|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Теоретическая информатика и компьютерные технологии

**Отчет по домашней работе № 2**

***по курсу «Моделирование»***

**«Обработка и анализ данных при моделировании. Задача классификации»**

Студент ИУ9-81Б **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  Костриця М.И.

(Группа) (Подпись, дата) (Фамилия И.О.)

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Домрачева А. Б. (Подпись, дата) (Фамилия И.О.)

*Москва, 2023 г.*

# Цель работы

Аналитики коммерческой компании заметили, что из нее стали уходить клиенты. В наличии есть данные о поведении клиентов и расторжении договоров с компанией за прошлые периоды. Нужно спрогнозировать, уйдёт ли конкретный клиент в ближайшее время или нет. Сравните применимость трех моделей для задачи классификации: логистическая регрессия, дерево, случайный лес. Постройте модель с предельно большим значением F1-меры, метрика не должна превышать до 0.65. Предварительно проверьте F1-меру на тестовой выборке.

Признаки:

* RowNumber — индекс строки в данных
* CustomerId — уникальный идентификатор клиента
* Surname — фамилия
* Score — рейтинг клиента
* Geography — страна проживания
* Gender — пол
* Age — возраст
* Tenure — сколько лет человек является клиентом компании
* Balance — баланс, доступный для оплаты услуг компании на карте
* NumOfProducts — количество продуктов компании, используемых клиентом
* Has — наличие привилегий
* IsActiveMember — активность клиента
* EstimatedSalary — предполагаемая зарплата
* Exited — факт ухода клиента

Необходимо с помощью различных моделей решить задачу классификации, а именно предсказать, уйдет ли конкретный клиент из компании или нет.

# Теоретическая часть

*Задача классификации* — задача, в которой имеется множество объектов (ситуаций), разделённых, некоторым образом, на классы. Задано конечное множество объектов, для которых 1 известно, к каким классам они относятся. Это множество называется выборкой. Классовая принадлежность остальных объектов неизвестна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества.

*Классифицировать объект* — значит, указать номер (или наименование) класса, к которому относится данный объект.

*Классификация объекта* — номер или наименование класса, выдаваемый алгоритмом классификации в результате его применения к данному конкретному объекту.

*Логистическая регрессия* — статистическая модель, используемая для прогнозирования вероятности возникновения некоторого события путём его сравнения с логистической кривой.

*Дерево принятия решений* (также называют деревом классификации или регрессионным деревом) — средство поддержки принятия решений, использующееся в машинном обучении, анализе данных и статистике. Структура дерева представляет собой «листья» и «ветки». На рёбрах («ветках») дерева решения записаны признаки, от которых зависит целевая функция, в «листьях» записаны значения целевой функции, а в остальных узлах — признаки, по которым различаются случаи. Чтобы классифицировать новый случай, надо спуститься по дереву до листа и выдать соответствующее значение. Подобные деревья решений широко используются в интеллектуальном анализе данных. Цель состоит в том, чтобы создать модель, которая предсказывает значение целевой переменной на основе нескольких переменных на входе. Метод случайного леса — алгоритм машинного обучения, заключающийся в использовании ансамбля решающих деревьев. Алгоритм сочетает в себе две основные идеи: метод бэггинга Бреймана и метод случайных подпространств, предложенный Тин Кам Хо. Алгоритм применяется для задач классификации, регрессии и кластеризации. Основная идея заключается в использовании большого ансамбля решающих деревьев, каждое из которых само по себе даёт очень невысокое качество классификации, но за счёт их большого количества результат получается хорошим.

*F-мера* — гармоническое среднее между точностью и полнотой. Она стремится к нулю, если точность или полнота стремится к нулю.

*Точность* равна доле истинноположительных классификаций к общему числу положительных классификаций.

*Полнота* определяется как число истинноположительных классификаций относительно общего числа положительных наблюдений.

1. **Практическая реализация**

Перед началом решения задачи необходимо привести столбцы под удобный формат для обучения модели и проведения предсказания на валидационной выборке.

Для этого были выбраны значимые столбцы (Gender - пол, Geography - страна проживания), которые впоследствии были преобразованы в численный формат. Таким образом, в столбце Gender значения Male были

замечены на 1, а Female - на 0.

В столбце Geography можно заметить, что у компании клиентывсего из трех стран - Франция, Испания и Германия. Следовательно, можно заменить значение France на 0, Spain на 1, Germany на 2.

После чтения датасета и приведения всех данных в его значимых столбцах к численному формату, можно приступить к решению задачи. Были выброшены незначительные для обучения столбцы (Id, CustomerId, Surname, Exited).

После чего данные были разбиты на обучающую и валидационную выборки в соотношении 75:25, а признаки кодированы во избежание доминирования одного из них. Затем была реализована функция обучения модели и проведение предсказания на валидационной выборке, а также вычисление F1-меры модели.

Затем были применены модели логистической регрессии, дерево, случайный лес и для каждой из них был выведен итоговый список с предсказанием, уйдет ли конкретный клиент или

нет, а также выведено значение F1-меры.

def all\_models\_accuracy(features\_train, target\_train, features\_valid, target\_valid):

model\_DTC = DecisionTreeClassifier(random\_state=1515)

DTC\_score = model\_DTC.fit(features\_train, target\_train).score(features\_valid, target\_valid)

model\_RFC = RandomForestClassifier(max\_depth=7, n\_estimators=43, min\_samples\_leaf=3, random\_state=1515)

RFC\_score = model\_RFC.fit(features\_train, target\_train).score(features\_valid, target\_valid)

model\_LgR = LogisticRegression(solver = 'liblinear')

LgR\_score = model\_LgR.fit(features\_train, target\_train).score(features\_valid, target\_valid)

print("Точность:" "дерево решений", DTC\_score, "случайный лес ", RFC\_score, "логистческая регрессия", LgR\_score)

Функция оценки точности всех моделей

def rec\_prec\_f1(target\_valid, prediction):

print("Полнота" , recall\_score(target\_valid, prediction))

print("Точность", precision\_score(target\_valid, prediction))

print("F1-мера", f1\_score(target\_valid, prediction))

Функцию для изучения полноты, точности и F1-меры

def downsample(features, target, fraction):

features\_zeros = features[target == 0]

features\_ones = features[target == 1]

target\_zeros = target[target == 0]

target\_ones = target[target == 1]

features\_downsampled = pd.concat(

[features\_zeros.sample(frac=fraction, random\_state=12345)] + [features\_ones])

target\_downsampled = pd.concat(

[target\_zeros.sample(frac=fraction, random\_state=12345)] + [target\_ones])

features\_downsampled, target\_downsampled = shuffle(

features\_downsampled, target\_downsampled, random\_state=12345)

return features\_downsampled, target\_downsampled

Функцию для уменьшения представленной класса в выборке

def upsample(features, target, repeat, upsampled\_сlass):

features\_zeros = features[target == 0]

features\_ones = features[target == 1]

target\_zeros = target[target == 0]

target\_ones = target[target == 1]

if upsampled\_сlass == 0:

features\_upsampled = pd.concat([features\_zeros]\* repeat + [features\_ones] )

target\_upsampled = pd.concat([target\_zeros]\* repeat + [target\_ones] )

features\_upsampled, target\_upsampled = shuffle(

features\_upsampled, target\_upsampled, random\_state=12345)

elif upsampled\_сlass == 1:

features\_upsampled = pd.concat([features\_zeros] + [features\_ones] \* repeat)

target\_upsampled = pd.concat([target\_zeros] + [target\_ones] \* repeat)

features\_upsampled, target\_upsampled = shuffle(

features\_upsampled, target\_upsampled, random\_state=12345)

else:

features\_upsampled = 0

target\_upsampled = 0

return features\_upsampled, target\_upsampled

Функция для увеличения представленной класса в выборке

1. **Результаты**

В выборке отрицательных ответов ≈80% , положитительных ≈ 20%. С уверенностью можно сказать, что имеется дисбаланс.

Необходимо увеличить количество положительных ответов в 4 раза для достижения баланса. Либо же уменьшить кол-во отрицтаельных ответов.

1) Разделить обучающую выборку на отрицательные и положительные объекты;

2) Скопировать несколько раз положительные объекты;

3) С учётом полученных данных создадать новую обучающую выборку;

4) Перемешать данные: идущие друг за другом одинаковые вопросы не помогут обучению.

5) Обучить новые модели

При применении модели логистической регрессии значение F1-меры составляло 0.5714285714285715.

Также в итоге по прогнозу модели получаем, что 12 клиентов уйдет в ближайшее время.

При применении модели дерева значение F1-меры составляло также 0.5714285714285715. По прогнозу модели получаем, что 20 клиентов уйдет в ближайшее время.

При применении модели случайный лес значение F1-меры составляло 0.5. По прогнозу модели получаем, что 19 клиентов уйдет в ближайшее время.

Важно заметить, что при разном значении random\_state при разбивании данных мы получаем разное значение F1-меры, и путем подстановки различных значений было выведено, что

наиболее подходящий random\_state является значение 13, т.к. в этом случае во всех трех моделях значение F1-меры не превышает 0.65.

На новых данных все модели показали результат выше, чем на несбалансированной выборке. Лучшие показатели были у модели случайного леса:

Полнота 0.722488038277512

Точность 0.5392857142857143

F1-мера 0.6175869120654396

accuracy\_score финальной модели: 0.7975

1. **Вывод**

В ходе выполнения лабораторной работы был изучен процесс предсказания результата на валидационной выборке с применением различных моделей, таких как логистическая регрессия, дерево, случайный лес. Важно отметить, что необходимо привести все значимые данные к одному формату для более точного предсказания. Также большое значение играет выбранная величина random\_state при разбиении данных на обучающую и валидационную выборки.

Сравнивая результаты предсказания с использованием различных моделей между собой, можно заметить, что наибольшим значением F1-меры обладают логистическая регрессия и лес решений.