УДК 004.9:331.44

ИССЛЕДОВАНИЕ БАЛАНСИРОВКИ НА РЕАЛЬНЫХ ДАННЫХ

Потёмин И.В., студент

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Голда О. – канд. экон. наук, доцент

Аннотация. В работе исследуется влияние дисбаланса классов на эффективность алгоритмов машинного обучения. На примере датасета прогнозирования профессионального выгорания проведено сравнение трёх методов балансировки: передискретизации, недодискретизации и взвешивания классов. Экспериментально оценены логистическая регрессия, дерево решений и метод опорных векторов. Установлено, что игнорирование дисбаланса приводит к неспособности моделей выявлять минорный класс при высокой общей точности. Наиболее универсальным решением признано взвешивание классов, обеспечивающее высокую сбалансированную точность и PR-AUC без изменения объёма данных. Результаты формируют рекомендации по выбору стратегии балансировки для задач с неравномерным распределением классов.

Ключевые слова. Машинное обучение, дисбаланс классов, балансировка данных, передискретизация, недодискретизация, взвешивание классов, бинарная классификация, логистическая регрессия, дерево решений, метод опорных векторов, сбалансированная точность, PR-AUC.

В современных задачах машинного обучения дисбаланс классов снижает эффективность моделей, так как стандартные алгоритмы оптимизируют общую функцию потерь, игнорируя минорный класс [1]. Цель работы – оценка методов балансировки данных: передискретизации, недодискретизации и взвешивания классов. Эксперимент проведен на датасете прогнозирования выгорания с использованием логистической регрессии, дерева решений и метода опорных векторов. Оценка качества выполнялась по сбалансированной точности и PR-AUC. Результаты формируют рекомендации по выбору стратегии балансировки для задач, где цена пропуска целевого события превышает стоимость ложной тревоги.

В качестве экспериментальных данных был использован датасет, моделирующий информацию о сотрудниках организации для задачи бинарной классификации наличия профессионального выгорания. Набор данных включает в себя демографические признаки, такие как возраст и пол, профессиональные характеристики, включая должность и стаж работы, а также психофизиологические показатели: уровень стресса, удовлетворенность работой, количество рабочих часов в неделю и долю удаленной работы.

Целевой переменной выступал бинарный признак Burnout, указывающий на наличие или отсутствие выгорания. На этапе предобработки данных первоначально была выполнена проверка на наличие пропущенных значений с помощью визуализации heatmap и были удалены пропуски в датасете.

После проверки из набора данных был удален столбец Name, не несущий смысловой нагрузки для моделирования. Для категориальных признаков Gender и JobRole было применено кодирование методом One-Hot Encoding, что позволило преобразовать текстовые метки в числовой формат, пригодный для обработки алгоритмами машинного обучения. Подготовленная выборка была разделена на обучающую и тестовую части в пропорции 80/20 с фиксацией параметра random_state равного 42. Для сохранения исходного распределения целевого класса в обеих подвыборках и предотвращения ситуации, когда в одной из них может оказаться недостаточное количество объектов минорного класса, разделение проводилось со стратификацией (рисунок 1).

```
# разделение на тестовую и тренировчную выборку
target_col = 'Burnout'

X = df.drop(columns=[target_col])
y = df[target_col]

X = df.drop(columns=[target_col])

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify = y)
```

Рисунок 1 – Разделение выборки на тренировочную и тестовую со стратификацией

Числовые переменные были подвергнуты стандартизации с использованием StandardScaler. Параметры масштабирования (среднее и стандартное отклонение) вычислялись исключительно на обучающей выборке (fit), после чего полученные трансформации применялись к тестовой выборке

(transform). Это гарантирует, что модель не получает косвенной информации о статистических характеристиках тестовых данных в процессе обучения (рисунок 2).

```
# OHE и стандартизация
df = pd.get_dummies(df, columns=["Gender", "JobRole"])
numerical_cols = ['Age', 'Experience', 'WorkHoursPerWeek', 'RemoteRatio', 'SatisfactionLevel', 'StressLevel']

scaler = StandardScaler()
df[numerical_cols] = scaler.fit_transform(df[numerical_cols])

df.to_csv("_synthetic_employee_burnout.csv")
```

```
df.to_csv("_synthetic_employee_burnout.csv")
df.head()
```

	Age	Experience	WorkHoursPerWeek	RemoteRatio	SatisfactionLevel	StressLevel	Burnout	Gender_Female	Ge
0	-0.770520	-0.773509	0.880175	-0.994132	1.216101	-1.538798	0	False	
1	-0.061548	-0.117483	-0.218776	0.584237	-0.783652	-1.191597	0	True	
2	-0.681899	-0.882847	-0.472380	-1.028445	-0.359462	-0.844395	0	True	
3	-0.504656	-0.445496	-0.472380	0.687174	0.203239	0.891614	0	True	
4	1.622262	-0.226821	-0.979588	-0.136323	1.224758	-1.538798	0	False	

Next steps: New interactive sheet

Рисунок 2 – Пример данных после кодирования и стандартизации

Подготовленная выборка была разделена на обучающую и тестовую части в пропорции восемьдесят к двадцати с фиксацией параметра random_state равного сорока двум для обеспечения воспроизводимости результатов эксперимента. Для решения проблемы дисбаланса классов в обучающей выборке были реализованы три подхода, первый из которых RandomOverSampler, предполагал искусственное увеличение количества объектов меньшего класса путем дублирования существующих примеров, второй метод RandomUnderSampler, заключался в случайном удалении части объектов большего класса до достижения баланса, и третий подход предусматривал использование встроенного механизма взвешивания классов, параметр class_weight равный balanced, который автоматически корректирует функцию потерь алгоритма в зависимости от частоты классов.

В качестве базовых алгоритмов классификации для сравнения эффективности методов балансировки были выбраны три разнородные модели: логистическая регрессия LogisticRegression, выступающая в качестве линейного базового классификатора, дерево решений DecisionTreeClassifier, представляющее нелинейный алгоритм, основанный на правилах, и метод опорных векторов SVC с радиальным базисным ядром RBF, способный строить сложные разделяющие поверхности (таблица 1).

Таблица 1 – Гиперпараметры используемых моделей

Модель	Гиперпараметры
LogisticRegression	C=0.7, max_iter=500, random_state=42
DecisionTreeClassifier	max_depth=3, min_samples_leaf=50, random_state=42
SVC	kernel='rbf', C=0.8, gamma='scale', probability=True

Обучение и оценка моделей проводились с использованием стандартных метрик качества, включая общую точность, сбалансированную точность, матрицу ошибок и площадь под PR кривой, что позволило комплексно оценить влияние методов балансировки на способность моделей выявлять меньший класс.

Для каждой из трех выбранных моделей машинного обучения – логистической регрессии, дерева решений и метода опорных векторов – было проведено обучение в четырех вариантах: базовая модель без коррекции дисбаланса, модель с применением RandomOverSampler меньшего класса, модель с использованием RandomUnderSampler большего класса и модель с назначением сбалансированных весов классов через параметр class_weight='balanced' [2]. Все эксперименты выполнялись на идентичных обучающей и тестовой выборках, сформированных в пропорции восемьдесят к двадцати с фиксацией параметра random_state равного сорока двум для гарантии воспроизводимости результатов (рисунок 3).

```
print("================ БОРЬБА С ДИСБАЛАНСОМ =================")

ros = RandomOverSampler(random_state=42)
X_ros, y_ros = ros.fit_resample(X_train, y_train)
rus = RandomUnderSampler(random_state=42)
X_rus, y_rus = rus.fit_resample(X_train, y_train)

print("Размеры до:", X_train.shape, y_train.shape)
print("Размеры после oversampling:", X_ros.shape, y_ros.shape)
print("Размеры после undersampling:", X_rus.shape, y_rus.shape)
```

```
================ БОРЬБА С ДИСБАЛАНСОМ =================
Размеры до: (1600, 13) (1600,)
Размеры после oversampling: (3006, 13) (3006,)
Размеры после undersampling: (194, 13) (194,)
```

Рисунок 3 – Размеры выборок до и после применения методов балансировки

Оценка качества моделей проводилась с использованием комплекса метрик, позволяющих всесторонне проанализировать эффективность классификации в условиях дисбаланса [3]. Основным критерием сравнения служила сбалансированная точность Balanced Accuracy, учитывающая долю верно классифицированных объектов в каждом классе независимо от их количества. Дополнительно для детального анализа качества предсказаний по каждому классу использовались метрики Precision, Recall и F1-score, представленные в классификационном отчете classification report, что позволяло оценить компромисс между точностью положительных предсказаний и полнотой выявления целевого класса (рисунок 4).

```
=== Logistic Regression (class_weight=balanced) ===
Accuracy: 0.9475
Balanced accuracy: 0.9714673913043479
              precision    recall  f1-score   support

           0       1.00      0.94      0.97       368
           1       0.60      1.00      0.75        32

    accuracy                           0.95       400
   macro avg       0.80      0.97      0.86       400
weighted avg       0.97      0.95      0.95       400

=== Logistic Regression (Oversampling) ===
Accuracy: 0.955
Balanced accuracy: 0.9755434782608696
              precision    recall  f1-score   support

           0       1.00      0.95      0.97       368
           1       0.64      1.00      0.78        32

    accuracy                           0.95       400
   macro avg       0.82      0.98      0.88       400
weighted avg       0.97      0.95      0.96       400

=== Logistic Regression ===
Accuracy: 0.9475
Balanced accuracy: 0.6861413043478262
Classification Report:
              precision    recall  f1-score   support

           0       0.95      1.00      0.97       368
           1       0.92      0.38      0.53        32

    accuracy                           0.95       400
   macro avg       0.94      0.69      0.75       400
weighted avg       0.95      0.95      0.94       400

=== Logistic Regression (Undersampling) ===
Accuracy: 0.93
Balanced accuracy: 0.9619565217391304
              precision    recall  f1-score   support

           0       1.00      0.92      0.96       368
           1       0.53      1.00      0.70        32

    accuracy                           0.93       400
   macro avg       0.77      0.96      0.83       400
weighted avg       0.96      0.93      0.94       400
```

Рисунок 4 – classification_report для логистической регрессии

Для оценки способности модели ранжировать объекты по вероятности принадлежности к минорному классу рассчитывалась площадь под PR-кривой, которая является чувствительной метрикой к качеству классификации целевого класса в условиях дисбаланса и позволяет сравнивать модели на основе компромисса между точностью предсказаний (precision) и полнотой выявления (recall) независимо от выбранного порога классификации (рисунок 5).

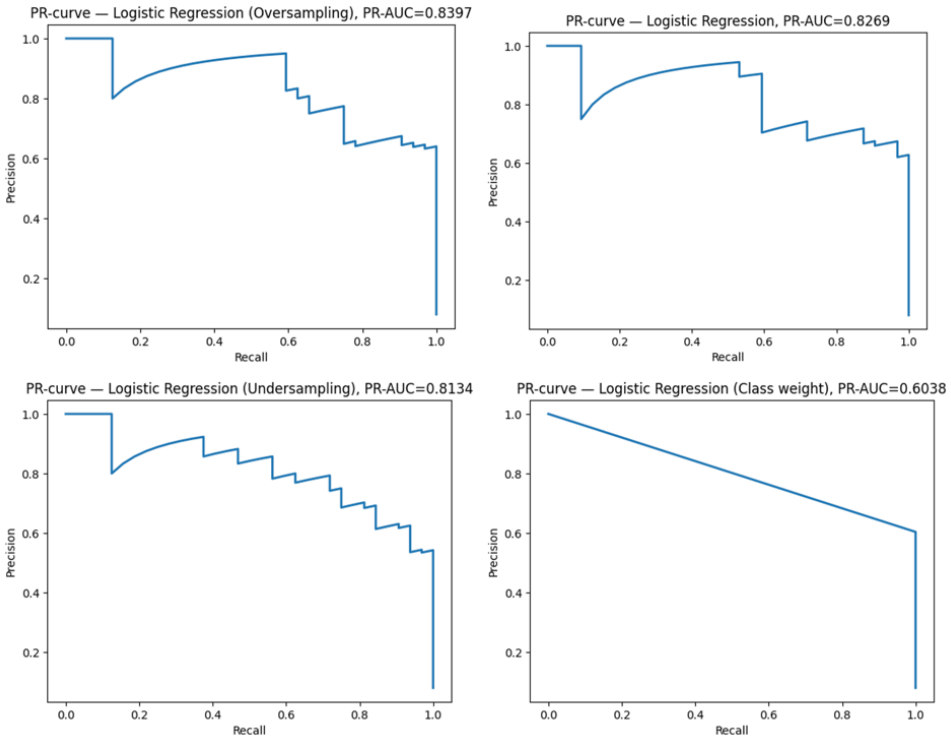

Рисунок 5 – PR-кривые для логистической регрессии

Визуализация результатов включала построение матриц ошибок Confusion Matrix для наглядного представления структуры ошибок модели, в частности соотношения ложноотрицательных и ложноположительных предсказаний, что имеет критическое значение для задач, где цена пропуска целевого события существенно превышает стоимость ложной тревоги (рисунок 6).

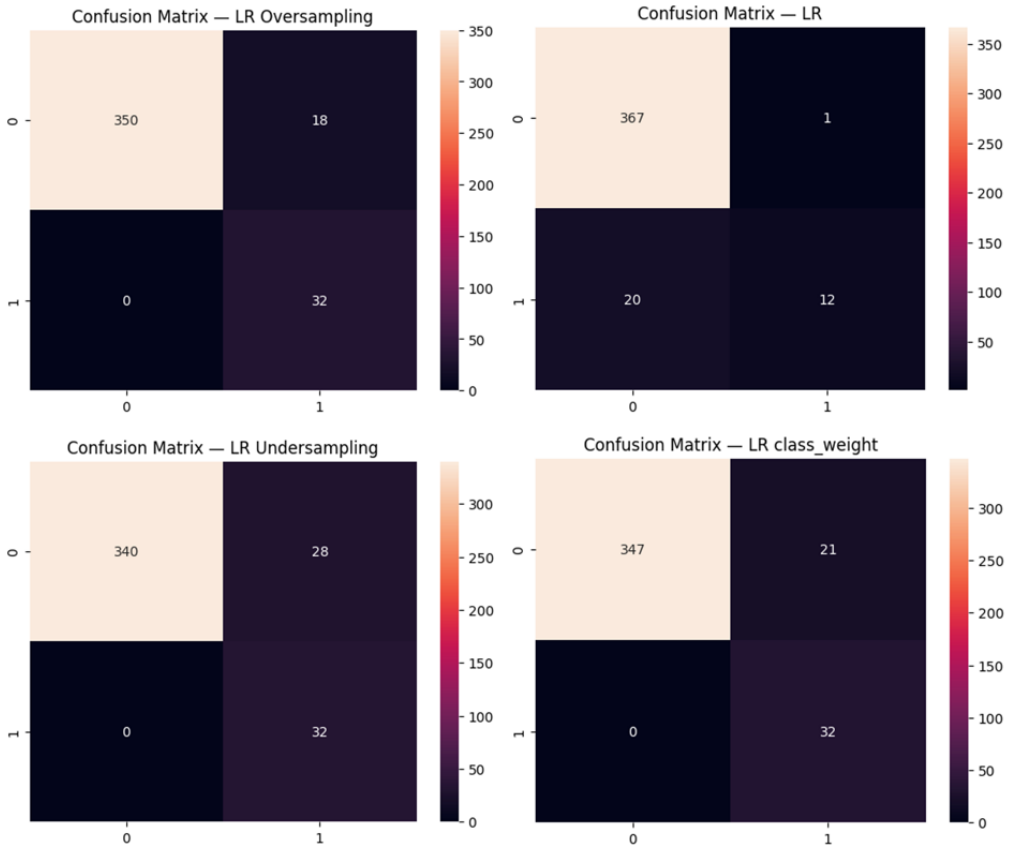

Рисунок 6 – Матрица ошибок для логистической регрессии

Все вычисления выполнялись с использованием библиотеки scikit-learn и импортированных функций balanced_accuracy_score, classification_report, confusion_matrix, pr_curve и auc, а методы ресемплинга реализовывались через модуль imblearn.over_sampling и imblearn.under_sampling. Полученные в ходе экспериментов количественные показатели и графические материалы легли в основу сравнительного анализа эффективности различных стратегий борьбы с дисбалансом данных.

Базовые модели всех трех алгоритмов, обученные на исходных несбалансированных данных, продемонстрировали схожую проблему смещения в сторону большего класса: логистическая регрессия и метод опорных векторов показали идентичную сбалансированную точность 0.686 при общей точности около 0.9475, а дерево решений оказалось наиболее уязвимым к дисбалансу с сбалансированной точностью 0.50 и полной неспособностью выявить случаи выгорания. Эти результаты подтверждают, что игнорирование проблемы дисбаланса делает модели непригодными для практического применения независимо от типа алгоритма, так как все они оптимизируют общую функцию потерь в ущерб выявлению меньшего класса (рисунок 7).

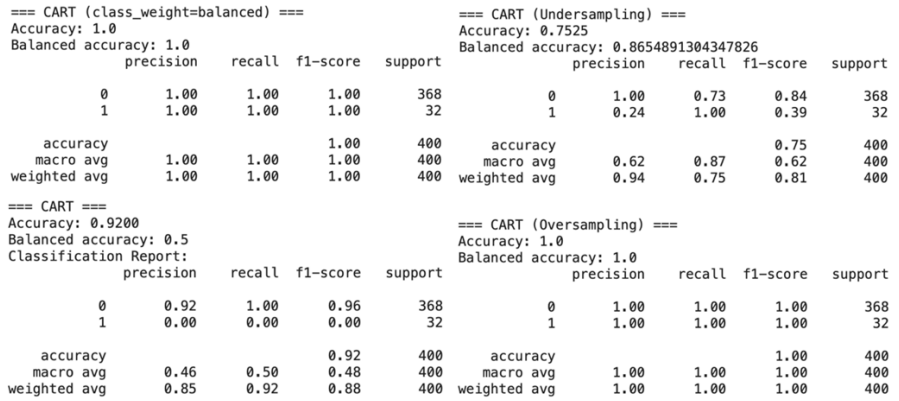

Рисунок 7 – classification_report для CART

Применение методов балансировки кардинально изменило качество всех моделей, однако эффективность конкретных подходов оказалась различной для разных алгоритмов. Метод случайной передискретизации продемонстрировал наилучшие результаты для метода опорных векторов, обеспечив общую точность 0.9825, сбалансированную точность 0.962 и precision предсказаний меньшего класса 0.86 при recall 0.94, что является оптимальным балансом между выявлением случаев выгорания и минимизацией ложных срабатываний.

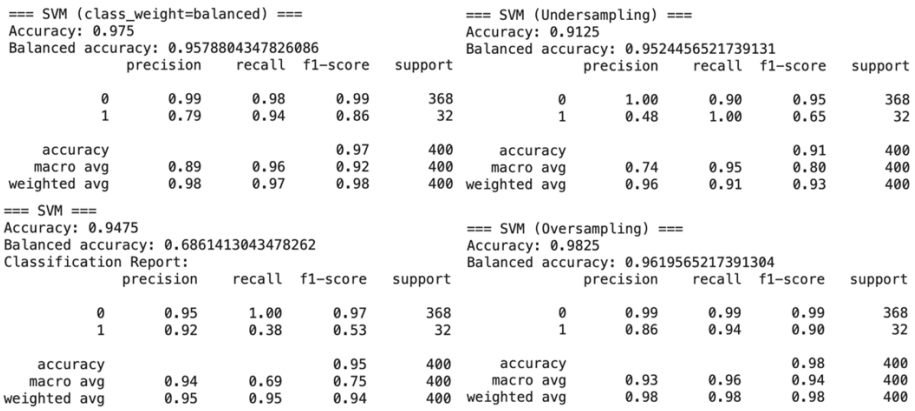

Рисунок 8 – classification_report для SVM

Для дерева решений передискретизация привела к идеальной классификации с метриками 1.00 по всем показателям, однако столь высокие результаты могут свидетельствовать о переобучении модели на датасете, особенно учитывая дублирование объектов меньшего класса. Логистическая регрессия с передискретизацией показала recall 1.00 для класса выгорания при precision 0.64 (таблица 2).

Метод взвешивания классов продемонстрировал устойчивость и эффективность среди всех алгоритмов: для метода опорных векторов он обеспечил точность 0.975, сбалансированную точность 0.958 и precision меньшего класса 0.79 при recall 0.94, что лишь незначительно уступает передискретизации, но не требует изменения объема данных. Для дерева решений взвешивание

классов также привело к идеальной классификации с метриками 1.00, а для логистической регрессии обеспечило показатели, сопоставимые с передискретизацией (recall 1.00, precision 0.60). Это делает метод взвешивания классов наиболее универсальным и вычислительно эффективным подходом, особенно для больших датасетов, где изменение объема данных может быть критичным.

Таблица 2 – Сравнение метрик качества моделей при различных методах балансировки

Модель	Метод	Balanced accuracy	Precision	Recall	PR-AUC
Логистическая регрессия	Oversampling	0.9755	0.64	1	0.8397
	Undersampling	0.9620	0.53	1	0.8134
	Без балансировки	0.6891	0.92	0.38	0.8269
	Class weight	0.9715	0.60	1	0.6038
CART	Oversampling	1	1	1	1
	Undersampling	0.8655	0.24	1	0.3148
	Без балансировки	0.5	0	0	0.5161
	Class weight	1	1	1	1
SVM	Oversampling	0.9620	0.86	0.94	0.8086
	Undersampling	0.9524	0.48	1	0.4776
	Без балансировки	0.6861	0.92	0.38	0.3962
	Class weight	0.9579	0.79	0.94	0.7451

Метод случайной недодискретизации показал наименее стабильные результаты: для метода опорных векторов он обеспечил recall 1.00 при precision меньшего класса 0.48 и общей точности 0.9125, для дерева решений precision упала до 0.24 при общей точности 0.7525, а для логистической регрессии составила 0.53 при recall 1.00. Это подтверждает гипотезу о том, что удаление части данных большего класса приводит к потере информативных закономерностей и ухудшению обобщающей способности моделей, особенно для сложных нелинейных алгоритмов.

Сравнение алгоритмов выявило, что метод опорных векторов с радиальным базисным ядром продемонстрировал наилучшую устойчивость к дисбалансу и наиболее сбалансированные результаты при применении техник балансировки, обеспечивая высокую precision предсказаний меньшего класса без чрезмерного количества ложных срабатываний. Дерево решений показало наибольшую чувствительность к методам балансировки с переходом от полного провала (balanced accuracy 0.50) до идеальной классификации (balanced accuracy 1.00), что указывает на его высокую гибкость, но также и на риск переобучения. Логистическая регрессия продемонстрировала стабильные предсказуемые результаты с recall 1.00 для выявления выгорания при всех методах балансировки.

С практической точки зрения, для задач прогнозирования профессионального выгорания, где цена пропуска сотрудника в группе риска существенно превышает стоимость ложной тревоги, приоритетными являются конфигурации, обеспечивающие максимальный recall выявления меньшего класса. Метод опорных векторов с передискретизацией или взвешиванием классов представляет собой оптимальный выбор, так как обеспечивает recall 0.94 при высокой precision 0.86 и 0.79 соответственно, минимизируя количество ложных срабатываний, которые могут привести к неэффективному расходованию ресурсов отдела кадров. Для логистической регрессии предпочтительным является использование взвешивания классов как вычислительно эффективного метода, обеспечивающего recall 1.00. Дерево решений с методами передискретизации или взвешивания классов может быть использовано для интерпретируемого анализа факторов выгорания, однако требует дополнительной валидации на независимых выборках для исключения переобучения.

В ходе исследования подтверждена критическая важность применения методов балансировки данных при решении задач классификации с дисбалансом классов. Базовые модели всех алгоритмов без коррекции показали низкую способность выявлять меньший класс: сбалансированная точность

составила 0.686 для логистической регрессии и SVM, и 0.50 для дерева решений, при recall класса выгорания 0.00 для CART.

Наилучшей конфигурацией для задачи прогнозирования выгорания признан метод опорных векторов с передискретизацией (accuracy 0.9825, balanced accuracy 0.962, precision 0.86, recall 0.94). Метод взвешивания классов показал сопоставимые результаты при меньших вычислительных затратах, что делает его универсальным решением. Дерево решений с балансировкой продемонстрировало идеальные метрики 1.00, однако требует валидации на независимых данных из-за риска переобучения. Недодискретизация показала наименее стабильные результаты со значительным снижением precision.

Практическая рекомендация: для задач HR-аналитики, где цена пропуска целевого события высока, следует использовать модели с максимальным recall меньшего класса и оценивать качество по сбалансированной точности и PR-AUC, а не по общей точности. Оптимальной конфигурацией для внедрения является SVM с class_weight='balanced'.

Список использованных источников:

1. Вьюгин, В. Математические основы машинного обучения и прогнозирования / В. Вьюгин. – Москва : МЦНМО, 2014. – 218 с.

2. Флах, П. Машинное обучение : наука и искусство построения алгоритмов, которые извлекают знания из данных / П. Флах. – Москва : ДМК-Пресс, 2012. – 412 с.

3. Силен, Д. Основы Data Science и Big Data. Python и наука о данных / Д. Силен, А. Мейсман, М. Али. – Санкт-Петербург : Питер, 2017. – 336 с.

4. Гудфеллоу, Я. Глубокое обучение / Я. Гудфеллоу, И. Бенджио, А. Курвилль. – 2-е изд., испр. – Москва : ДМК-Пресс, 2018. – 652 с.

5. Флах, П. Машинное обучение : наука и искусство построения алгоритмов, которые извлекают знания из данных / П. Флах. – Москва : ДМК Пресс, 2015. – 400 с.

6. Вентцель, Е. С. Исследование операций: задачи, принципы, методология : учебное пособие / Е. С. Вентцель. – Москва : Высшая школа, 2007. – 208 с.

7. Жерон, О. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow : концепции, инструменты и техники для создания интеллектуальных систем / О. Жерон. – Санкт-Петербург : Диалектика, 2018. – 688 с.

UDC 004.9:331.44

RESEARCH OF BALANCING ON REAL DATA

Potyomin I.V., student

Belarusian State University of Informatics and Radioelectronics
Minsk, Republic of Belarus

Golda O. – PhD in Economics, associate professor

Annotation. The paper examines the influence of class imbalance on the efficiency of machine learning algorithms. Using the example of a dataset of professional burnout forecasting, three balancing methods are compared: oversampling, undersampling and weighing classes. Logistic regression, decision tree and reference vector method were experimentally evaluated. It has been established that ignoring the imbalance leads to the inability of models to identify a minor class with high overall accuracy. Class weighing is recognised as the most versatile solution, providing high balanced accuracy and PR-AUC without changing the amount of data. The results form recommendations for choosing a balancing strategy for tasks with an uneven distribution of classes.

Keywords. Machine learning, class imbalance, data balancing, oversampling, undersampling, class weighing, binary classification, logistic regression, decision tree, reference vector method, balanced accuracy, PR-AUC.