается целевая функция и удаляются лишние колонки. Последним этапом подготовки данных является распределение данных на тренировочные и тестовые.

В данном случае использовался набор данных Home Equity (HMEQ [2]) содержит базовую информацию и информацию о выдачи кредитов для почти шести тысяч ссуд под недвижимость. Выбор пал именно на эти данные из-за их доступности, а также из-за того, что эти данные реального банк. Целевая переменная в этом наборе (BAD) это двоичная переменная, указывающая, нарушил ли заёмщик в конечном итоге свои финансовые обязательства или нет. Неблагоприятный исход произошел в 1189 случаях (приблизительно 20 %). Для каждого кандидата было записано 13 входных переменных:

BAD: принимает два параметра, где 0 кредит был возвращён:

– заёмщик допустил дефолт по кредиту или серьезно просрочил;
– LOAN: сумма кредита;
– MORTDUE: сумма к оплате по существующей ипотеке;
– VALUE: стоимость текущего имущества;
– REASON: содержит два параметра DebtCon – консолидация долга;
– homeImp – улучшение условий жизни;

– JOB: профессия;
– YOJ: время работы на текущей работе;
– DEROG: количество негативных характеристик заёмщика;
– DELINQ: количество просроченных кредитных линий;
– CLAGE: возраст самой давней кредитной истории в месяцах;
– NINQ: количество последних кредитных запросов;
– CLNO: количество уже имеющихся кредитов;
– DEBTINC: отношение размера кредита к доходу.

Нормализация входных данных. Далее следует избавиться от повторяющихся столбцов, и столбцов с высокой степенью корреляции:

hv.extension ('bokeh', 'matplotlib', logo = False).
df = pd.read_csv ('data_set.csv', low_memory = False).
df.drop ('DEBTINC', axis = 1, inplace = True).
df.dropna (axis = 0, how = 'any', inplace = True).

Теперь больше нет повторяющихся столбцов и нет столбцов с высокой степенью корреляции. Просматривая и анализируя данные, можно прийти к следующему заключению: значение столбца DEBTINC может быть удалено.

Модель на основе логистической регрессии

Логистическая регрессия – это простейшая линейная модель для классификации [3]. В этой модели вероятности, описывающие возможные результаты одного испытания, моделируются с использованием логистической функции. Задача оптимизации решается минимизацией функции стоимости (рис. 2.).

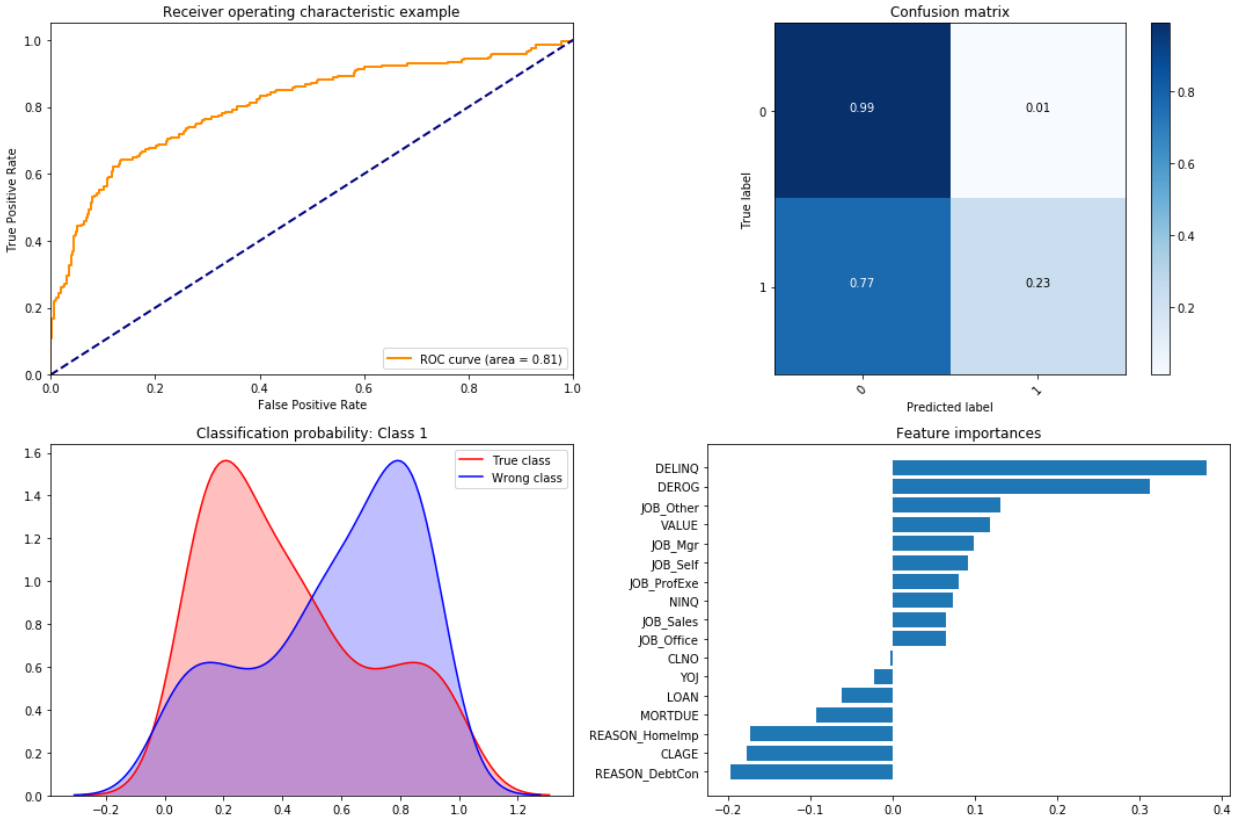

Рисунок 1. Результат для логистической модели

```
svc_xval_res = cross_validate_model(pipe, X, y, verbose=False)
svc_xval_res.T[['value','error']].style.format("{:.2f}")
```

	value	error
F1	0.37	0.09
Precision	0.77	0.16
Recall	0.25	0.07
Roc auc	0.79	0.05

Рисунок 2. Точность для логистической регрессии

Стохастический градиентный спуск

Стохастический градиентный спуск (SGD) (выполняющий итерацию градиента на отдельных примерах), который следует отличать от градиентного спуска. Градиент потерь оценивается для каждой выборки за раз и модель обновляется по мере обучения. Данный метод хорош для обучения, но плох для оптимизации: может потребоваться много итераций, чтобы свести к минимуму эмпирическую ошибку [4].

Логика визуализации такая же, как и для логистической модели, но результат иной, лучше, рисунок 3.

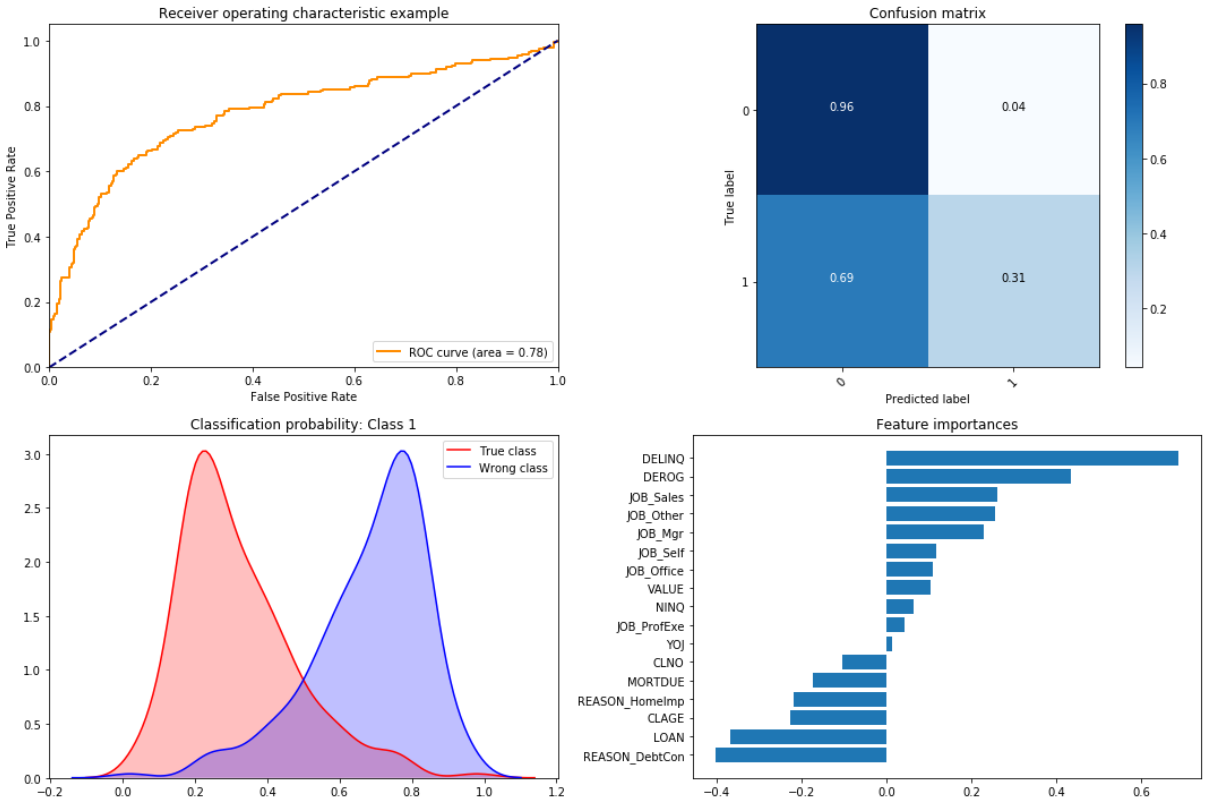

Рисунок 3. Результат для стохастического градиентного спуска

В итоге получаем, рисунок 4:

```
1  sgd_xval_res = cross_validate_model(pipe, X, y, verbose=False)
2  sgd_xval_res.T[['value','error']].style.format("{:.2f}")
```

	value	error
F1	0.42	0.09
Precision	0.71	0.11
Recall	0.31	0.09
Roc auc	0.76	0.06

Рисунок 4. Точность для стохастического градиентного спуска

Метод опорных векторов

Данный метод можно использовать для классификации, регрессии благодаря разделению пространства большой или бесконечной размерности на гиперплоскость или набора гиперплоскостей [5]. Хорошее разделение достигается благодаря гиперплоскости, которая имеет наибольшее расстояние до ближайших обучающих данных любого класса (функциональный запас). Считается, что чем больше этот запас, тем ниже ошибка обобщения классификатора.

Основные преимущества метода опорных векторов.

3. Эффективен в пространствах больших размеров.
4. Эффективен в случаях, когда количество измерений превышает количество образцов.
5. Эффективно использует память, т. к. в функции принятия решения используется подмножество обучающих точек.

Визуальный результат работы представлен на рис. 5, точность и ошибка на рис. 6:

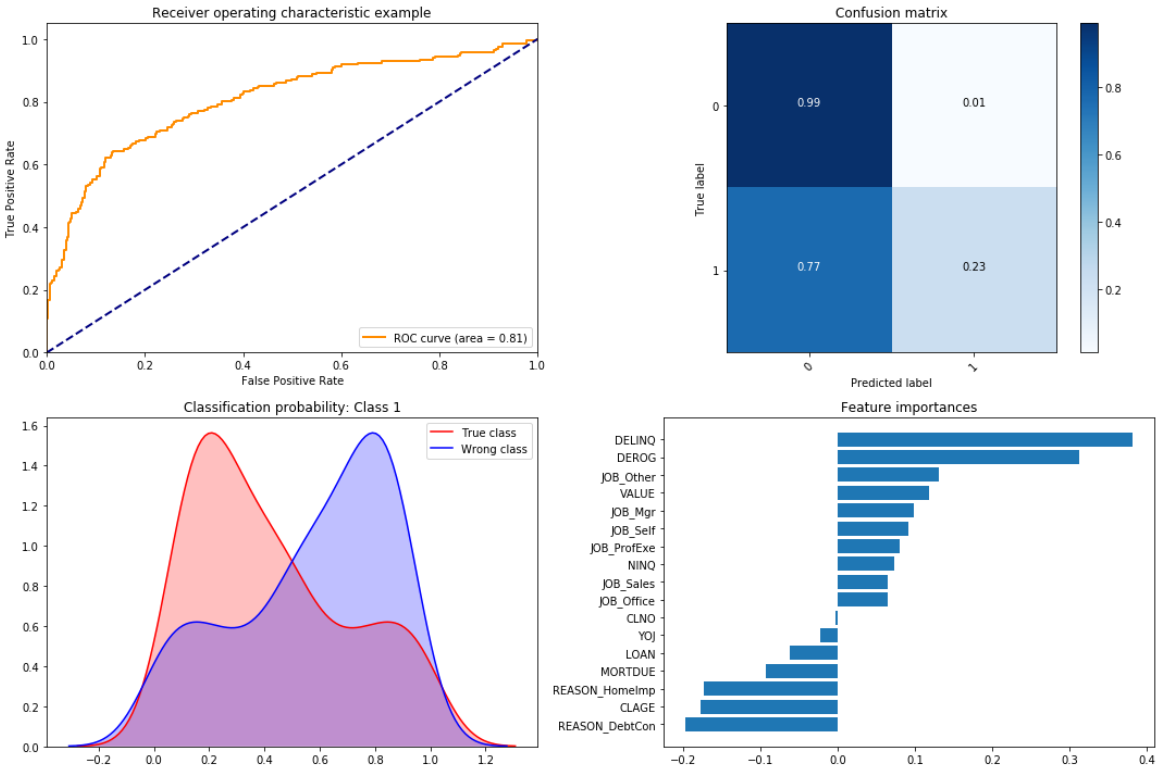

Рисунок 5. Результат для метода опорных векторов

```
1  svc_xval_res = cross_validate_model(pipe, X, y, verbose=False)
2  svc_xval_res.T[['value','error']].style.format("{:.2f}")
```

	value	error
F1	0.37	0.09
Precision	0.77	0.16
Recall	0.25	0.07
Roc auc	0.79	0.05

Рисунок 6. Точность для метода опорных векторов

Градиентный спуск

Данный метод строит модель поэтапно, как и другие аналогичные методы, и обобщает их, позволяя оптимизировать произвольную дифференцируемую функцию потерь [6]. Градиентный спуск можно использовать как для регрессионных, так и для классификационных задач.

Теперь можно визуализировать результат, рисунок 7, а на рисунке 8 показаны значение и ошибка, рассчитанные различными метриками.

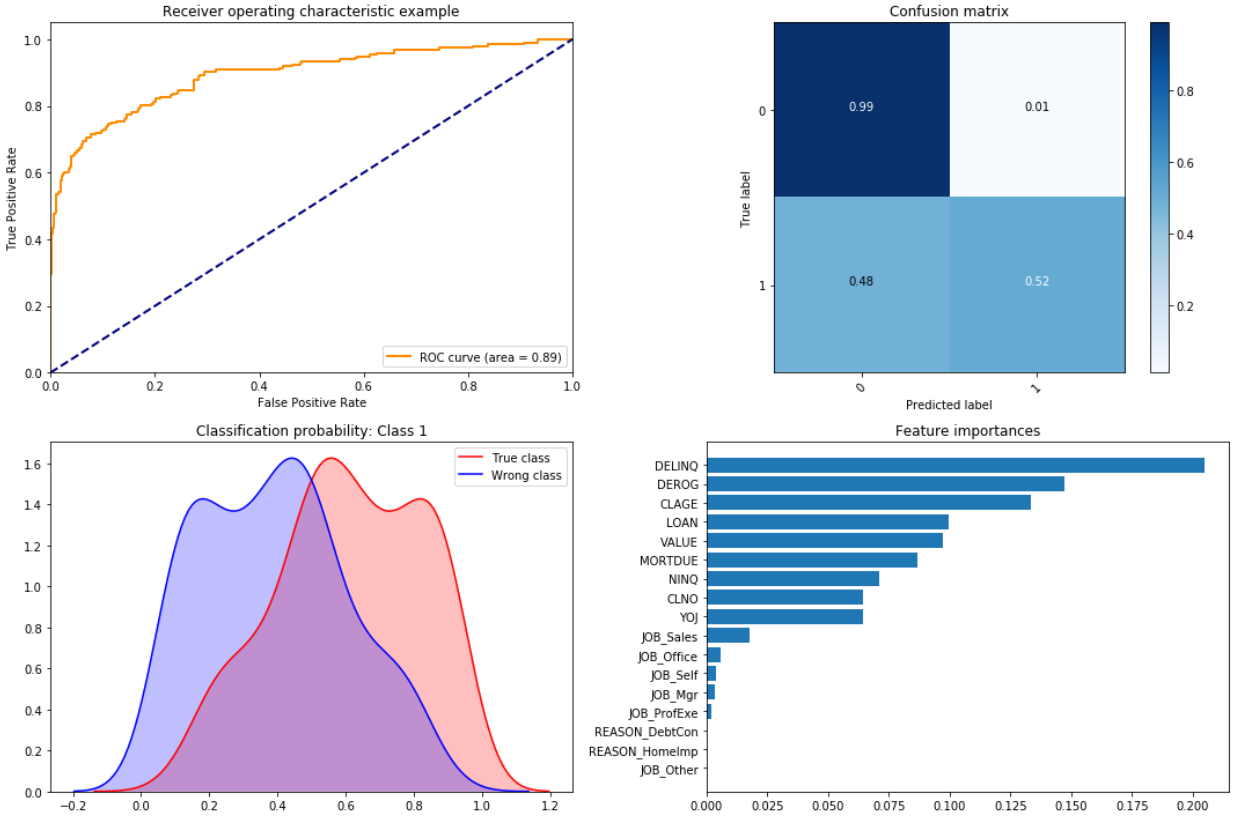

Рисунок 7. Результат для градиентного спуска

```
1  gbc_xval_res = cross_validate_model(pipe, X, y, verbose=False)
2  gbc_xval_res.T[['value','error']].style.format("{:.2f}")
```

	value	error
F1	0.36	0.17
Precision	0.46	0.30
Recall	0.38	0.21
Roc auc	0.56	0.24

Рисунок 8. Точность для градиентного спуска

Леса рандомизированных деревьев

Деревья решений (DT) – это непараметрический контролируемый метод обучения, используемый для классификации и регрессии. Цель состоит в том, чтобы создать модель, которая предсказывает значение целевой переменной, изучая простые правила принятия решений, выведенные из характеристик данных. Техника леса рандомизированных деревьев включает два алгоритма усреднения, основанных на рандомизированных деревьях решений: алгоритм RandomForest и метод Extra-Trees [7].

В случайных лесах каждое дерево строится из выборки, взятой с заменой (т. е. выборкой начальной загрузки) из обучающего набора. Кроме того, при разделении узла во время построения дерева выбранное разделение больше не является лучшим разделением среди всех функций. Вместо этого выбранное разбиение является лучшим разбиением среди случайного подмножества функций. В результате этой случайности, смещение леса обычно немного увеличивается (по сравнению с смещением одного неслучайного дерева), но из-за усреднения его дисперсия также уменьшается, обычно более чем компенсируя увеличение смещения, следовательно, дает в целом лучшую модель [8].

На рисунках 9 и 10 показаны результаты для алгоритма случайного леса (RandomForest).

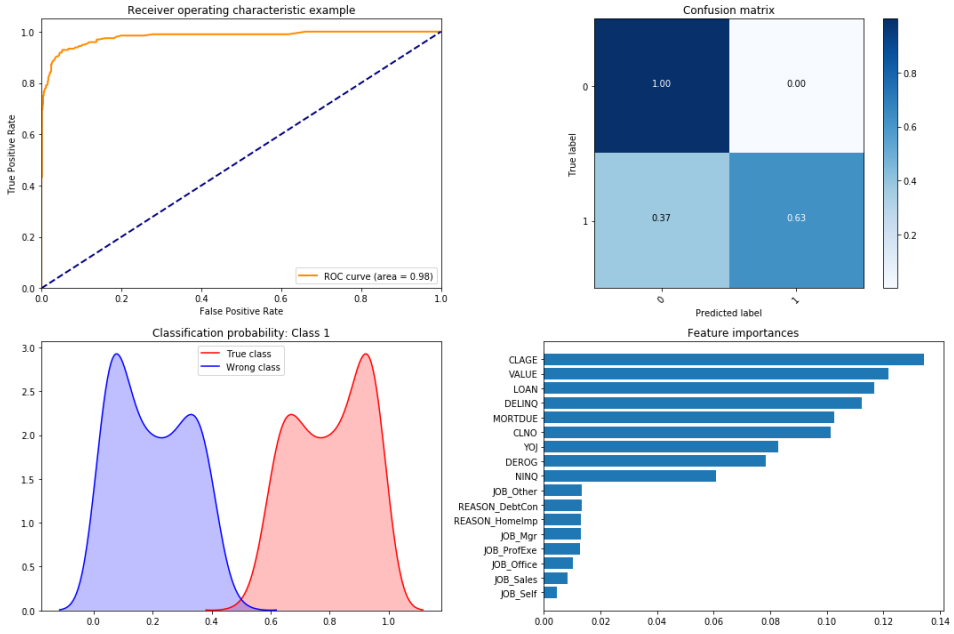

Рисунок 9. Результат для алгоритма случайного леса

```
1  rfc_xval_res = cross_validate_model(pipe, X, y, verbose=False)
2  rfc_xval_res.T[['value','error']].style.format("{:.2f}")
```

	value	error
F1	0.63	0.12
Precision	0.81	0.23
Recall	0.58	0.15
Roc auc	0.87	0.09

Рисунок 10. Точность для алгоритма случайного леса

Сильно рандомизированные деревья

В сильно рандомизированных деревьях случайность идет на один шаг дальше в способе вычисления разбиений. Как и в случайных лесах, используется случайное подмножество функций-кандидатов, но вместо поиска наиболее отличительных пороговых значений пороги выбираются случайным образом для каждой функции-кандидата, и в качестве правила разделения выбирается лучший из этих случайно сгенерированных пороговых значений. Обычно это позволяет немного уменьшить дисперсию модели за счет немного большего увеличения смещения (рис. 12.).

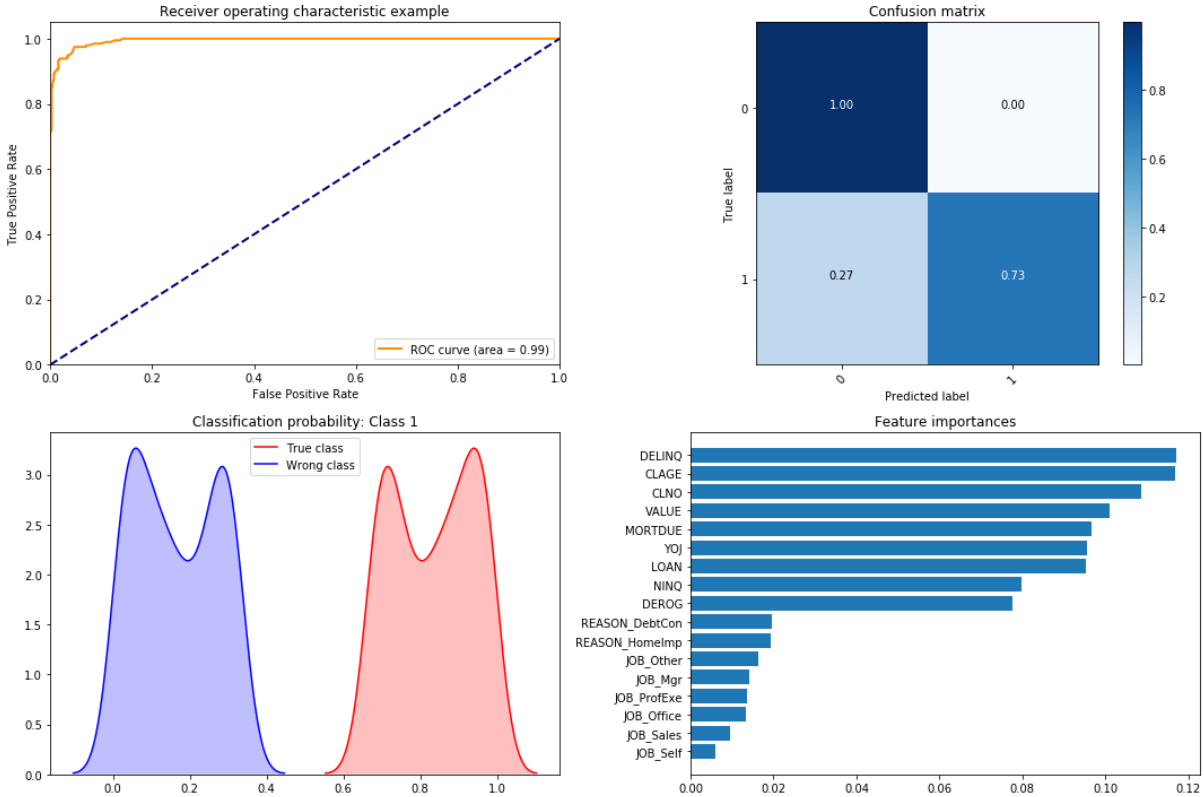

Рисунок 11. Результат для сильно рандомизированных деревьях

```
1  ert_xval_res = cross_validate_model(pipe, X, y, verbose=False)
2  ert_xval_res.T[['value','error']].style.format("{:.2f}")
```

	value	error
F1	0.75	0.12
Precision	0.91	0.16
Recall	0.66	0.14
Roc auc	0.96	0.06

Рисунок 12. Точность для сильно рандомизированных деревьях

Заключение. Сравнительный анализ моделей

В приведенной ниже таблице 1 представлены характеристики моделей классификации, которые были исследованы выше. Результаты упорядочиваются по убыванию значения F1. Таблица была получена так:

Наилучшие результаты дает сильно рандомизированное дерево, за которым следуют случайный лес и логистическая регрессия.

Таблица 1 – Результат работы каждой модели.

	F1	Precision	Recall	Roc auc
Model				
Extermely random tree classifier	0.75	0.91	0.66	0.96
Random forest classifier	0.63	0.81	0.58	0.87
Logistic regression	0.44	0.73	0.32	0.80
SGD classifier	0.42	0.71	0.31	0.76
Supporting vector classifier	0.37	0.77	0.25	0.79
Gradient boost classifier	0.36	0.46	0.38	0.56

Сильно рандомизированное дерево позволяет идентифицировать до 66 % ссуд, которые могут вызвать дефолт, при сохранении 91 % ссуд, которые будут выплачены вовремя. Значение ROC AUC достигает 96 %, что указывает на то, что вероятность того, что классификатор будет работать лучше при случайном выборе, составляет всего 4 %.

Список литературы

[1] Процедура построения скоринговой модели (скоринговой карты) [Электронный ресурс] – Режим доступа: http://www.machinelearning.ru/wiki/images/b/b5/ScoringANE(dragged).pdf – Дата доступа: 10.12.2020.

[2] Credit risk analytics [Electronic resource] – Mode of access: http://www.creditriskanalytics.net/datasets-private2.html – Date of access: 12.10.2020.

[3] Логистическая регрессия и ROC-анализ – математический аппарат [Электронный ресурс] – Режим доступа: https://loginom.ru/blog/logistic-regression-roc-auc – Дата доступа: 10.12.2020.

[4] Стохастический градиентный спуск [Электронный ресурс] – Режим доступа: https://ru.coursera.org/lecture/supervised-learning/stokhastichieskii-ghradiientnyi-spusk-xRY50 – Дата доступа: 11.01.2021.

[5] Введение в метод опорных векторов [Электронный ресурс] – Режим доступа: https://zen.yandex.ru/media/id/5e048b1b2b616900b081f1d9/vvedenie-v-metod-opornyh-vektorov-5fc7482df29188080efef35e – Дата доступа: 11.01.2021.

[6] Градиентный спуск: всё, что нужно знать [Электронный ресурс] – Режим доступа: https://neurohive.io/ru/osnovy-data-science/gradient-descent/ – Дата доступа: 12.01.2021.

[7] Жаров И. А., Верба В. А. Предсказания стоимости поездки на такси с помощью методов машинного обучения [Электронный ресурс] – Режим доступа: https://scienceforum.ru/2019/article/2018014019 – Дата доступа: 14.01.2021.

[8] Как работает случайный лес? [Электронный ресурс] – Режим доступа: https://zen.yandex.ru/media/nuancesprog/kak-rabotaet-sluchainyi-les-5eebb0479c2f793cb65c2fd1. – Дата доступа: 14.01.2021.

METHODS FOR CREDIT RISKS ASSESSMENT

D.V. SHYCHKOV
Master's student of the BSUIR,
software engineer SKB Radiotehproekt

I.I. FROLOV*PhD,*
Computer Science Department, BSUIR

Belarusian State University of Informatics and Radioelectronics, Republic of Belarus
OOO SKB Radiotehproekt, Republic of Belarus
E-mail: dmitry_shychkov@mail.ru

Abstract. This article explores supervised machine learning classifiers to predict lending outcomes. Given the large number of algorithms, the analysis starts with simple methods such as logistic regression, gradually increasing the complexity of the model to methods of randomized trees. Next, the performance of each model is compared and the classification solution most suitable for this lending problem is discussed. The following models were investigated and built:

1. logistic regression;
2. stochastic gradient descent;
3. support vector machine;
4. gradient descent;
5. randomized trees.

Keywords: credit scoring, credit risk, logistic regression, stochastic gradient descent, support vector machine, gradient descent, randomized trees.