Министерство образования Российской Федерации
Научно-Исследовательский Университет Высшая Школа Экономики

Факультет Компьютерных Наук
Факультет Экономических Наук

Credit Approval

Мирзоева Алина и Шевченко Артём, БЭАД223

Научно-учебная лаборатория методов анализа больших данных

Научно-Исследовательский Семинар Т.А. Рамазян

Москва, 2023

5 ноября 2023 г.

Содержание

1 Введение

Кредитный скоринг - это важный инструмент в финансовой сфере, который позволяет банкам и финансовым учреждениям оценивать кредитоспособность заемщиков на основе различных финансовых и личных данных. В данном проекте объектом исследования является датасэт с данными о заявках по выдаче кредитных карт. Датасэт содержит зашифрованные данные из анкеты потенциального заемщика. В своем исследовании мы попытаемся определить модель машинного обучения, которая наиболее точно предсказывает целесообразность выдачи кредита заемщику.

2 Задача

Наш датасет содержит информацию о решениях банка относительно выдачи кредитных карт, где каждая заявка подвергается определению к одному из классов : "одобрен"или "не одобрен". Главная задача банка заключается в том, чтобы с достаточно высокой точностью определять надежных и ненадежных заемщиков, чтобы минимизировать потери от невозврата кредитов и одновременно избегать большого количества упущенной прибыли, связанной с отказом хорошим кандидатам.

Таким образом, наша цель заключается в разработке эффективной системы классификации, которая может точно отличать между "кредитоспособными"и "некредитоспособными"клиентами на основе предоставленных векторов признаков. Это позволит банку принимать обоснованные решения при выдаче кредитов, минимизируя риски и максимизируя прибыль.

3 Методология

Для построения различных моделей машинного обучения в своем исследовании мы использовали библиотеку scikit-learn на языке программирования Python. В нашем проекте мы исследовали модель линейной, логистической регрессии и классификации с помощью метода случайного леса. Особенности каждой из этих моделей будут описаны чуть позже, а пока разберемся с инструментами оценивания точности моделей. В нашей работе основными инструментами оценивания точности модели будут показатели, получаемые из **Матрицы ошибок**. Для начала объясним структуру матрицы ошибок:

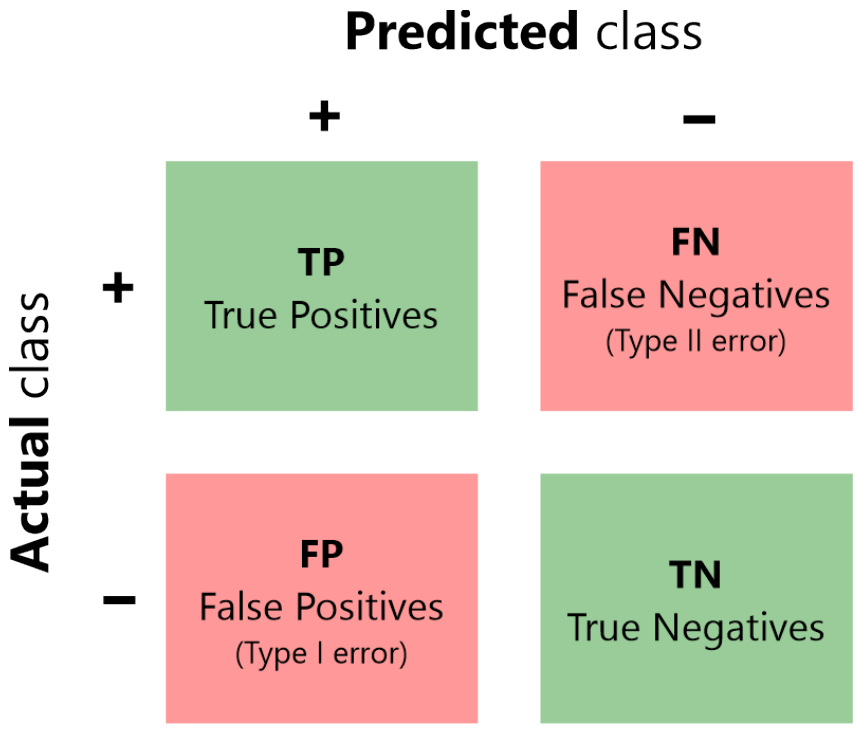


Рис. 1: Матрица ошибок

Пусть:
0 — объекты относятся к нулевому классу, (в нашей модели - отказ в выдаче займа)
1 — объекты относятся к классу 1 (в нашей задаче - одобренный займ).

Матрица ошибок - матрица, в нашем случае размера 2 на 2, такая что столбцы означают то, к какому классу объект относится в реальности, а строки - к какому признаку относится объект по прогнозу модели. В ячейках на пересечении обычно записываются числа, означающие количество объектов на валидационной выборке с одной из 4 возможных комбинаций.

С помощью матрицы ошибок мы можем ввести несколько следующих метрик:
Accuracy: $\frac{TP+TN}{TP+TN+FP+FN}$ — доля исходов, которую наша модель определила правильно.

Precision: $\frac{TP}{TP+FP}$ — какая доля от предсказанных объектов класса 1 и в реальности является объектом класса 1.

Recall: $\frac{TP}{TP+FN}$ — какая доля от тех значений, которые в реальности должны относится к классу 1, были правильно определены моделью.

Иными словами, в условиях нашей задачи:
Precision - сколько мы выдали кредитов тем людям, которым должны были выдать в реальности. Тут учитывается, что, мы могли выдать кредиты тем, кто в последствии будет недобросовестным заемщиком (то есть в реальности принадлежит классу 0). Из-за этого банк может понести убытки.

Recall - сколько мы выдали кредитов тем, кто должен был их получить, с учетом того, что мы отсеяли какую-то часть добросовестных заемщиков (представителей класса 1) и как банк упустили выгоду. Recall - способность нашей модели определять класс 1.

Также в проекте мы использовали показатель *"Среднеквадратичная ошибка"*.
Среднеквадратичная ошибка (Mean Squared Error, MSE) используется для измерения средней квадратичной разницы между фактическими значениями и предсказанными значениями в регрессионных моделях.

Среднеквадратичная ошибка вычисляется по формуле:

$$\boxed{\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Где:

- n - количество наблюдений в выборке.
- y_i - фактическое значение переменной i.
- $\hat{y}_i$ - предсказанное значение переменной i.

MSE измеряет средний квадрат разницы между фактическими и предсказанными значениями. Чем меньше значение MSE, тем лучше модель соответствует данным, поскольку это означает, что предсказанные значения ближе к фактическим значениям.

4 Исследуемые модели машинного обучения

4.1 Множественная линейная регрессия

Множественная линейная регрессия - это статистический метод, используемый для анализа отношения между независимой переменной (или переменными) и зависимыми переменной. Основная идея множественной линейной регрессии заключается в том, чтобы определить линейную зависимость между независимыми и зависимой переменными путем нахождения n-мерной плоскости, которая наилучшим образом описывает это отношение.

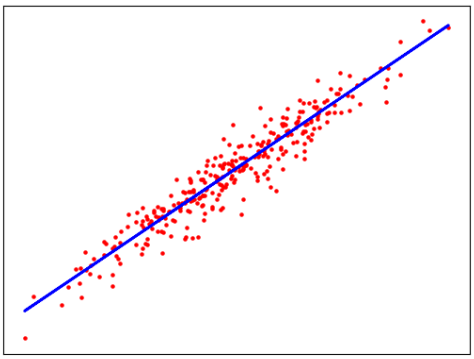


Рис. 2: Результат работы множественной линейной регрессии в двумерном случае - такая модель для удобства назыывается линейной регрессией

Стоит сразу заметить и обозначить, что множественная линейная регрессия не является наилучшим выбором для задачи кредитного скоринга, поскольку она предназначена для решения задачи регрессии (предсказание непрерывного числового значения), в то время как задача кредитного скоринга является задачей классификации (предсказание бинарного результата: одобрение кредита или отказ).

Мы же рассмотрим эту обучим эту модель для примера и небольшого анализа её работы. Кроме того, в самом коде мы познакомим читателя с базовой структурой рассмотрения модели, включая вывод матрицы ошибок.

Как мы уже договорились, линейная регрессия получает в ypred набор некоторых действительных числовых значений. Так как в валидационной выборке ytrain для каждого набора значений переменных лишь 0 или 1 (выдан кредит или нет), то и значение эти самые значения в ypred должны жить в отрезке от нуля до единицы (либо, во всяком случае, если превосходить по модулю единицу, то не сильно). Тогда, заменим значения в ypred на единицу в случае, если они ближе к ней, чем к нулю, и наоборот.

По этой логике, в работе были выведены результаты работы множественной линейной регрессии, включая описанный коэффициент из метода наименьших квадратов.

Заметим, что в теории, возможно перебрать значения threshold так, чтобы приблизиться к логике принятия решений в банке. За этим стоит следующая логика - положим, что у банка есть некоторый действительный числовой рейтинг потенциального заемщика, который хранится в ypred. Так как принятие решения о выдаче или невыдаче кредита влечёт крупные риски, которые описывались нами выше, то и банк вероятно не будет выдавать кредит заемщику с рейтингом около среднего или чуть выше него.

Однако кроме того, что это с высокой вероятностью приведёт к переобучению, мы слишком подробно рассматриваем не совсем подходящую под задачу модель - пора переходить к следующим.

4.2 Логистическая регрессия

Логистическая регрессия - это статистический метод машинного обучения, который используется для моделирования вероятности бинарного (в данном случае) события в зависимости от некоторых признаков. В данном случае она будет являться методом классификации, который применяется для прогнозирования вероятности принадлежности объекта к одному из двух классов (например, "1"или "0").

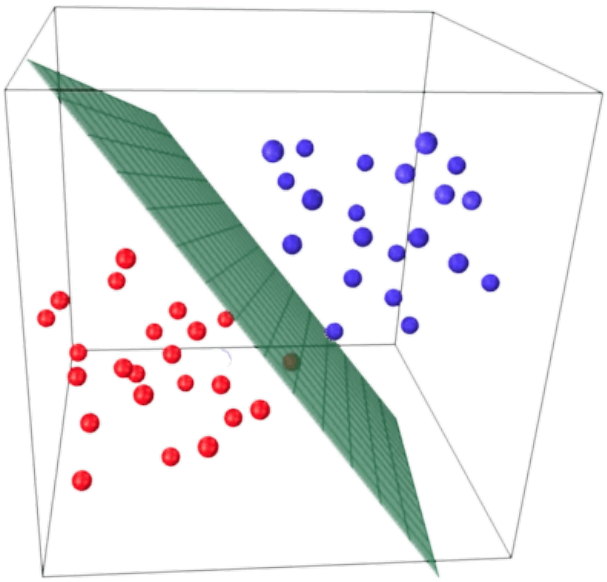


Рис. 3: Результат работы логистической регрессии в трёхмерном случае

Переформулировав модель на язык задачи кредитного скоринга, мы можем сказать, что логистическая регрессия в контексте кредитного скоринга используется для определения вероятности того, будет ли заявителю выдан кредит (класс "1") или не будет (класс "0") на основе различных признаков и их влияния на решение банка. Так, обратившись к картинке выше, стой наша задача в трёхмерном случае, мы бы могли наглядно разделить классы 0 и 1 двумерной плоскостью.

Как и в практически любой модели машинного обучения, у логистической регрессии есть ряд так называемых гиперпараметров (это параметры самой модели, задаваемые до обучения). В частности, для достижения наилучшей точности, мы можем работать с

- гиперпараметрами регуляризации (те, что, говоря простым языком, боряются с переобучением), например "penalty"(тип регуляризации) и "c"(обратное значение коэффициента регуляризации);
- с гиперпараметрами оптимизации алгоритма - maxiter (максимальное количество итераций алгоритма - по умолчанию равен 100 для логистической регресси в sklearn), solver (важный параметр, фиксирующий сам алгоритм оптимизации);
- а также некоторыми другими, которые мы рассматривать не будем, так как они в основном важны для задач многоклассовой классификации, тогда как мы имеем дело с бинарной.

Исходя из данной нам для исследования выборки и поставленной задачи, рассмотрим только гиперпараметры оптимизации, поскольку рандомизированное деление выборки на валидационную и тестовую обеспечит достаточную точность.

Тогда, для достижения наилучшей точности в поставленной задаче, мы попробовали перебрать доступные solver-ы, а также установить разное количество итераций для установления достаточного их количества для сходимости accuracy модели. Заранее предположим, что лучшую точность в нашей задаче обеспечит либо solver=”liblinear”, который подходит для сравнительно небольших наборов данных с задачами бинарной классификацией, либо solver=”lbfgs”, который подходит для чуть больших размеров датасетов, включая как бинарную, так и многоклассовую классификацию.

После проведения эксперимента оказалось, в работе выведено оптимальное (для наивысшей точности) количество итераций, а также оптимальный solver.

4.3 Модель классификации

Так как наша таргетная переменная представляет собой бинарный категориальный признак ("одобрен"или "не одобрен"кредит), то задачу предсказания таких ответов можно отнести к задаче классификации в машинном обучении.

Задача классификации – получение категориального ответа на основе набора признаков. Классификацию можно производить с помощью множества алгоритмов, но мы рассмотрим один из самых простых - метод случайного леса. Чуть более подробно, чем в ноутбуке с кодом остановимся на алгоритме:

- Создание выборок: Для построения каждого дерева случайным образом выбираются с повторениями примеры из исходных данных.
- Построение деревьев решений: Для каждой из выборок строится отдельное решающее дерево. Построение дерева включает в себя разбиение данных на узлы (по одному признаку в каждом узле), выбор лучшего разбиения и продолжение этого процесса до достижения необходимой глубины дерева.
- Прогнозирование: Когда все деревья построены, они могут быть использованы для прогнозирования новых данных. Для задачи классификации, каждое дерево голосует за класс объекта, и класс с наибольшим числом голосов становится предсказанием случайного леса.

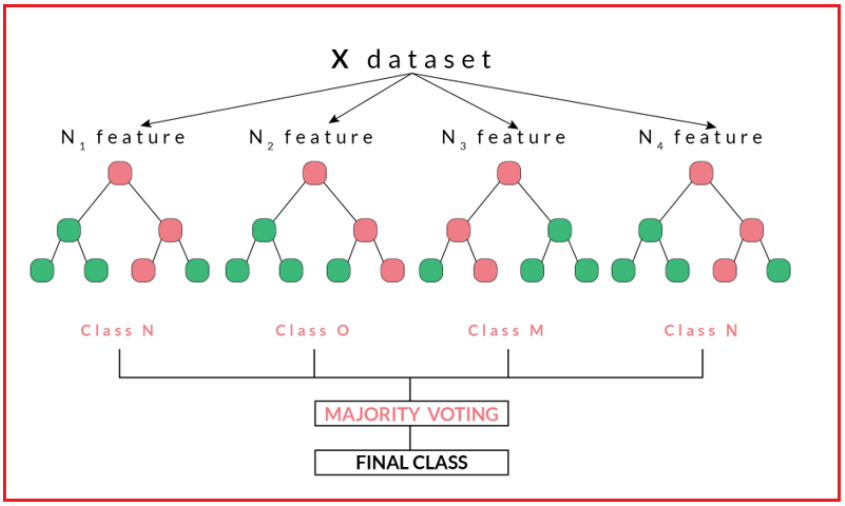


Рис. 4: Алгоритм случайного леса

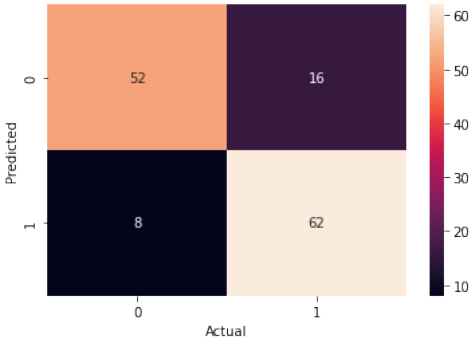
У функции RandomForestClassifier существует несколько параметров, но мы в исследовании остановились подробнее на двух основных:

- n-estimators (по умолчанию: 100): Этот параметр указывает количество деревьев в случайном лесе. Большее число деревьев может улучшить качество модели, но также может увеличить время обучения. В этом мы убедимся чуть позже.
- max-depth (по умолчанию: None): Этот параметр ограничивает максимальную глубину каждого дерева в случайном лесе. Если установлено значение None, деревья не будут ограничены по глубине.

  В проекте мы подобрали оптимальное количество деревьев в случайном лесу, проиллюстрировав изменение точности и precision на графике, чтобы достичь максимального precision, так как этот показатель посчитали наиболее важным для модели, чтобы она удовлетворяла потребностям банка (Подробно объяснили в разделе "Методология".

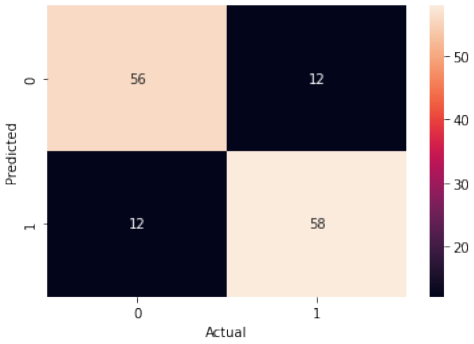
5 Результаты

Для модели **линейной регрессии**, матрица ошибок выглядела следующим образом:



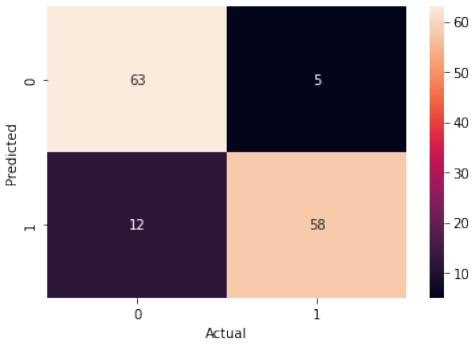
- Коэффициент метода наименьших квадратов составил 0.135
- Precision составил 0.795
- Recall составил 0.886

Для модели **логистической регрессии**, матрица ошибок выглядела следующим образом:



- Precision составил 0.829
- Recall составил 0.829

Для модели **случайного леса**, матрица ошибок выглядела следующим образом:



- Коэффициент метода наименьших квадратов составил 0.123
- Precision составил 0.921
- Recall составил 0.829

6 Выводы

- **Линейная регрессия:**

  Модель линейной регрессии была использована для оценки влияния набора независимых факторов на кредитный рейтинг заемщика. Эта модель предоставила базовый анализ, выявив взаимосвязи между различными переменными и кредитным рейтингом. Однако она, недостаточно учла нелинейные зависимости между факторами и кредитной способностью.

- **Логистическая регрессия:**

  Логистическая регрессия была применена для задачи бинарной классификации, где целью было определение вероятности одобрения или отклонения заявки на кредит. Эта модель успешно разделила заемщиков на две группы в зависимости от их вероятности выдачи кредита. Логистическая регрессия также позволила оценить важность различных факторов влияющих на решение о кредите.

- **Метод случайных лесов:**

  Метод случайных лесов, благодаря своей способности к обработке нелинейных взаимосвязей и обработке большого количества переменных, справился отлично в задаче кредитного скоринга. Он предоставил более точные прогнозы кредитоспособности заемщиков, учитывая множество факторов, включая кредитную историю, доход и другие параметры.

Итак, работа показала, что комбинирование различных методов машинного обучения может повысить точность и надежность модели кредитного скоринга. Каждая модель имеет свои преимущества и может использоваться в разных аспектах задачи кредитного скоринга, в зависимости от целей и требований бизнеса.