ПРИКЛАДНЫЕ АСПЕКТЫ МАТЕМАТИКИ И ИНФОРМАТИКИ

УДК 004.89

ИНСТРУМЕНТАРИЙ И ОСОБЕННОСТИ РЕШЕНИЯ ЗАДАЧИ КЛАССИФИКАЦИИ В СИСТЕМАХ КРЕДИТНОГО СКОРИНГА

Андрей Анатольевич Гришин
к.экон.н., доцент
grishin-andrey@yandex.ru
г. Орел
Сергей Павлович Строев
к.экон.н., доцент
stroewsp@mail.ru
г. Орел

Орловский государственный университет им. И.С. Тургенева

Аннотация. В работе проведено исследование эффективности применения алгоритмов машинного обучения для решения задачи выявления недобросовестных клиентов финансовой организации. На реальных данных о клиентах показаны особенности реализации алгоритмов градиентного бустинга и нейронных сетей и вычислены параметры их эффективности. Показана практическая возможность применения данных алгоритмов в системах кредитного скоринга.

Ключевые слова: кредитный скоринг, машинное обучение, градиентный бустинг, нейронные сети.

В последние несколько лет кредитные организации активно развивают информационные системы управления кредитными портфелями. Одной из главных компонент такой системы являются системы прогнозирования дефолта клиента по своим обязательствам. Безусловно, процесс уменьшения доли просроченной задолженности является следствием не только эффективности банков, но и других процессов, например, чрезмерной перегруженности долгом заемщиков, списание части просроченной задолженности и т.д. Тем не менее, основным является развитие методов и систем управления в самих кредитных организациях.

Существуют различные способы построения моделей управления кредитными рисками [1]. В последние несколько лет все больше систем строятся на основе алгоритмов машинного обучения и искусственного интеллекта. В данной статье приводится вариант решения задачи прогнозирования дефолта клиента финансовой организации по кредитным обязательствам на основе нейронных сетей и деревьев решений.

В основу алгоритмов машинного обучения положен принцип построения алгоритма на основе имеющихся апостериорных данных (обучающая выборка) с последующим тестированием (тестовая выборка). В распоряжении авторов имелась обезличенная выборка по кредитам одного из банков, которые подкреплялись данными из бюро кредитных историй, объемом более 50 тысяч уникальных записей. О клиентах было известно: дата взятия и закрытия кредиты, объем денежных средств, тип кредита, набор полей, характеризующих дисциплину клиента, в том числе просрочки в 30, 60, 90 и более дней, количество просрочек, поля, характеризующие динамику погашения кредита. В целом, набор полей достаточно распространенный и исчерпывающий. Отдельным столбцом для каждого

клиента указывалось клиент добросовестный или нет, эти данные носили двоичный характер. Таким образом, решаемая задача относится к классической задаче бинарной классификации, где в качестве результата вычислялась вероятность попадания клиента в класс недобросовестных клиентов.

Таблица 1.

Значения оптимальных параметров для алгоритмов градиентного бустинга

Библиотека	**Значение метрики AUC**	**Параметр алгоритма**	**Значение**
XGBoost	0.7042	Количество потоков	'nthread':[4]
		Функция ошибки	'objective':['binary:logistic']
		Метрика, вычисляемая при обучении	'eval_metric':['auc']
		Размер шага, предотвращающий переобучение	'learning_rate':[0.04]
		Случайный подпространственный метод	'colsample_bytree':[0.8]
		Максимальная глубина дерева	'max_depth': [6]
		Соотношение подвыборок при обучении	'subsample':[0.6]
CatBoost	0.7104	Количество потоков	'thread_count':[4]
		Функция ошибки	'objective':['Logloss']
		Метрика, вычисляемая при обучении	'eval_metric':['AUC']
		Размер шага, предотвращающий переобучение	'learning_rate':[0.02]
		Тоже что и colsample_bytree в XGBoost	'rsm':[0.8]
		Максимальная глубина дерева	'max_depth':[6]

Решение с использованием алгоритмов градиентного бустинга. Основываясь на результатах работ [1,2], реализован алгоритм решения задачи с использованием градиентного бустинга. Так как необходимо было определить вероятность дефолта, то в качестве основной метрики использовали AUC, т.е. значение площади под кривой ROC – receiver operating characteristic, которая является рабочей характеристикой и дает наиболее полную оценку качества бинарной классификации. Для алгоритмов на основе деревьев решений этого достаточно. Программная реализация была выполнена на языке программирования Python с использованием библиотек XGBoost [5] и CatBoost [6], обучение производилось методом перекрестной проверки с количеством подвыборок равным 5. Сами выборки формировали при помощи функции StratifiedKFold() библиотеки Scikit-learn [5] для большей сбалансированности. В первом приближении, когда параметры алгоритмов, реализованных в применяемых библиотеках, не изменялись максимальное значение вероятности (метрики AUC) колебалось в районе 0.6 – 0.61, чего явно недостаточно, так как

мало отличается от 0.5, т.е. порогового значения метрики для задачи классификации при которой алгоритм не обладает сколь-нибудь значимой прогностической способностью.

Для повышения качества алгоритма реализована программа оптимизации параметров методом поиска по сетке с учетом возможностей регуляризации [4]. В результате были получены следующие значения, представленные в таблице 1.

Показательно (таблица 1), что помимо оптимальных характеристик алгоритмов, значения коэффициентов регуляризации – нулевые. Это свидетельствует о том, что сами алгоритмы – «жесткие», в смысле обладают малой вариативностью относительно построенного решения. Полученные значения вероятностей выше, чем в тестовом решении из-за меньшего значения шага, их можно охарактеризовать как удовлетворительные.

Изменяя шаг обучения алгоритмов, были получены значения точности прогноза и основной метрики, используемой при построении решения, результаты представлены в таблице 2.

Таблица 2.
Значение точности прогноза алгоритмов градиентного бустинга

№ п/п	Шаг модели	Значение метрики AUC		Точность модели (Accuracy)	
		XGBoost	CatBoost	XGBoost	CatBoost
1	0.05	0.7130	0.7022	0.8792	0.8800
2	0.04	0.7096	0.7042	0.8797	0.8807
3	0.03	0.7082	0.7057	0.8791	0.8807
4	0.02	0.7074	0.7104	0.8794	0.8802
5	0.01	0.7060	0.7097	0.8790	0.8798
6	0.005	0.7054	0.7038	0.8791	0.8792

Из таблицы 2 видно, что в решаемой задаче можно добиться наилучшего решения при уменьшении шага. Для каждой из применяемых библиотек он уникален, что отражает специфику построения градиентов каждого алгоритма. Так, для алгоритма, реализуемого в библиотеке XGBoost, оптимальным является шаг 0.04, а CatBoost приблизительно – 0.02. При уменьшении шага в случае использования XGBoost происходит переобучение алгоритма и ошибка начинает увеличиваться. Это продемонстрировано на рисунке 1. Показательно, что алгоритм градиентного бустинга, реализованный в библиотеке CatBoost, является наиболее устойчивым, но более ресурсоемким в плане вычислительных мощностей.

После того, как модель была построена и алгоритм обучен, необходимо проверить насколько хорошо он распознает объекты по совокупности критериев. Для оценки данного показателя используется матрица ошибок, суть которой заключается в определении ошибок ложной и истинной классификации. На практике [3] истинная положительная оценка (true positives rate – TPR) и истинная отрицательная оценка (false positives rate – FPR), определяются как функция порогового значения вероятности θ отнесения объекта к одному из классов:

$$TPR(\theta) = \frac{\text{экземпляр 1 классифицированный как 1}}{\text{экземпляр 1}}$$

$$FPR(\theta) = \frac{\text{экземпляр 1 классифицированный как 0}}{\text{экземпляр 0}}$$

Изменение ошибки обучения

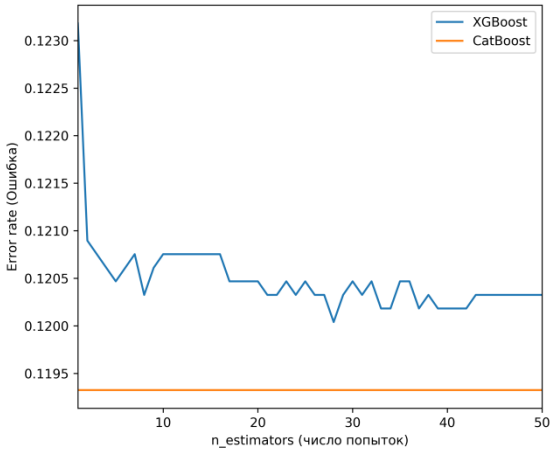
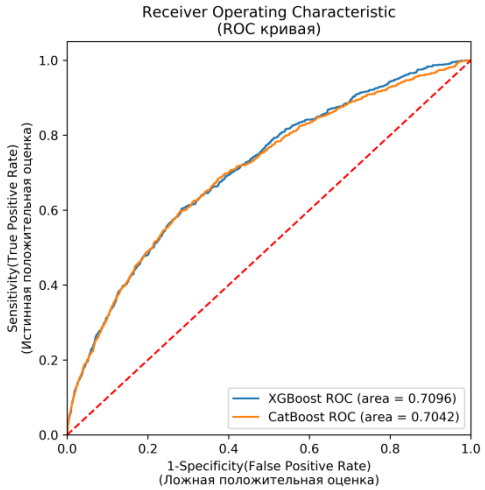
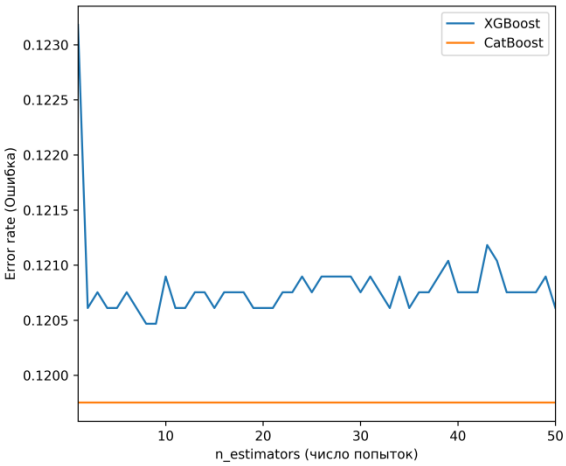
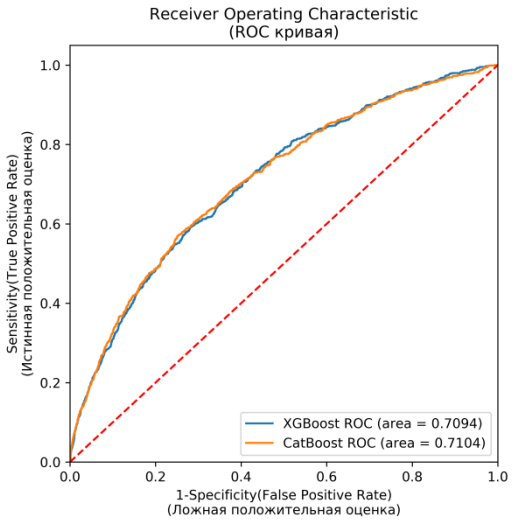


График ROC — кривой

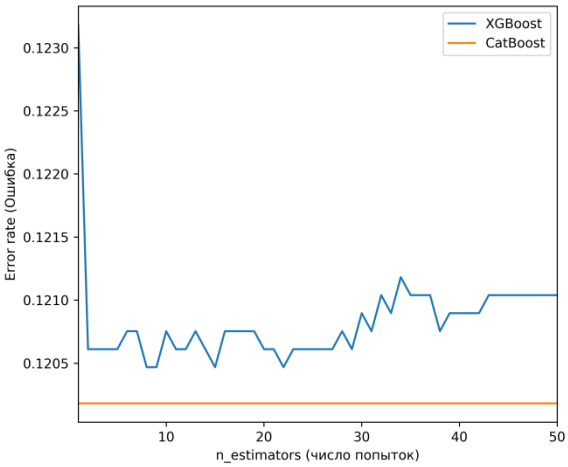


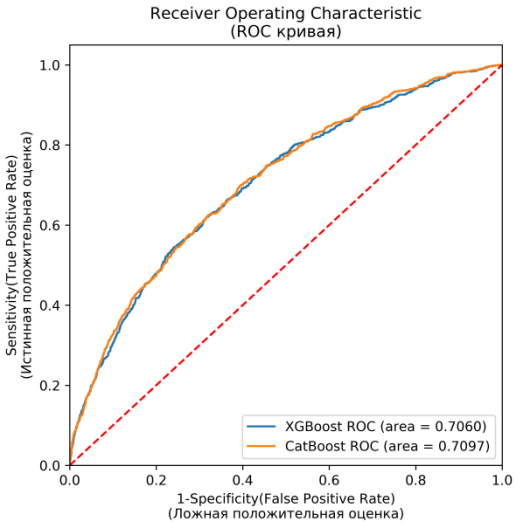
Шаг модели – 0.04





Шаг модели – 0.02





Шаг модели – 0.01

Рис. 1. Графики изменения ошибки обучения и ROC-кривой при различных значениях шага градиента

Параметр вероятности θ, если рассматривать его на плоскости, то он соответствует ROC-кривой. Так как данный показатель нами определен, то матрица ошибок может быть рассчитана:

Матрица ошибок без нормализации (Confusion matrix, without normalization)			
Истинный класс (True label)	0	True Positive Rate (TPR)	False Negative Rate (FNR)
	1	False Positive Rate (FPR)	True Negative Rate (TNR)
		0	1
		Предсказанный класс (Predicted label)	

Основные показатели точности и ошибки обучения, представленные на рисунке 1, рассчитывались соответственно:

$$Accuracy = \frac{TPR + TNR}{TPR + FPR + TNR + FNR}$$

$$ErrorRate = \frac{FPR + FNR}{TPR + FPR + TNR + FNR}$$

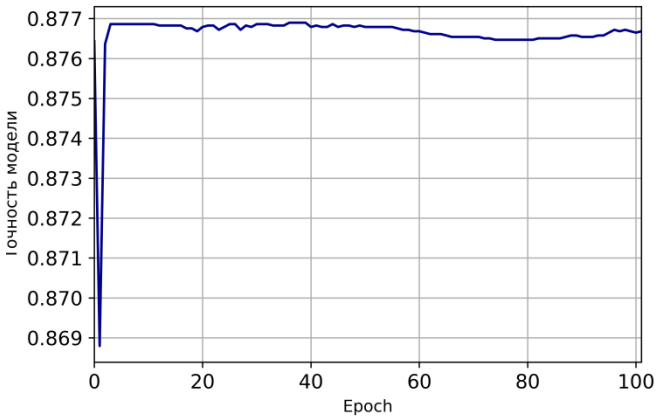
Матрицы ошибок для тестовой выборки в более чем 7000 объектов:

Матрица ошибок без нормализации (Confusion matrix, without normalization)			
Истинный класс (True label)	0	6153	7
	1	836	10
		0	1
		Предсказанный класс (Predicted label)	

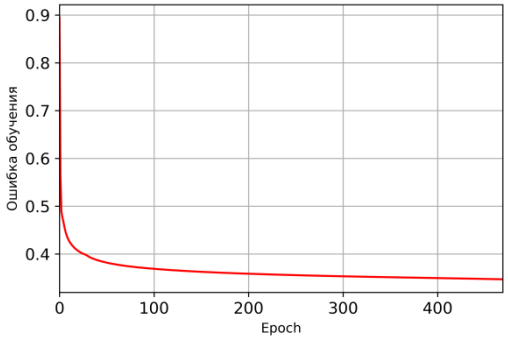
Видно, что один класс распознан практически полностью. В него входят клиенты, которые помечены как добросовестные. Точность распознавания равная приблизительно 88% составляет те объекты, которые были распознаны верно, т.е. 6153+10=6163 из 7006. Класс же клиентов с плохой историей платежей по кредитам не был распознан. Это является следствием большого разброса значений признаков, характеризующих дисциплину недобросовестных клиентов. Тем не менее, алгоритм может быть признан работоспособным, так как одну из подсовокупностей он определяет верно.

Для построения скоринговых систем, помимо алгоритмов на деревьях решений, применяют нейронные сети. Они реализованы в виде подключаемых библиотек ко многим языкам программирования. Среди самых распространенных реализаций наибольшую популярность имеют библиотеки TensorFlow, Theano, PyTorch, CNTK, MXNet. Для решения задачи использовали библиотеку TensorFlow в связке с Keras.

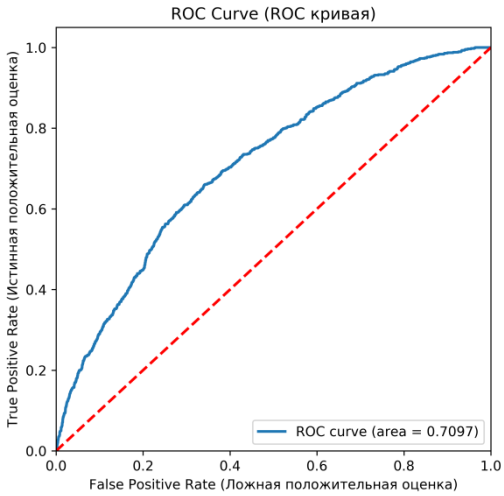
Проведя анализ различных топологий нейронных сетей с учетом невысокой размерности входного вектора данных, для решения задачи классификации наиболее подходящей является топология глубоких полносвязных сетей.



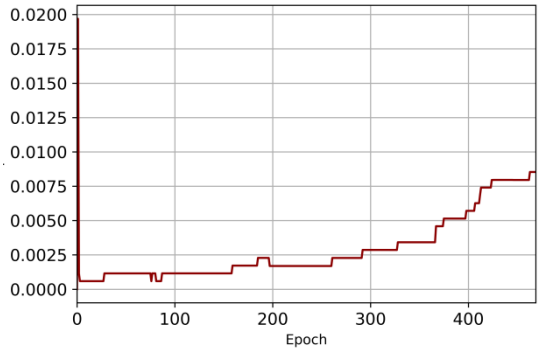
а) Изменение точности обучения



б) Изменение ошибки обучения



в) График ROC-кривой



г) График метрики F1-Score

Рис. 2. Графики, иллюстрирующие процесс обучения нейронной сети

В численном эксперименте для определения архитектуры сети также использовали поиск по сетке параметров, которая включала:

- методы оптимизации;
- методы инициализации весовых коэффициентов;
- функции активации нейронов в скрытом слое сети;
- функции активации последнего нейрона.

В результате проведенного исследования было выявлено 45 возможных архитектур, которые позволяют классифицировать объекты со значением AUC от 0.6043 до 0.6275. Максимальное значение соответствует архитектуре с двумя скрытыми слоями по 40 нейронов в каждом:

- метод оптимизации – adamax;
- метод инициализации весовых коэффициентов – normal;
- функция активации нейронов в скрытом слое сети – relu;
- функция активации последнего нейрона – linear.

Значение вероятности (AUC) недостаточно, даже в сравнении с алгоритмами градиентного бустинга, поэтому воспользуемся методами регуляризации сетей. Для этого, не меняя архитектуру сети, произвели поиск по сетке параметров:

- введем слой нормализации выборки (Batch Normalization) на входе сети;
- метод L1 – регуляризации;
- метод L2 – регуляризации;
- последовательное уменьшение шага обучения;
- метод замедления обучения сети.

В результате проведенных экспериментов значение удалось повысить AUC = 0.7097 при следующих значениях параметров регуляризации:

- коэффициент L1 – регуляризации – 0.001;
- коэффициент L2 – регуляризации – 0.001;
- последовательное уменьшение шага обучения – 0.015.

На рисунке 2 представлены результаты обучения нейронной сети.

В случае решения задачи классификации с использованием нейронной сети и, учитывая её вариативность (присутствуют методы регуляризации), дополнительно рассчитали изменение метрики F1-Score [8], которая показывает степень различимости классов и иногда применяется как альтернатива ROC-кривой. Из рисунка 2 видно, что вариативность, которая была добавлена методами регуляризации, приводит к ухудшению распознавания алгоритмом классов, это видно на рисунке 2 (г), где точность распознавания снижается при более чем 50 эпохах обучения. Тем не менее, значение метрики F1-Score не превышает 0.01. Значит, по крайней мере, один из классов распознан правильно.

Матрица ошибок для нейронной сети при той же тестовой выборке показывает следующие значения:

Матрица ошибок без нормализации (Confusion matrix, without normalization)			
Истинный класс (True label)	0	6156	4
	1	840	6
		0	1
		Предсказанный класс (Predicted label)	

Видно, что матрица ошибок в случае нейронной сети практически повторяет результаты, полученные с использованием алгоритмов градиентного бустинга. Точность распознавания составляет около 87%, причем класс, характеризующий дисциплину недобросовестных клиентов, не распознан.

Подводя итог проведенным исследованиям, можно сделать вывод, что современные алгоритмы, реализующие градиентный бустинг и нейронные сети, позволяют эффективно решать задачу распознавания недобросовестных клиентов в системах кредитного скоринга. В ходе исследований выявлено, что значимых преимуществ алгоритмы по сравнению друг с другом не имеют, т.к. имеют схожие результаты работы. В работе не приведены данные по скорости работы алгоритмов. Считаем, что при современном уровне развития электронных компонентов этот параметр не актуален. Используя GPU, библиотеки CatBoost и TensorFlow позволяют получать результат практически мгновенно, при этом нейронная сеть немного

быстрее. В ходе численных экспериментов при попытке улучшить показатели по распознаванию была проведена попытка построить более глубокие деревья (более 6) для алгоритма градиентного бустинга, результат не изменился. В случае построения более глубокой сети точность прогноза снижалась.

Список литературы

1. Гришин А.А., Строев С.П. Исследование эффективности алгоритмов классификации, использующих деревья решений, для экономического прогнозирования на основе апостериорных данных // Вестник технологического университета. 2018. Т. 21. № 9. С. 180–187.
2. Гришин А.А., Строев С.П. Разработка модели поведенческого скоринга с использованием методов градиентного бустинга // Научно-технический вестник Поволжья. 2018. № 9. С. 93–98.
3. Bialas P. A multi-instance deep neural network classifier: application to Higgs boson CP measurement [Электронный ресурс] // arXiv: 1803.00838v2. URL: https://arxiv.org/pdf/1803.00838.pdf.
4. Bühlmann P., Hothorn T. Boosting algorithms: Regularization, prediction and model fitting // Statistical Science. 2007. T. 22. № 4. Pp. 477–505.
5. Brownlee J. XGBoost With Python: Gradient Boosted Tree With XGBoost and scikit-learn. Machine Learning Mastery, 2017. 115 p.
6. Dorogush A.V., Ershov V., Gulin A. CatBoost: gradient boosting with categorical features support // Workshop on ML Systems at NIPS 2017. December 8, 2017.
7. Hossin M., Sulaiman M.N. A Review on Evaluation Metrics for Data Classification Evaluations: IJDKP // International Journal of Data Mining & Knowledge Management Process. 2015. T. 5. № 2. Pp. 1–11.
8. Lipton Z.C. Thresholding Classifiers to Maximize F1 Score [Электронный ресурс] // arXiv: 1402.1892v2. URL: https://arxiv.org/pdf/1402.1892.pdf.

TOOLS AND PECULIARITIES OF SOLVING THE CLASSIFICATION PROBLEM IN CREDIT SCORING SYSTEMS

A.A. Grishin
Cand. Sci. (Economics), Associate Professor
grishin-andrey@yandex.ru
Orel
S.P. Stroyev
Cand. Sci. (Economics), Associate Professor
stroewsp@mail.ru
Orel

Turgenev Orel State University

Abstract. The paper examines the efficiency of applying machine learning algorithms to the problem of identifying the probability of default for a client of a financial organization. The authors use the real client data to demonstrate the peculiarities of implementing gradient boosting and neural networks algorithms and calculate their efficiency parameters. The paper also shows the practical possibility of using these algorithms in credit scoring systems.

Keywords: credit scoring, machine learning, gradient boosting, neural networks.

References

1. Grishin, A.A., Stroyev, S.P. (2018). Efficiency Research for Classification Algorithms Using Decision Trees for Economic Forecasting Based on Posteriori Data [*Issledovaniye effektivnosti algoritmov klassifikatsii, ispol'zuyushchikh derev'ya resheniy, dlya ekonomicheskogo prognozirovaniya na osnove aposteriornykh dannykh*]. *Journal of the Technological University*. Vol. 21(9). Pp. 180–187.
2. Grishin, A.A., Stroyev, S.P. (2018). Development of a Behavioral Scoring Model Using Gradient Boosting Methods [*Razrabotka modeli povedencheskogo skoringa s ispol'zovaniyem metodov gradiyentnogo bustinga*]. *Scientific and Technical Journal of the Volga Region.* Vol. 9. Pp. 93–98.
3. Bialas, P. A. Multi-instance deep neural network classifier: application to Higgs boson CP measurement [Online Resource]. arXiv: 1803.00838v2. URL: https://arxiv.org/pdf/1803.00838.pdf.
4. Bühlmann, P., Hothorn, T. (2007). Boosting algorithms: regularization, prediction and model fitting. *Statistical Science*. Vol. 22 (4). Pp. 477–505.
5. Brownlee, J. (2017). *XGBoost With Python: Gradient Boosted Tree With XGBoost and scikit-learn*. Machine Learning Mastery.
6. Dorogush, A.V., Ershov, V., Gulin, A. (2017). CatBoost: gradient boosting with categorical features support. Workshop on ML Systems at NIPS 2017.
7. Hossin, M., Sulaiman, M.N. (2015). A Review on Evaluation Metrics for Data Classification Evaluations: IJDKP. *International Journal of Data Mining & Knowledge Management Process*. Vol. 5 (2). Pp. 1–11.
8. Lipton, Z.C. Thresholding Classifiers to Maximize F1 Score [Online Resource]. arXiv: 1402.1892v2. URL: https://arxiv.org/pdf/1402.1892.pdf

УДК 519.652

«ЗЕРКАЛЬНЫЙ» МЕТОД РАСЧЕТА СПЛАЙН КРИВОЙ

Анастасия Валерьевна Журавлева
студент
anastasiya_zhuravleva98@mail.ru
г. Краснодар
Кубанский государственный университет

Алдияр Бексеитович Даулеткулов
учитель физики и информатики
aldauletkulov@ mail.ru
г. Алматы
Школа ОРФО «БЭСТ»

Аннотация. Авторами статьи предложен «зеркальный» метод расчета сплайн кривой, в котором учитываются значения функций $f(x_{i-1})$, $f(x_i)$, $f(x_{i+1})$. Использование этих значений приводит к «физически» более правильному «отражению» интерполяционной кривой в области опорных точек, появляется возможность избежать ложных максимумов и минимумов. Проверка данного метода показала, что появляется возможность его применения как для интерполяции «простых» функций, так и для «тяжелых» для интерполяции экспериментальных данных, а это значит, что «зеркальный» метод можно назвать универсальным. «Зеркальный» метод устойчив, это говорит о том, что результаты не зависят от направления вычислений, имеет радиус локальности 1. В статье показано, что формула для расчета «центральных» разностей (метода