МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ РАДИОФИЗИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ
Кафедра интеллектуальных систем

РАКОВЕЦ
Андрей Владимирович

АНАЛИЗ ЗНАЧИМОСТИ ВХОДНЫХ ПРИЗНАКОВ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ С УЧИТЕЛЕМ

Дипломная работа

Научный руководитель:
старший преподаватель
Щетько Н.Н.

Допущен к защите
«____» _____2022.
Зав. кафедрой интеллектуальных систем
кандидат физ.-мат. наук, доцент Козлова В.И.

Минск, 2022

ОГЛАВЛЕНИЕ

РЕФЕРАТ 3
ВВЕДЕНИЕ 6
ГЛАВА 1 МОДЕЛИ С РЕШЕННОЙ ЗАДАЧЕЙ ОЦЕНКИ ЗНАЧИМОСТИ ПРИЗНАКОВ 7
1.1 Бинарные деревья 7
1.2 Случайные леса 15
1.3 Перестановочная важность 18
1.4 Суррогатные модели 20
ГЛАВА 2 МЕТОДЫ ОЦЕНКИ ЗНАЧИМОСТИ ПРИЗНАКОВ В ЗАДАЧЕ ОТБОРА ПРИЗНАКОВ 22
2.1 Задача отбора признаков 22
2.2 Методы – фильтры(filters) 24
2.3 Методы-обертки (Wrapper methods) 31
2.4 Встроенные методы (embedded) 35
ГЛАВА 3 ПРИМЕНЕНИЕ МЕТОДОВ ОЦЕНКИ ВАЖНОСТИ ПРИЗНАКОВ В РЕАЛЬНОЙ ЗАДАЧЕ 38
3.1 Описание задачи 38
3.2 Обучение моделей 38
3.3 Внесение дополнительных шумовых признаков 44
3.4 Подбор гиперпараметров 49
ЗАКЛЮЧЕНИЕ 55
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 56

РЕФЕРАТ

Дипломная работа объемом 55 страниц содержит 2 таблицы, 34 формулы, список использованной литературы из 38 наименований.

АНАЛИЗ ЗНАЧИМОСТИ ВХОДНЫХ ПРИЗНАКОВ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ С УЧИТЕЛЕМ

Целью исследования является исследование возможности интерпретации поведения обученных моделей машинного обучения с учителем на основе методов оценки важности входных признаков.

В качестве задач исследования, вытекающих из цели, ставилось рассмотреть проблему "черного ящика" при построении моделей машинного обучения с учителем, рассмотреть подход к анализу значимости признаков в моделях на основе деревьев принятия решений, рассмотреть задачу определения значимости признаков в контексте понижения размерности, исследовать алгоритмы на устойчивость к шумовым признакам с различными видами распределений.

Методы исследования: аналитический, статистический.

Новизна заключается в модификации одного из методов оценки важности признаков. Проведенные исследования показали, что данный модифицированный метод склонен увеличивать оценки важности признаков, которые были признаны маловажными оригинальным методом.

Ключевые слова: оценка важности входных признаков, отбор признаков, дерево решений, случайный лес, методы-фильтры, обёрточные методы, встроенные методы.

РЭФЕРАТ

Дыпломная праца аб'ёмам 55 старонак змяшчае 2 табліцы, 34 формулы, Спіс выкарыстанай літаратуры з 38 найменняў.

АНАЛІЗ ЗНАЧНАСЦІ ЎВАХОДНЫХ ПРЫКМЕТ МАДЭЛЯЎ МАШЫННАГА НАВУЧАННЯ З НАСТАЎНІКАМ

Мэтай даследавання з'яўляецца вывучэнне магчымасці інтэрпрэтацыі паводзін навучаных мадэляў машыннага навучання з настаўнікам на аснове метадаў ацэнкі важнасці ўваходных прыкмет.

У якасці задач даследавання, якія вынікаюць з мэты, ставілася разгледзець праблему "чорнай скрыні" пры пабудове мадэляў машыннага навучання з настаўнікам, разгледзець падыход да аналізу значнасці прыкмет у мадэлях на аснове дрэў прыняцця рашэнняў, разгледзець задачу вызначэння значнасці прыкмет у кантэксце паніжэння памернасці, даследаваць алгарытмы на ўстойлівасць да шумавых прыкметах з рознымі відамі размеркаванняў.

Метады даследавання: аналітычны, статыстычны.

Навізна заключаецца ў мадыфікацыі аднаго з метадаў ацэнкі важнасці прыкмет. Праведзеныя даследаванні паказалі, што дадзены мадыфікаваны метад схільны павялічваць ацэнкі важнасці прыкмет, якія былі прызнаныя малаважнымі арыгінальным метадам.

Ключавыя словы: ацэнка важнасці ўваходных прыкмет, адбор прыкмет, дрэва рашэнняў, выпадковы лес, метады-фільтры, абгортачныя метады, убудаваныя метады.

ABSTRACT

Thesis: 55-pages, 2 tables, 34 formulas, a list of references from 38 titles.

ANALYSIS OF THE SIGNIFICANCE OF INPUT FEATURES OF MACHINE LEARNING MODELS WITH A TEACHER

The aim of the research is to investigate the possibility of interpreting the behavior of trained machine learning models with a teacher based on methods for assessing the importance of input features.

As research tasks arising from the goal, it was set to consider the problem of a "black box" when building machine learning models with a teacher, to consider an approach to analyzing the significance of features in models based on decision trees, to consider the task of determining the significance of features in the context of dimension reduction, to investigate algorithms for resistance to noise features with different types of distributions.

Research methods: analytical, statistical.

The novelty lies in the modification of one of the methods for assessing the importance of features. The conducted studies have shown that this modified method tends to increase the estimates of the importance of features that were considered unimportant by the original method.

Keywords: evaluation of the importance of input features, feature selection, decision tree, random forest, filter methods, wrapper methods, embedded methods.

ВВЕДЕНИЕ

Одна из фундаментальных проблем машинного обучения в том, что обученные модели работают как «черный ящик». Машинное обучение в целом сводится к тому, что какая-то очень гибкая и настраиваемая функцию подгоняем под такой вид, который хорошо описывает какие-то имеющиеся данные. Например, зная, как формируется нейронная сеть, зная, как происходит её обучение, но после того, как она обучена, её поведение очень сложно интерпретировать — она может имеющиеся у нас данные описывать просто идеально, но информация о том, какие именно принципы лежат в полученном функциональном преобразовании, будет по-прежнему неизвестна. Т.е. зная выходной результат, исследователь не знает, почему получил именно такой результат, а не какой-то другой.

Цель: исследовать возможность интерпретации поведения обученных моделей машинного обучения с учителем на основе методов оценки важности входных признаков для модели.

Задачи:

- рассмотреть проблему «черного ящика» при построении моделей машинного обучения с учителем,
- рассмотреть подход к анализу значимости признаков в моделях на основе деревьев принятия решений,
- рассмотреть задачу определения значимости признаков в контексте понижения размерности,
- исследовать алгоритмы на устойчивость к шумовым признакам с различными видами распределений

ГЛАВА 1
МОДЕЛИ С РЕШЕННОЙ ЗАДАЧЕЙ ОЦЕНКИ ЗНАЧИМОСТИ ПРИЗНАКОВ

1.1 Бинарные деревья

Рассмотрим бинарное дерево, в котором:

- каждой вершине v ветви приписана функция (предикат) $\beta_v : X \to \{0, 1\}$;
- каждой листовой вершине v приписан прогноз $c_v \in Y$ (в случае классификации листу приписывается вектор долей распределения классов).

Рассмотрим алгоритм p(x), который начинает свое выполнение из корневой вершины v_0 и вычисляет значение функции β_{v0}. Если оно равно нулю, то алгоритм переходит в левую дочернюю вершину, если единице - в правую, вычисляет значение предиката в новой вершине и повторяет данную процедуру. Процесс продолжается, пока не будет достигнута листовая вершина. Данный алгоритм возвращает класс, приписанный данной вершине. Такой алгоритм называется решающим бинарным деревом и описан в работе [38].

На практике в большинстве случаев используются одномерные предикаты β_v, которые сравнивают значение одного из признаков с числом:

$$\beta_v(x; j, t) = [x_j < t].$$

Существуют и многомерные предикаты, например:

- линейные $\beta_v(x) = [\langle w, x\rangle < t]$;
- метрические $\beta_v(x) = [\rho(x, x_v) < t]$, где точка x_v является одним из объектов выборки - некоторой точкой признакового пространства.

Многомерные предикаты позволяют получить более сложные разделяющие гиперповерхности, но редко используются на практике — например, из-за того, что увеличивают и без того огромные риски переобучения данных деревьев. Далее будем рассматривать только одномерные предикаты.

Очевидно, что для любой выборки существует такое решающее дерево, которое не допускает на ней ни одной ошибки — дерево, в каждой листовой вершине которого находится ровно по одному объекту из выборки. Высока вероятность того, что такое дерево окажется переобученным и не сможет показать хорошую точность предсказаний на новых данных. Рассчитывать на наличие у дерева некоторой обобщающей способности можно у такого дерева,

которое является минимальным с точки зрения количества листовых вершин среди всех решающих деревьев, которые допускают минимум ошибок на обучении. Однако, задача поиска такого дерева является NP-полной, и поэтому приходится ограничиваться жадными алгоритмами построения дерева [7].

Опишем жадный алгоритм построения бинарного решающего дерева. Начиная со всей обучающей выборки X находиться наилучшее ее разбиение с помощью предиката $\beta_v(x; j, t) = [x_j < t]$ на две части $R_1(j, t) = \{x \mid \beta_v\}$ и $R_2(j, t) = \{x \mid \overline{\beta_v}\}$ с точки зрения заданного функционала качества $Q(X, j, t)$ - сравнительного качества различных способов разбиения заданной совокупности элементов на классы, т. е. того количественного критерия, следуя которому можно было бы предпочесть одно разбиение другому. Найдя наилучшие значения j и t, на выделенной основе создаётся корневая вершина дерева, ставиться ей в соответствие предикат β_v. Объекты разбиваются на две (для бинарного дерева) группы — одни для левого поддерева, другие для правого. Для каждой из указанных выше групп процедура повторяется, в результате строятся вершины, дочерние для корневой, и так далее.

В каждой вершине проверяется выполнение некоторого условия останова — условия, при выполнении которого рекурсия останавливается, и данная вершина назначается листом. Когда дерево построено, каждому листу ставится в соответствие некоторая выходная метрика. В случае задачи классификации это либо класс с наибольшим числом объектов в листе, или вектор вероятностей попаданий в соответствующий класс (например, вероятность попадания в класс может быть равна доле объектов данного класса в листе). Для задач регрессии возможными выходными метриками являются: среднее значение, квантиль или некоторая другая функция, зависящая от целевых переменных в листе. Выбор метрики также зависит от функционала качества в исходной задаче [38].

Решающие деревья обладают способностью обрабатывать такие ситуации, в которых у некоторых объектов пропущены значения одного или нескольких признаков. Для этого необходимо модифицировать процедуру разбиения выборки в вершине, что можно сделать несколькими способами [7], один из которых будет описан ниже.

Таким образом, конкретный метод построения решающего дерева определяется [7]:

1. Функцией-предикатом в вершинах;
2. Функционалом качества $Q(X, j, t)$;
3. Критерием останова;
4. Выходной метрикой;
5. Методом обработки пропущенных значений.

При построении дерева для выбора необходимого предиката необходимо задать функционал качества, на основе которого будет осуществляется такой выбор, и, соответственно, разбиение выборки на каждом шаге. Обозначим через R_m множество объектов в вершине, для которой на данном шаге строится разбиение, а через R_ℓ и R_r — множества объектов, попадающих в левое и правое поддерево соответственно при данном разбиении. Обычно используются функционалы следующего вида (формула 1.1) [38]:

$$Q(R, j, s) = H(R_m) - \frac{|R_l|}{|R_m|} H(R_l) - \frac{|R_r|}{|R_m|} H(R_r) \tag{1.1}$$

Здесь $H(R)$ — это критерий информативности (англ. impurity criterion), который оценивает распределение целевой переменной среди объектов множества R. Чем меньше разброс целевой переменной, тем меньше должно быть значение критерия информативности. Функционал качества $Q(R, j, s)$ при этом будем максимизировать. В таком случае, так как R_m заранее заданное множество, ищется минимум значений $\frac{|R_l|}{|R_m|} H(R_l)$ и $\frac{|R_r|}{|R_m|} H(R_r)$. Как уже обсуждалось выше, каждому листу дерева приписана метрика — вещественное число, вероятность или класс. Исходя из этого, информативность множества объектов R может оценивается тем, насколько хорошо их целевые переменные предсказываются константой (при оптимальном значении данной константы) (формула 1.2):

$$H(R) = \min_{c \in Y} \frac{1}{|R|} \sum_{(x_i, y_i) \in R} L(y_i, c) \tag{1.2}$$

где $L(y_i, c)$ — некоторая функция потерь. Далее рассмотрим, какие именно критерии информативности часто используются в задачах регрессии и классификации.

Одним из распространённых критериев в задачах **регрессии** является сумма квадратов отклонения целевых переменной от постоянной в качестве функции потерь $L(y_i, c)$. Подставив данную функцию в формулу (1.2), получим:

$$H(R) = \min_{c \in Y} \frac{1}{|R|} \sum_{(x_i, y_i) \in R} (y_i - c)^2 \tag{1.3}$$

Как несложно показать, минимум в этом выражении будет достигаться на среднем значении целевой переменной. Значит, формула (1.3) преобразуется в:

$$H(R) = \min_{c \in Y} \frac{1}{|R|} \sum_{(x_i, y_i) \in R} \left(y_i - \frac{1}{|R|} \sum_{(x_i, y_i) \in R} y_i \right)^2 \quad (1.4)$$

При таком выборе информативность вершины измеряется дисперсией данной вершины — чем меньше разброс целевой переменной, тем меньше значение критерия информативности и тем лучше вершина. Можно использовать и другие функции потерь $L(y_i, c)$ — например, при выборе абсолютного отклонения в качестве функции потерь значение постоянной оказывается равной медиане.

Для задач **классификации** будем использовать следующие обозначения:

Обозначим через p_k долю объектов класса k ($k \in \{1, \ldots, K\}$), попавших в вершину R:

$$p_k = \frac{1}{|R|} \sum_{(x_i, y_i) \in R} [y_i = k]. \quad (1.5)$$

Через k_* обозначим класс с наибольшей долей среди объектов данной вершины: $k_* = arg \max_k p_k$

Рассмотрим индикатор ошибки как функцию потерь:

$$H(R) = \min_{c \in Y} \frac{1}{|R|} \sum_{(x_i, y_i) \in R} [y_i \neq c]. \quad (1.6)$$

Легко увидеть, что прогнозом является наиболее популярный класс k_* — значит, критерий ошибки классификации выражается следующей формулой:

$$H(R) = \min_{c \in Y} \frac{1}{|R|} \sum_{(x_i, y_i) \in R} [y_i \neq k_*] = 1 - p_k. \quad (1.7)$$

Данный критерий ошибки классификации является грубым, поскольку учитывает частоту появления лишь одного класса.

Рассмотрим метрику, при которой критерий информативности оценивается на основе распределения на всех классах $c = (c_1, \ldots, c_k), \sum_{k=1}^{K} c_k = 1$. Качество

такого распределения можно измерить, например, с помощью **критерия Бриера**:

$$H(R) = \min_{\sum_k c_k=1} \frac{1}{|R|} \sum_{(x_i,y_i)\in R} \sum_{k=1}^{K} (c_k - [y_i = k])^2 \quad (1.8)$$

Можно показать, что оптимальный вектор распределения классов для такого критерия состоит из долей соответствующих классов p_k [7]:

$$c_* = (p_1, \dots, p_k)$$

Прямой подстановкой такого вектора в исходный критерий информативности после ряда преобразований получим **критерий Джини** [7]:

$$H(R) = \sum_{k=1}^{K} p_k(1 - p_k) \quad (1.9)$$

Энтропийный критерий, в отличие от рассмотренных выше критериев, оценивает качество распределения логарифмическими потерями, или логарифмом правдоподобия:

$$H(R) = \min_{\sum_k c_k} \left(-\frac{1}{|R|} \sum_{(x_i,y_i)\in R} \sum_{k=1}^{K} [y_i = k] log c_k \right) \quad (1.10)$$

Для вывода оптимальных значений c_k вспомним, что каждая вершина должна назначить объекту, попавшему в вершину, класс, поэтому сумма всех значений c_k равна единице. Как известно из методов оптимизации, для поиска условного экстремума необходимо искать минимум лагранжиана:

$$L(c, \lambda) = \min_{c_k} \left(-\frac{1}{|R|} \sum_{(x_i,y_i)\in R} \sum_{k=1}^{K} [y_i = k] log c_k + \lambda \sum_{k=1}^{K} c_k \right) \quad (1.11)$$

Дифференцируя, получаем:

$$\frac{\partial}{\partial c_k} L(c, \lambda) = -\frac{1}{|R|} \sum_{(x_i,y_i)\in R} [y_i = k] \frac{1}{c_k} + \lambda = -\frac{p_k}{c_k} + \lambda = 0 \quad (1.12)$$

откуда выражаем $c_k = p_k/\lambda$.

Учитывая указанное выше, найдем параметр оптимизации λ:

$$1 = \sum_{k=1}^{K} c_k = \frac{1}{\lambda} \sum_{k=1}^{K} p_k = \frac{1}{\lambda}. \quad (1.13)$$

откуда λ = 1. Подставляя данное значение в формулу вычисления c_k, получим $c_k = p_k$, как и в предыдущем случае. Подставляя данные выражения в критерий, получим, что он будет представлять собой энтропию распределения классов [7]:

$$H(R) = -\sum_{k=1}^{K} p_k \log(p_k). \quad (1.14)$$

Из теории вероятностей известно, что энтропия ограничена снизу нулем, причем минимум достигается на вырожденных распределениях ($p_i = 1$, $p_j = 0$ для $i \neq j$). Максимальное же значение энтропия примет для равномерного распределения ($p = p_i = K^{-1}$) и равна логарифму от K. Отсюда понятно, что энтропийный критерий будет распределять классы в вершине со смещением в сторону более «вырожденных» распределений [7].

Существует большое количество **критериев останова** – правил, по которым алгоритм построения останавливается. Перечислим некоторые из них:

• Останов в случае достижения максимальной глубины дерева.

• Останов в случае достижения минимального числа объектов в листе.

• Останов в случае достижения максимального количества листьев в дереве.

• Останов в случае, если все объекты в листе относятся к одному классу.

• Требование улучшения функционала качества при разбиении как минимум на s процентов.

Оптимальный выбор подобных критериев и их параметров может существенно повлиять на качество предсказаний для решающего дерева. Тем не менее, такой подбор является трудозатратным и требует проведения большого количества оценок качества обучения при переборе.

Стрижка дерева является альтернативным способом построения дерева. При использовании данного метода построения сначала строится переобученное дерево (каждый лист предсказывает один объект, например), а затем производится некоторая оптимизация структуры дерева ради улучшения способности к обобщению. Существует ряд исследований, показывающих, что стрижка позволяет достичь лучшего качества по сравнению с ранним остановом построения дерева на основе различных критериев [7]. Тем не менее, на данный

момент методы стрижки редко используются и не реализованы в большинстве библиотек для анализа данных.

Причина такого поведения заключается в том, что решающие деревья являются слабыми алгоритмами и не представляют большого интереса для разработчиков, а при использовании в композициях существуют более простые методы получения качественных результатов, из-за чего данный метод построения одиночного дерева не нашел широкого применения.

Впрочем, рассмотрим один из методов стрижки, называемый cost-complexity pruning. Значение функционала *Q(T)* будет минимально на таком дереве T_0 (среди всех поддеревьев), для которого выполняется такое условие, что в каждом из листьев находятся объекты только одного класса. Однако данный функционал характеризует лишь качество предсказаний для решающего дерева на обучающей выборке, и чрезмерная подгонка под нее приводит к переобучению.

Чтобы преодолеть эту проблему, воспользуемся методом штрафных функций, и введем новую функцию $Q_\alpha(T)$, представляющий собой сумму исходной функции *Q(T)* и штрафной функции, отражающей стоимость штрафа за размер дерева:

$$Q_\alpha(T) = Q(T) + \alpha|T|, \tag{1.15}$$

где $|T|$ — мощность множества листовых вершин, определяющегося числом листьев в поддереве T, а $\alpha \geq 0$ — параметр регуляризации. Это один из примеров регуляризации – встроенного в структуру модели метода отбора признаков, которые вносят в структуру штраф за сложность построенной модели с целью увеличения точности классификации. Можно показать, что существует последовательность вложенных деревьев с одинаковыми корнями:

$$T_k \subset T_{k-1} \subset \cdots \subset T_0,$$

(здесь T_k — тривиальное дерево, состоящее из корня дерева T_0), в которой каждое дерево T_i минимизирует критерий (1.15) для α из интервала $\alpha \in [a_i, \alpha_{i+1})$, причем

$$0 < \alpha_0 < \alpha_1 < \cdots < a_k < \infty.$$

Эту последовательность можно достаточно эффективно найти путем обхода дерева. Далее из нее выбирается оптимальное дерево по точности предсказаний модели [7].

Одним из основных достоинств решающих деревьев является возможность обработки ситуаций, возникающих при наличии **пропущенных значений**. Пусть необходимо вычислить функционал качества для предиката $\beta(x) = [x_j < t]$, однако в выборке объектов R для данной вершины для некоторых объектов пропущены значения признака j — обозначим такие объекты через V_j . В таком случае при вычислении функционала можно удалить из выборки такие объекты, сделав поправку на потерю информации от этого по формуле 1.16:

$$Q(R, j, s) \approx \frac{|R \setminus V_j|}{|R|} Q(R \setminus V_j, j, s) \tag{1.16}$$

Затем, если предикат $\beta(x) = [x_j < t]$ останется лучшим для разбиения, поместим объекты из V_j как в левое, так и в правое поддерево. Если объект попал в вершину, предикат которой не может быть вычислен из-за пропущенного значения, то прогнозы для такой вершины вычисляются в обоих поддеревьях, и усредняются с весами, пропорциональными числу объектов в указанных поддеревьях. Иными словами, если прогноз вероятности для класса k в выборке R_m обозначается через $a_{mk}(x)$, $a_{lk}(x)$ и $a_{rk}(x)$ – прогнозы поддеревьев в выборках R_l и R_r соответственно, то получаем такую формулу:

$$a_{mk}(x) = \begin{cases} a_{lk}(x), \beta_m = 0; \\ a_{rk}(x), \beta_m = 1; \\ \frac{|R_l|}{|R_m|} a_{lk}(x) + \frac{|R_r|}{|R_m|} a_{rk}(x), \ \beta_m \text{ нельзя вычислить.} \end{cases} \tag{1.17}$$

Иной подход заключается в концепции искусственных предикатов в каждой вершине. Так называется предикат, который применяет иной признак, однако при этом составляет разбиение, предельно близкое к данному. Отметим, что зачастую похожее свойство демонстрируют и гораздо более легковесные методы обработки пробелов — например, возможно променять все пробелы на ноль. Для деревьев вдобавок можно променять пробелы в симптоме на числа, что превышают все значения исходного признака. Тогда в дереве можно будет избрать такое разделение по этому признаку, что все объекты с известными значениями пойдут в левое поддерево, а все объекты с пропусками — в правое [7].

Самый очевидный жадный способ **обработки категориальных признаков** заключается в разбиении вершины на столько поддеревьев, каково число кардинальности у множества значений признака (multi-way splits) [7].

Рассмотрим подробнее другой подход. Пусть категориальный признак x_j имеет множество значений $Q = \{u_1, \ldots, u_q\}$, $|Q| = q$. Разобьем множество значений на два непересекающихся подмножества: $Q = Q_1 \cup Q_2$, и определим

предикат как индикатор попадания в первое подмножество: $\beta(x) = [x_j \in Q_1]$. Таким образом, объект переносится в левое поддерево, если признак x_j попадает в множество Q_1, и в правое поддерево в противном случае. Основная проблема такого подхода заключается в том, что для построения оптимального разбиения необходимо перебрать $2^{q-1} - 1$ вариантов, что вычислительно затратно для больших значений q.

Как можно увидеть, полного перебора в случаях с задачами бинарной классификацией и регрессии можно избежать. Обозначим через $R_m(u_j)$ множество объектов вершины m, у которых признак x_j имеет значение u_j, а через $N_m(u_j)$ обозначим число таких объектов. В случае бинарной классификации упорядочим все значения категориального признака на основе того, какая доля объектов с таким значением имеет класс +1:

$$\frac{1}{N_m(u_1)} \sum_{x_i \in R_m(u_1)} [y_i = +1] \leq \cdots \leq \frac{1}{N_m(u_q)} \sum_{x_i \in R_m(u_q)} [y_i = +1], \qquad (1.18)$$

после чего заменим категорию u_i на число i, и будем искать разбиение как для вещественного признака. Можно показать, что поиск оптимального разбиения по критерию Джини или энтропийному критерию позволяет получить такое же разбиение, как и при полном переборе [7,12].

1.2 Случайные леса

Композиция — это парадигма, в которой целое составляется из частей. В машинном обучении такая парадигма породила класс методов, называемых ансамблевыми, в которых несколько моделей (называемых "плохими учениками") обучаются для решения одной и той же проблемы и, некоторым образом, агрегируются для достижения лучших результатов. Основная гипотеза заключается в том, что при правильной композиции можно получить более точные и/или надежные модели [4]. Смещение и разброс модели являются двумя основными характеристиками модели. Чтобы модель могла точно предсказать результаты, необходимо, чтобы модель имела оптимальное количество степеней свободы, чтобы построить оптимальную гиперповерхность разделения. Это известный компромисс между смещением и разбросом (рис. 1.1). [4].

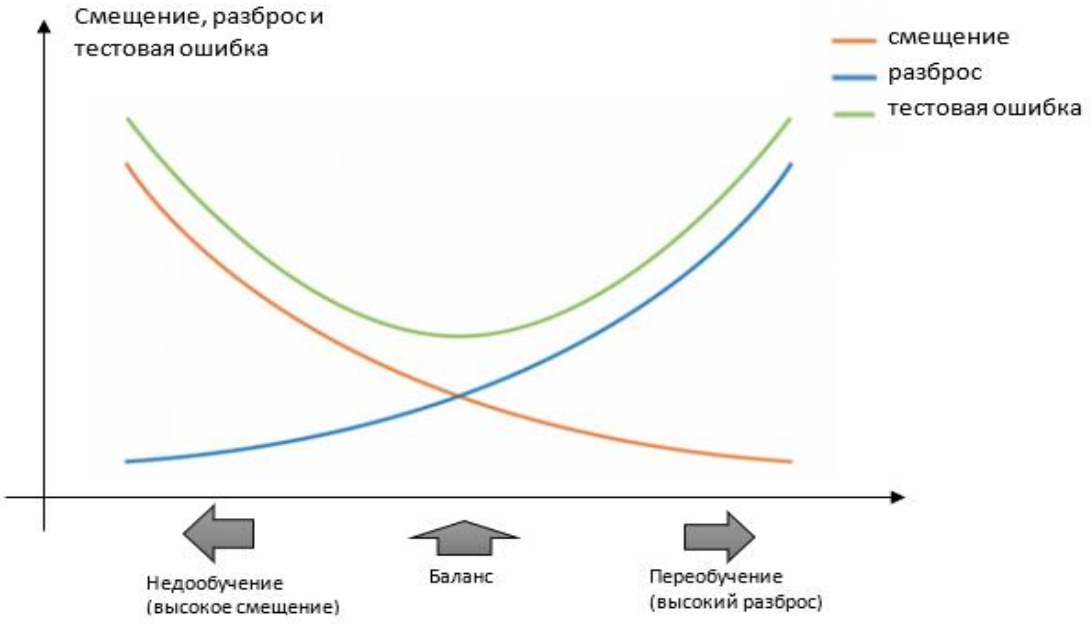

Рисунок 1.1. - Иллюстрация компромисса между смещением и разбросом

В большинстве случаев базовые модели работают сами по себе менее точно из-за того, что они имеют высокое смещение (например, модели с низкой степенью свободы), либо имеют слишком высокий разброс, чтобы быть устойчивыми (например, модели с высокой степенью свободы). Идея ансамблевых методов состоит в том, чтобы попытаться уменьшить смещение и/или разброс таких слабых учеников путём их агрегирования, чтобы создать предсказание ***сильного ученика*** (или ***модель ансамбля***), который достигает лучших результатов [4].

Существуют несколько общих методов объединения результатов работы базовых алгоритмов в композиции: голосование, взвешенное голосование, смесь экспертов (Mixture of Experts [32]). Данные методы часто применяются, когда базовые алгоритмы существенно отличаются друг от друга. В случаях, когда необходимо построить композицию, состоящую из единственного базового алгоритма, можно использовать **бэггинг** (bagging, boostrap aggregation) [22].

Идея такого метода состоит в том, что такой алгоритм многократно обучается на случайных подмножествах строк из обучающей выборки с повторениями. Такой метод построения модели позволяет получать модели с разными предсказаниями на общей выборке, которые впоследствии можно объединить общими методами объединения, а метод генерации подмножества строк принято называть **бутсрап** (bootsrap).

Похожим на бэггинг методом является **метод случайных подпространств** (random subspace method, RSM [27]). Его идея заключается в выборе случайных

подмножеств признаков (случайного подмножества столбцов таблицы, если данные представимы в таком виде). Широко известным примером использования бэггинга и RSM является случайный лес [23]. Одним из преимуществ таких методов является возможность параллельно обучать разные алгоритмы для их последующей аггрегации, чего не скажешь про **бустинг.** Идея бустинга состоит в жадном выборе очередного алгоритма для добавления в композицию так, чтобы он лучшим образом компенсировал имеющиеся на этом шаге ошибки. Широко известные примеры бустинга — AdaBoost [24] и градиентный бустинг (Gradient boosting [25]).

Стекинг (Stacked generalization) был впервые предложен Д. Волпертом в 1992 году в работе [20]. Основная идея стекинга заключается в использовании очередного классификатора для агрегации результатов, полученных из базовых алгоритмов. Автор обобщает идею стекинга тем, что предлагает строить иерархическую структуру, обучая классификаторы первого уровня над метапризнаками первого уровня, получать метапризнаки второго уровня и так далее.

Деревья решений являются популярными базовыми моделями для ансамблевых методов. Агрегированные модели, состоящие из нескольких деревьев решений, можно назвать «лесами». Как уже упоминалось, нахождение оптимальной высоты дерева для наилучших предсказаний - задача нетривиальная, поэтому деревья для агрегации могут быть выбраны либо невысокими (высотой в небольшое число вершин), либо высокими (глубиной в множество вершин, если переобученные). Невысокие деревья имеют малый разброс, но высокое смещение (рисунок 1.1), и тогда для них лучшим выбором методов аггрегации станут **последовательные методы – бустинг, стекинг**. Высокие деревья, с другой стороны, имеют низкое смещение, но высокий разброс и, таким образом, являются подходящим выбором для **бэггинга** [4].

Случайный лес чаще всего использует метод бэггинга. Тем не менее, случайные леса также могут использовать и другие методы, упомянутые выше, как и объединять некоторые из них, например, рассмотрим сочетание бэггинга с методом случайных подпространств.

Случайность подпространств приводит к тому, что различные деревья смотрят на различную информацию для предсказаний и, таким образом, уменьшают взаимосвязь между различными выходами модели. Другое преимущество метода случайных подпространств заключается в том, что модель становится более устойчивой к отсутствующим или пропущенным данным: информацию о значениях, отсутствующих для данной модели можно восстанавливать на основе деревьев, которые учитывают признаки, где данные не отсутствуют. Именно таким образом алгоритм случайного леса может сочетать в себе концепции бэггинга и метода случайных подпространств для создания более устойчивых моделей [4].

1.3 Перестановочная важность

Перестановочная важность признаков — это метод проверки модели, который может быть использован для любой обученной модели, когда данные представимы в виде таблиц. Это особенно полезно для нелинейных моделей или моделей с сложным алгоритмом интерпретации. Перестановочная важность признаков определяется как уменьшение оценки точности модели машинного обучения при случайном перемешивании одного атрибута объекта. Такая процедура разрывает связь между признаком и целевым объектом, таким образом, снижение оценки точности модели указывает на численное значение зависимости от признака. Данный метод не зависит от модели и может быть вычислен много раз с различными перестановками признака [33].

Признаки, которые считаются малозначимыми для плохой модели (имеющие низкую точность или иную метрику), могут быть очень важны для сильной модели. Поэтому сперва важно оценить прогностическую способность модели, используя проверенный набор, прежде чем вычислять важность. Перестановочная важность отражает не ценность признака как такового, а то, насколько важен признак для конкретной модели.

Такая оценка может быть дана с помощью функции permutation_importance библиотеки sklearn[33] - она вычисляет важность признаков для данного набора данных. Параметр n_repeats задает количество случайного перемешивания признака.

Перестановочная важность признаков может быть вычислена как на обучающем наборе, так и на наборе отложенного тестирования. Использование расширенного множества позволяет выделить, какие признаки имеют высокую важность для обобщающей способности модели. Признаки, которые важны на тренировочной выборке, но не на тестовой выборке, могут привести к переобучению модели.

Важность функции перестановки оценивается как уменьшение оценки точности модели при случайном перемешивании одного признака. Функция оценки, которая будет использоваться для вычисления значимости, может быть задана с помощью аргумента scoring, который также принимает несколько оценщиков. Использование нескольких показателей более эффективно с точки зрения вычислений, чем последовательный вызов permutation_importance несколько раз с другим показателем, поскольку он повторно использует прогнозы модели [33].

Ранжирование функций примерно одинаково для разных показателей, даже если шкалы значений важности сильно различаются. Однако, это не гарантируется, и разные метрики могут привести к значительному различию значимости признаков, в частности, для моделей, обученных для

несбалансированных задач классификации, для которых выбор метрики классификации может иметь решающее значение.

Рассмотрим схему алгоритма вычисления перестановочной важности:

- Входными данными являются: обученная прогностическая модель m, набор данных D.
- Вычислить базовую оценку s модели m по данным D (например, точность для классификатора).
- Для каждого признака j(столбца D):
  - Для каждого повторения D в $1, \dots, K$:
    - Произвольно перемешать столбец j набора данных D, чтобы сгенерировать измененную версию данных с именем $\widetilde{D}_{k,j}$.
    - Вычислить оценку модели $s_{k,j}$ по поврежденным данным $\widetilde{D}_{k,j}$.
  - Вычислить важность i_j для признака f_j, определенного по формуле 1.19:

$$i_j = s - \frac{1}{K}\sum_{k=1}^{K} s_{k,j} \tag{1.19}$$

Деревья решений предоставляют альтернативную меру важности признаков, основанную на критериях информативности (MDI). Информативность определяется количественно с помощью критерия разделения деревьев решений (критерий Джини, логарифмические потери или средний квадрат ошибки [7]). Однако такой метод придаёт большое значение признакам, которые могут не поддаваться прогнозированию на отсутствующих данных, когда модель переобучена. С иной стороны, перестановочная важность признаков позволяет избегать такой проблемы, поскольку такая оценка может быть вычислена на основе прогнозируемых данных.

Кроме того, важность признаков деревьев смещена и отдает предпочтение категориальным признакам с высокой кардинальностью (обычно числовым объектам) по сравнению с признаками с низкой кардинальностью, такими как бинарные признаки или категориальные признаки с небольшим числом категорий.

Значения функций, основанные на перестановках, не проявляют такого смещения. Кроме того, важность функции перестановки может быть вычисленной метрикой производительности для прогнозов модели и может использоваться для анализа любой модели, а не только древовидных.

Когда два признака коррелируются и один из признаков перемешивается, модель по-прежнему будет иметь доступ к признаку через его коррелированный соседа. Это приводит к низкому значению важности для обоих признаков, даже

если они важны. Один из способов справляться с таким поведением - группировать коррелированные признаки и сохранять только один признак из каждого кластера [18].

1.4 Суррогатные модели

В некоторых задачах для экономии времени бывает необходимо заменить точную, но вычислительно сложную модель на быстро вычисляемую модель. Такие модели называются суррогатными, и данных подход нашел свое применение и в задаче машинного обучения. Суррогатная модель восстанавливает зависимость исходной модели с необходимой точностью, и строится по конечному множеству пар набор исходных параметров — значения выходных пронозов исходной модели для набора входных параметров [34]. Такой метод отличается от стекинга не только тем, что использует данные только одной модели для обучения новой, но и тем, что использует исходные параметры, полученные моделью. Схема суррогатных моделей приведена на рисунке 1.2.

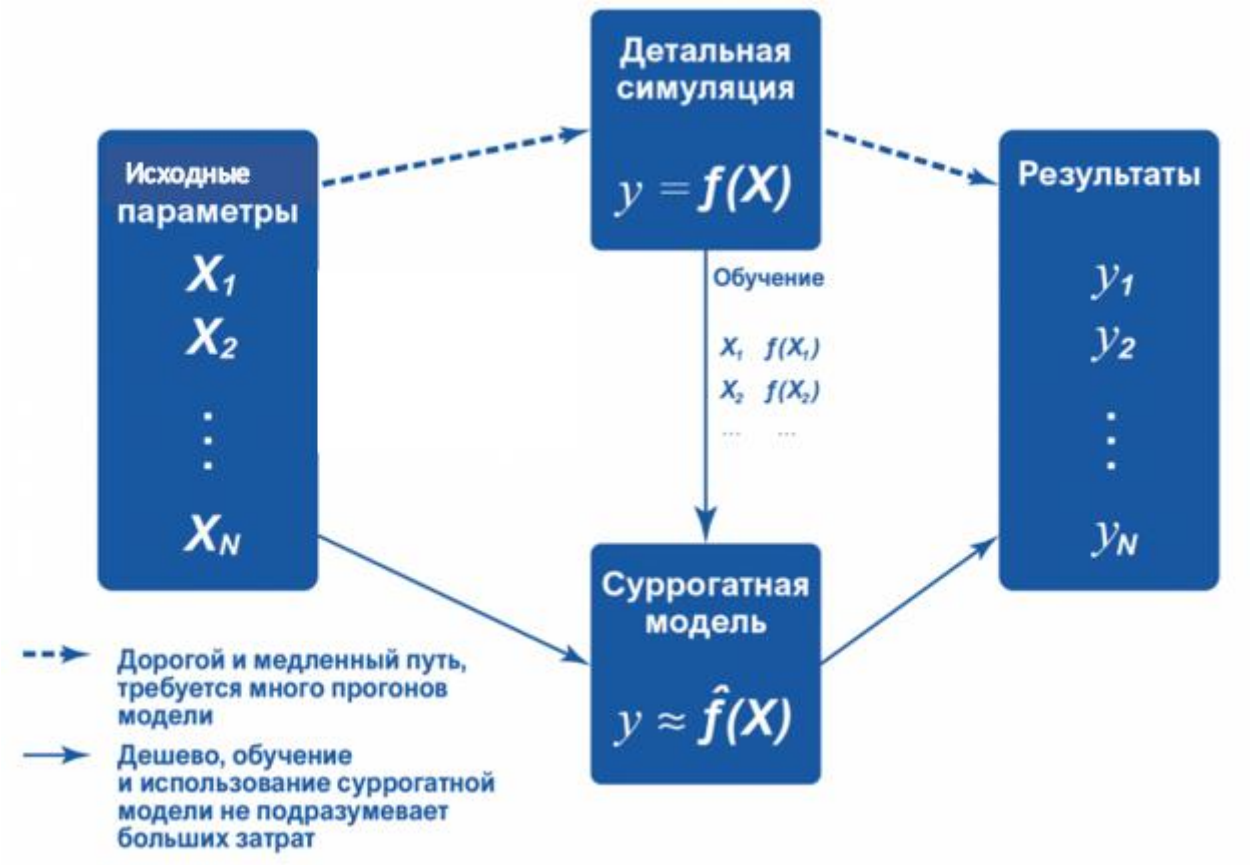

Рисунок 1.2 Схема суррогатных моделей

Пусть задана некоторая обучающая выборка S_{learn}. Кроме того, выберем некоторую структуру модели g из некоторого семейства моделей машинного обучения G (под семейством могут пониматься модели типа линейных, древовидных, нейронных сетей и т.п.), которая позволяет по S_{learn} построить предсказания модели $y = g(x, \theta)$, где θ — параметры модели. Под построением модели здесь понимается выбор семейства G и выборки S_{learn} - значений признаков θ, которые задают модель $g = g(x, \theta) \in G$, оптимальную в смысле некоторого критерия [34].

Общая схема построения суррогатной модели состоит из следующих шагов:

1) Для избавления от избыточности в признаках, проводится отбор признаков с последующим снижением размерности.

2) Декомпозиция пространства признаков на области $X = \bigcup_{j=0}^{N_y-1} X_j$, которым соответствуют различные домены значений выходов. Такая декомпозиция полезна, если требуется для разных областей обеспечить разную точность приближения.

3) Дополнительное разбиение областей X_j на связные подобласти X_{jk}, для улучшения аппроксимации на каждой из них.

4) Построение по подобластям соответствующих базовых аппроксимирующих моделей $S_{jk} = \{(x, y) \in S_{learn}: x \in X_{jk}\}$.

5) Построение кластеризатора, который относит новые точки x к подобластям X_j.

Теперь для каждой из подобластей существует прогноз определенной модели, который и будет ставиться в соответствие выходу общей суррогатной модеи. Важность признаков в таком случае будет оцениваться по теми моделями, которые выдают предсказания для подобластей, после чего такие значения некоторым образом аггрегируются.

ГЛАВА 2
МЕТОДЫ ОЦЕНКИ ЗНАЧИМОСТИ ПРИЗНАКОВ В ЗАДАЧЕ ОТБОРА ПРИЗНАКОВ

2.1 Задача отбора признаков

Признаки, которые используются для обучения модели, оказывают очень большое влияние на результаты. Неинформативные/слабо информативные признаки чаще всего понижают как точность, так и производительность многих моделей. После устранения или преобразования неинформативных или слабо информативных признаков уменьшается размер набора данных в памяти, и соответственно часто упрощается и ускоряется работа алгоритмов машинного обучения на нем, а также часто повышают точность моделей и понижают риск переподгонки.

Можно выделить 3 задачи:

- **Выбор признаков (feature selection)** – подвыборка ненужных признаков (неинформативных, слабо информативных) и выбор признаков, имеющих наиболее большое значение взаимосвязи с выходной переменной.
- **Выделение и настройка признаков (feature extraction and feature engineering)** – превращение специфических данных предметной области в понятные для модели наборы чисел;
- **Трансформация признаков (feature transformation)** – преобразование данных для повышения точности/эффективности алгоритма (например, стандартизация данных).

Каждая из указанных задач направлена на обеспечение следующих преимуществ:

- **Уменьшение риска переобучения.** Чем меньше избыточных данных, тем меньше вероятность для модели принимать решения на основе «шума».
- **Повышение точности прогнозирования модели.** Чем меньше противоречивых данных, тем выше может быть точность.
- **Сокращение необходимого времени на обучение.** Чем меньше данных, тем быстрее может обучается модель.
- **Увеличение семантического понимания модели.**

Выделяются методы ручного и автоматизированного отбора признаков.

Методы ручного отбора признаков основаны на анализе содержания каждого признака и/или их совокупности. Решение о добавлении либо исключении признака принимает исследователь.

В данном разделе будут рассмотрены некоторые из методов автоматизированного отбора признаков, которые применяются для подготовки данных. Такие методы уже могли быть реализованы, например, с помощью Python (например, в библиотеке scikit-learn). Подробное руководство по практическому использованию библиотеки scikit-learn, позволяющей проводить отбор признаков можно найти в документации к данной библиотеке, в разделе Feature selection [35].

Методы отбора признаков делятся на три большие группы:

1) методы фильтрации (filter methods) оценивают признаки на основе информации из обучающей выборки, и убирают наименее ценные из них. Такие методы применяются до запуска алгоритма обучения, на этапе предобработки;

2) обёрточные методы (wrapper methods) - классификатор запускается на некоторым образом преобразованных множествах (в частности, подмножествах исходного множества) обучающей выборки, а затем некоторым образом выбирается множество наиболее полезных для обучения признаков (рисунок 2.1);

Рисунок 2.1 - Процесс работы оберточных методов

3) встроенные методы (embedded methods) не отделяют процессы отбора и обучения (рисунок 2.2). Методы данного типа также обладают некоторыми иными достоинствами: хорошей приспособленностью к конкретной модели; отсутствием необходимости выделения специальных множеств для тестирования, как в предыдущей группе методов, и, как следствие из этого, меньший риск переобучения.

Рисунок 2.2 - Схема работы встроенных методов

Рассмотрим подробнее каждую группу.

2.2 Методы фильтрации (filter methods)

Фильтрацией в широком смысле называется любое преобразование обрабатываемых переменных с целью изменения соотношения между их различными компонентами.

Фильтры (англ. *filter methods*) измеряют численное значение релевантности признаков на основе некоторой функции, и решают по определенному правилу, зависящем от вида фильтра, какие признаки достаточно ценны в конечном множестве.

Фильтры классифицируются:

- Одномерные (англ. *univariate*) — функция, определяющая релевантность признака по сравнению с выходной переменной. В таком случае измеряется некоторое условное "качество" каждого признака с помощью статистических критериев (коэффициент ранговой корреляции Спирмена, критерий хи-квадрат и др.) и удаляют худшие. Например, библиотека scikit-learn реализует класс SelectKBest [16], реализующий выбор K лучших признаков для обучения [35]. Указанный ранее класс можно применять совместно с другими критериями отбора заданного количества признаков.
- Многомерные (англ. *multivariate*) — функция, определяющая релевантность подмножества признаков относительно целевой переменной.

Критерий χ^2 Пирсона – это метод, который оценивает значимость различий между исходами испытаний или качественных характеристик выборки, попадающих в каждую категорию, и теоретическим количеством, которое ожидается при справедливости гипотезы. Выражаясь проще, метод оценивает статистическую значимость различий нескольких относительных показателей (например, частот, долей) [36].

Критерий хи-квадрат был разработан и предложен в 1900 году английским математиком, статистиком, биологом и философом, основателем математической статистики и одним из основоположников биометрики Карлом Пирсоном (1857-1936) [36].

Условия и ограничения применения критерия хи-квадрат Пирсона:

1. Сопоставляемые значения должны быть измерены в номинальной или порядковой шкале (например, оценки знаний учащихся, принимающая значения 2 и от 4 до 10).
2. Соответствующие группы должны быть независимыми, то есть критерий хи-квадрат не должен применяться при сравнении, например, доапостериорных и апостериорных наблюдений. При сравнении двух

связанных совокупностей проводится **тест** Мак-Немара, а в случае сравнения трех и более групп - Q-критерий Кохрена.

3. Независимость наблюдений (отбор объектов из генеральной совокупности должен производится независимо друг от друга).

Рассмотрим алгоритм расчета критерия хи-квадрат Пирсона

1. Рассчитывается ожидаемое количество наблюдений для каждой из ячеек таблицы сопряженности (при выполнении условия 2) по формуле 2.1.

$$E_{ij} = \frac{\sum_i A_{ij} * \sum_j A_{ij}}{\sum_i \sum_j A_{ij}} \quad (2.1)$$

2. Находится значение критерия χ^2 по формуле 2.2:

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{\left(O_{ij} - E_{ij}\right)^2}{E_{ij}}, \quad (2.2)$$

где i – номер строки (от 1 до r), j – номер столбца (от 1 до c), O_{ij} – фактическое количество наблюдений в ячейке ij, E_{ij} – ожидаемое число наблюдений в ячейке ij, рассчитанное в пункте 1.

3. Определяем число степеней свободы как произведение уменьшенного на единицу числа рядов на уменьшенное на единицу числа столбцов таблицы сопряженности.
4. Сравниваем значение критерия χ^2 с критическим значением при полученном числе степеней свободы по таблице.

В том случае, если полученное значение критерия χ^2 больше критического, делаем вывод о наличии статистической взаимосвязи между изучаемыми признаками с соответствующим уровнем значимости [36]. Также существуют модификации данного метода для работы с непрерывными шкалами.

t-критерий Стьюдента – общее название класса методов статистической проверки гипотез (статистических критериев), основанных на распределении Стьюдента. Наиболее частые случаи применения такого критерия связаны с проверкой равенства средних значений в двух выборках.

Данный критерий был разработан Уильямом Сили Госсетом для оценки качества пива в компании Гиннесс. В связи с обязательствами перед компанией по неразглашению коммерческой тайны, статья Госсета вышла в 1908 году в журнале «Биометрика» под псевдонимом «Student» (Студент), что и дало название такому распределению.

t-критерий Стьюдента может применяться как в случаях сравнения независимых выборок, так и при сравнении связанных совокупностей, что

выгодно отличает его от предыдущего описанного метода. В последнем случае рассчитывается парный t-критерий Стьюдента [36].

Для применения t-критерия Стьюдента необходимо, чтобы исходные данные имели нормальное распределение. Для увеличения гауссоподобности ненормальных распределений может использоваться преобразование мощности – параметрическое монотонное преобразование, нацеленные на отображение данных из любого распределения в гауссовское. Также имеет значение равенство дисперсий сравниваемых групп (гомоскедастичность). При неравных дисперсиях применяется t-критерий в модификации Уэлча (Welch's t) [36].

При отсутствии нормального распределения сравниваемых выборок вместо t-критерия Стьюдента также могут использоваться методы непараметрической статистического анализа, например, U-критерий Манна — Уитни [36].

Для сравнения средних величин t-критерий Стьюдента рассчитывается по формуле 2.3:

$$t = \frac{M_1 - M_2}{\sqrt{m_1^2 + m_2^2}}, \tag{2.3}$$

где $\mathbf{M_1}$ - среднее арифметическое первой сравниваемой группы, $\mathbf{M_2}$ - среднее арифметическое второй сравниваемой группы, $\mathbf{m_1}$ - среднее отклонение от $\mathbf{M_1}$, $\mathbf{m_2}$ - среднее отклонение от $\mathbf{M_2}$.

Также необходимо знать количество объектов в каждой группе (n_1 и n_2), потому что число степеней свободы **f** находится по следующей формуле:

$$f = (n_1 + n_2) - 2 \tag{2.4}$$

После этого определяется критическое значение t-критерия Стьюдента для требуемого уровня значимости (например, p=0,05) и при данном числе степеней свободы **f** по таблице [36].

Если рассчитанное значение t-критерия Стьюдента равно или больше критического, найденного по таблице, делаем вывод о статистической значимости различий между сравниваемыми величинами.

Парный t-критерий Стьюдента – одна из модификаций метода Стьюдента, используемая для определения статистической значимости различий парных (повторных) измерений [36].

Зависимыми являются объекты с малым расстоянием между друг другом в признаковом пространстве. Нулевая гипотеза в таком случае будет об отсутствии статистически важных различий между выборками, альтернативная – о наличии [36].

Как и в случае сравнения независимых выборок, для применения парного t-критерия необходимо, чтобы исходные данные имели нормальное распределение. При несоблюдении этого условия для сравнения выборочных средних должны использоваться методы непараметрической статистики, такие как G-критерий знаков или Т-критерий Уилкоксона [36]. Также могут использоваться уже указанные выше методы отображения распределения в гауссовское.

Парный t-критерий может использоваться только при сравнении двух выборок. Если необходимо сравнить три и более похожих объектов, следует использовать однофакторный дисперсионный анализ (ANOVA) для повторных измерений.

Парный t-критерий Стьюдента рассчитывается по формуле 2.5:

$$t = \frac{M_d}{\sigma_d/\sqrt{n}}, \tag{2.5}$$

где M_d - среднее арифметическое разностей значений признаков, у близких объектов, σ_d - среднее квадратическое отклонение разностей показателей, n - число исследуемых объектов.

Интерпретация полученного значения парного t-критерия Стьюдента не отличается от оценки t-критерия для несвязанных совокупностей. Прежде всего, необходимо найти число степеней свободы f по формуле 2.6:

$$f = n - 1 \tag{2.6}$$

После этого определяем критическое значение t-критерия Стьюдента для требуемого уровня значимости (например, p <0,05) и при данном числе степеней свободы **f** по таблице [37].

Критерий корреляции Пирсона – это метод параметрической статистики, позволяющий определить наличие линейной связи между двумя признаками, а также оценить ее тесноту и статистическую значимость. В статистических расчетах и выводах коэффициент корреляции обычно обозначается как r_{xy} или R_{xy} [36].

Критерий корреляции Пирсона был разработан командой британских ученых во главе с Карлом Пирсоном (1857-1936) в 90-х годах 19-го века, для упрощения анализа ковариации двух случайных величин. Помимо Карла Пирсона над критерием корреляции Пирсона работали также Фрэнсис Эджуорт и Рафаэль Уэлдон.

Критерий корреляции Пирсона позволяет определить, какова линейность связи между двумя признаками, измеренными в абсолютной шкале. При помощи

дополнительных расчетов можно также определить, насколько статистически значима выявленная связь [36].

Условия и ограничения применения критерия корреляции Пирсона**:**

1. Сопоставляемые показатели должны быть измерены в количественной шкале (например, частота сердечных сокращений, температура тела, содержание лейкоцитов в 1 мл крови, систолическое артериальное давление).
2. Посредством критерия корреляции Пирсона можно определить лишь наличие и силу линейной взаимосвязи между величинами. Прочие характеристики связи, в том числе направление (прямая или обратная), характер изменений (прямолинейный или криволинейный), а также наличие зависимости одной переменной от другой - определяются в ходе регрессионного анализа.
3. Количество сопоставляемых признаков должно быть равно двум. В случае анализ взаимосвязи трех и более признаков следует воспользоваться методом факторного анализа.
4. Критерий корреляции Пирсона является параметрическим, в связи с чем условием его применения служит нормальное распределение каждой из сопоставляемых переменных. В случае необходимости корреляционного анализа показателей, распределение которых отличается от нормального, в том числе измеренных в порядковой шкале, следует использовать коэффициент ранговой корреляции Спирмена. Также допустимо использований преобразований мощности.

Зависимость величин является достаточным условием для наличия корреляционной связи между ними, но не наоборот.

Расчет коэффициента корреляции Пирсона производится по следующей формуле:

$$r_{xy} = \frac{\sum(d_x * d_y)}{\sqrt{\sum d_x^2 * \sum d_y^2}} \quad (2.7)$$

Значения коэффициента корреляции Пирсона интерпретируются исходя из его абсолютных значений. Возможные значения коэффициента корреляции варьируют от 0 до ±1. Чем больше значение модуля r_{xy} – тем выше линейность связи между величинами. $r_{xy} = 0$ говорит о независимости признаков. $r_{xy} = \pm 1$ – свидетельствует о наличии полностью линейной связи. Если значение критерия корреляции Пирсона оказалось больше 1 или меньше -1 – в расчетах допущена ошибка [36].

Для оценки тесноты, или силы, корреляционной связи обычно используют общепринятые критерии, согласно которым абсолютные значения $r_{xy} < 0.3$ свидетельствуют о *слабой* связи, значения r_{xy} от 0.3 до 0.7 - о связи *средней* тесноты, значения $r_{xy} > 0.7$ - о *сильной* связи.

Более точную оценку силы корреляционной связи можно получить, если воспользоваться таблицей Чеддока (таблица 2.1):

Таблица 2.1 Оценка силы корреляционной связи

Абсолютное значение r_{xy}	Теснота (сила) корреляционной связи
менее 0.3	слабая
от 0.3 до 0.5	умеренная
от 0.5 до 0.7	заметная
от 0.7 до 0.9	высокая
более 0.9	весьма высокая

Оценка статистической значимости коэффициента корреляции r_{xy} осуществляется при помощи t-критерия, рассчитываемого по формуле 2.8:

$$t_r = r_{xy}\frac{\sqrt{n-2}}{\sqrt{1-r_{xy}^2}} \quad (2.8)$$

Полученное значение t_r сравнивается с критическим значением при определенном уровне значимости и числе степеней свободы n-2. Если t_r превышает $t_{крит}$, то делается вывод о статистической значимости выявленной корреляционной связи [36].

Коэффициент ранговой корреляции Спирмена – это непараметрический метод, который используется с целью статистического изучения связи между явлениями. [36].

Данный критерий был разработан и предложен для проведения корреляционного анализа в 1904 году Чарльзом Эдвардом Спирменом, английским психологом, профессором Лондонского и Честерфилдского университетов.

Коэффициент ранговой корреляции Спирмена используется для выявления и оценки тесноты связи между двумя рядами сопоставляемых количественных показателей - признаков. В том случае, если ранги значений объектов, упорядоченных по степени возрастания или убывания, в большинстве случаев совпадают или являются близкими (большему значению одного показателя

соответствует большее значение другого показателя - например, при сопоставлении роста пациента и его массы тела), делается вывод о наличии прямой корреляционной связи. Если ранги показателей имеют противоположную направленность (большему значению одного показателя соответствует меньшее значение другого - например, при сопоставлении возраста и частоты сердечных сокращений), то говорят об **обратной** связи между показателями [37].

Коэффициент корреляции Спирмена обладает следующими свойствами:

1. Коэффициент корреляции может принимать значения от минус единицы до единицы, причем при rs=1 имеет место строго прямая связь, а при rs= -1 – строго обратная связь.
2. Если коэффициент корреляции отрицательный, то имеет место обратная связь, если положительный, то – прямая связь.
3. Если коэффициент корреляции равен нулю, то связь между величинами отсутствует.
4. Чем ближе модуль коэффициента корреляции к единице, тем более сильной является связь между измеряемыми величинами.

Условия использования коэффициента ранговой корреляции Спирмена:

- В связи с тем, что коэффициент является методом непараметрического анализа, проверка на нормальность распределения не требуется.
- Сопоставляемые показатели могут быть измерены как в непрерывной шкале, так и в порядковой (например, баллы экспертной оценки от 1 до 5).

Эффективность и качество оценки методом Спирмена снижается, если разница между различными значениями какой-либо из измеряемых величин достаточно велика. Не рекомендуется использовать коэффициент Спирмена, если имеет место неравномерное распределение значений измеряемой величины [36].

Расчет коэффициента ранговой корреляции Спирмена включает следующие этапы:

1. Сопоставить каждому из признаков их порядковый номер (ранг) по возрастанию или убыванию.
2. Определить разности рангов каждой пары сопоставляемых значений (d).
3. Возвести в квадрат каждую разность и суммировать полученные результаты.
4. Вычислить коэффициент корреляции рангов по формуле 2.9:

$$\rho = 1 - \frac{6 * \sum d^2}{n(n^2 - 1)} \qquad (2.9)$$

5. Определить статистическую значимость коэффициента при помощи t-критерия, рассчитанного по формуле 2.10

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \tag{2.10}$$

Для оценки тесноты связи может использоваться шкала Чеддока (таблица 2.1, страница 28)

Статистическая значимость полученного коэффициента оценивается при помощи t-критерия Стьюдента. Если рассчитанное значение t-критерия меньше табличного при заданном числе степеней свободы, статистическая значимость наблюдаемой взаимосвязи - отсутствует. Если больше, то корреляционная связь считается статистически значимой [36].

Преимуществом группы фильтров является низкая трудоёмкость вычисления релевантности признаков в наборе данных, но недостатком в таком подходе является игнорирование возможных зависимостей между признаками.

2.3 Обёрточные методы (Wrapper methods)

Принцип работы обёрточных методов [5]:

- исходное пространство признаков некоторым образом преобразуется;
- выполняется поиск по преобразованному пространству множества признаков;
- на каждой итерации поиска извлекается информация о некотором качестве обучения на текущем множестве признаков (в качестве функции оценки используется, например, точность классификатора)
- в зависимости от метода преобразования проводится отбор наименее важных признаков

Указанный процесс является циклическим и продолжается до тех пор, пока не будут достигнуты заданные условия останова. Методы данной группы учитывают более сложные зависимости между признаками и показывают большую точность, что является преимуществом по сравнению с фильтрами, но требуют большего количества вычислений, и, соответственно, времени выполнения, и повышается риск переобучения [5].

Существует несколько типов оберточных методов:

- детерминированные, которые изменяют множество признаков по определенному детерминированному правилу;

- рандомизированные, которые используют, например, генетические алгоритмы для выбора искомого множества признаков.

Среди детерминированных алгоритмов самыми простыми являются алгоритмы последовательного поиска [5]:

- SFS (Sequential Forward Selection – прямой последовательный отбор) — жадный алгоритм, который начинает с пустого множества признаков, на каждом шаге добавляя признак, обучение с которым показывает наилучшие результаты. На первом этапе качество модели оценивается по отношению к каждому признаку. На втором этапе – наилучшее сочетание из двух признаков, и так далее. Процесс продолжается до тех пор, пока не будет выполнено условие останова (например, выбрано указанное количество признаков).
- SBS (Sequential Backward Selection – обратный последовательный отбор) — алгоритм обратный прямому, который начинает с изначального множества признаков, и удаляет по одному или несколько худших признаков на каждом шаге. Следует отметить, что эмпирически обратный алгоритм обычно показывает лучшие результаты по сравнению с прямым жадным алгоритмом. Это связано с тем, что обратный жадный алгоритм учитывает признаки, информативные в совокупности, но неинформативные, если рассматривать их по отдельности.

На каждом шаге описанных алгоритмов для оценки качества обычно используются статистические критерии такие, как U-критерий Манна-Уитни, t-критерий Стъюдента, критерий Уилкоксона и иные. Сам алгоритм при этом принимает форму последовательности тестов.

U-критерий Манна-Уитни – непараметрический статистический критерий, используемый для сравнения двух независимых выборок по уровню какого-либо признака, измеренного количественно. Метод основан на определении того, насколько мала зона пересекающихся значений между двумя рядами (ранжированным рядом значений признака в первой и во второй выборке). Чем меньше значение критерия, тем вероятнее, что различия между значениями параметра в выборках статистически значимы [36].

Данный метод выявления различий между выборками был предложен в 1945 году американским химиком и статистиком Фрэнком Уилкоксоном. В 1947 году он был существенно переработан и расширен математиками Х.Б. Манном (H.B. Mann) и Д.Р. Уитни (D.R. Whitney), по именам которых сегодня обычно и называется.

U-критерий Манна-Уитни является непараметрическим критерием, поэтому, в отличие от t-критерия Стьюдента, не требует наличия нормального распределения сравниваемых совокупностей. U-критерий подходит для

сравнения малых выборок: в каждой из выборок должно быть не менее 3 значений признака. Допускается, чтобы в одной выборке было 2 значения, но во второй тогда должно быть не менее пяти [36].

Условием для применения U-критерия Манна-Уитни является отсутствие или малое число в сравниваемых группах совпадающих значений признака.

Аналогом U-критерия Манна-Уитни для сравнения трех и более групп является Критерий Краскела-Уоллиса.

Рассмотрим схему расчета U-критерия Манна-Уитни [36]:

Сначала из обеих сравниваемых выборок составляется **единый ранжированный ряд**, по правилам преобразования рядов в ранжированные ряды.

В составленном ряду общее количество рангов получится равным:

$$N = n_1 + n_2$$

где n_1 - количество элементов в первой выборке, а n_2 - количество элементов во второй выборке.

Далее вновь разделяем единый ранжированный ряд на два, состоящие соответственно из единиц первой и второй выборок, запоминая при этом значения рангов для каждой единицы. Подсчитываем отдельно сумму рангов, пришедшихся на долю элементов первой выборки, и отдельно - на долю элементов второй выборки. Определяем большую из двух ранговых сумм (T_x) соответствующую выборке с n_x элементами.

Наконец, находим значение U-критерия Манна-Уитни по формуле 2.11:

$$U = n_1 * n_2 + n_x * \frac{n_x + 1}{2} - T_x \qquad (2.11)$$

Полученное значение U-критерия сравниваем по таблице для избранного уровня статистической значимости с критическим значением U при заданной численности сопоставляемых выборок: если полученное значение U меньше табличного или равно ему, то признается статистическая значимость различий между уровнями признака в рассматриваемых выборках. Достоверность различий тем выше, чем меньше значение U.

Критерий Уилкоксона для связанных выборок (также используются названия Т-критерий Уилкоксона, критерий Вилкоксона, критерий знаковых рангов Уилкоксона, критерий суммы рангов Уилкоксона) – непараметрический статистический критерий, используемый для сравнения двух связанных (парных) выборок по уровню какого-либо количественного признака, измеренного в непрерывной или в порядковой шкале [36].

Суть метода состоит в том, что сопоставляются абсолютные величины выраженности сдвигов в том или ином направлении. Для этого сначала все

абсолютные величины сдвигов ранжируются, а потом суммируются ранги. Если сдвиги в ту или иную сторону происходят случайно, то и суммы их рангов окажутся примерно равны. Если же интенсивность сдвигов в одну сторону больше, то сумма рангов абсолютных значений сдвигов в противоположную сторону будет значительно ниже, чем это могло бы быть при случайных изменениях [36].

Тест был впервые предложен в 1945 году американским статистиком и химиком Фрэнком Уилкоксоном (1892-1965). В той же научной работе автором был описан еще один критерий, применяемый в случае сравнения независимых выборок.

Т-критерий Уилкоксона используется для оценки различий между двумя рядами измерений, выполненных для близких объектов признакового пространства. Данный тест способен выявить направленность и выраженность изменений - то есть, являются ли показатели больше сдвинутыми в одном направлении, чем в другом [36].

Классическим примером ситуации, в которой может применяться Т-критерий Уилкоксона для связанных совокупностей, является исследование "до-после", когда сравниваются показатели до и после лечения.

Условия и ограничения применения Т-критерия Уилкоксона:

1. Критерий Уилкоксона является непараметрическим критерием, поэтому, в отличие от парного t-критерия Стьюдента, не требует наличия нормального распределения сравниваемых совокупностей.
2. Число объектов при использовании Т-критерия Уилкоксона должно быть не менее 5.
3. Изучаемый признак может быть измерен как в количественной непрерывной, так и в порядковой шкале.
4. Данный критерий используется только в случае сравнения двух рядов измерений. Аналогом Т-критерия Уилкоксона для сравнения трех и более связанных совокупностей является Критерий Фридмана.

Рассмотрим схему рассчета Т-критерия Уилкоксона для связанных выборок.

1. Вычислить разность между значениями парных измерений для каждого объекта. Нулевые сдвиги далее не учитываются.
2. Определить, какие из разностей являются типичными, то есть соответствуют преобладающему по частоте направлению изменения показателя.
3. Перевести разности пар по их абсолютным значениям (то есть, без учета знака), в порядке возрастания в ранжированный ряд. Меньшему абсолютному значению разности приписывается меньший ранг.
4. Рассчитать сумму рангов, соответствующих нетипичным сдвигам.

Таким образом, Т-критерий Уилкоксона для связанных выборок рассчитывается по формуле 2.12:

$$T = \sum R_r \tag{2.12}$$

где ΣR_r - сумма рангов, соответствующих нетипичным изменениям показателя.

Полученное значение Т-критерия Уилкоксона сравниваем с критическим по таблице для избранного уровня статистической значимости при заданной численности сопоставляемых выборок n [37]:

Если расчетное (эмпирическое) значение $T_{эмп.}$ меньше табличного $T_{кр.}$ или равно ему, то признается статистическая значимость изменений показателя в типичную сторону (принимается альтернативная гипотеза). Достоверность различий тем выше, чем меньше значение Т.

Популярным оберточным методом также является RFE (Recursive Feature Elimination, рекурсивное исключение признаков), который также относят к встроенным методам отбора. Данный метод работает итеративно: начиная с полного множества признаков обучает классификатор, ранжирует признаки по весам, которые им присвоил классификатор, убирает какое-то число признаков и повторяет процесс с оставшегося подмножества признаков, если не было достигнуто их требуемое количество [5].

2.4 Встроенные методы (embedded methods)

Основным достоинством встроенных методов является необходимость в меньшем числе вычислений, чем для обёрточных методов (хотя и больше, чем для методов фильтрации).

Для отбора признаков непосредственно используется структура некоторой модели машинного обучения. В оберточных методах классификатор служит только для оценки работы на данном множестве признаков, тогда как встроенные методы используют некоторую информацию о признаках, которую классификаторы используют во время обучения.

Основным методом из этой категории является **регуляризация** [3].

Регуляризация — это добавление штрафа за сложность модели, добавляемый к функции потерь или иным методом ухудшающее оценку модели, который позволяет вести учёт слабо информативных признаков. Ошибочным будет и мнение, что регуляризация бывает только в линейных моделях. Пример регуляризации для бинарных деревьев описан в пункте 1.1.

Существуют различные разновидности регуляризации, однако основной принцип общий. Идея регуляризации в том, чтобы построить алгоритм,

минимизирующий не только ошибку, но и количество используемых переменных, уменьшая, в том числе, и сложность модели.

Данная группа методов часто применяется, когда много якобы независимых переменных имеют друг с другом сильную связь (т.е. имеет место мультиколлинеарность). Следствием этого, например, для линейных моделей, является плохая обусловленность матрицы $X^T X$ и неустойчивость оценок коэффициентов регрессии. Коэффициенты регрессии, например, могут иметь неправильный знак или значения, которые намного превосходят те, которые приемлемы из практических соображений.

Существует большое количество видов регуляризации в линейных моделях, которые позволяют уменьшить размерность данных и устранить/смягчить проблему переобучения. Рассмотрим подробнее два способа:

- L1-регуляризация — добавляет штраф к сумме абсолютных значений коэффициентов (формула 2.12). Данный метод получил название **Лассо-регрессии**. В процессе работы алгоритма величина приписанных алгоритмом коэффициентов окажется пропорциональна важности соответствующих переменных для классификации, а для переменных, которые дают наименьший вклад в устранение ошибки, коэффициенты станут нулевыми. Таким образом, более значимые признаки сохранят свои коэффициенты ненулевыми, а менее значимые – обнулятся. Стоит также отметить, что большие по модулю отрицательные значения коэффициентов тоже говорят о сильном влиянии.

$$J_{lasso} = \min_w \|Xw - y\|_2^2 + \alpha\|w\|_1^1 \qquad (2.12)$$

- L2-регуляризация — добавляет штраф к сумме квадратов коэффициентов (формула 2.13). Указанный выше метод называется **ридж**-регрессией.

$$J_{ridge} = \min_w \|Xw - y\|_2^2 + \alpha\|w\|_2^2 \qquad (2.13)$$

Уделим больше внимания параметру альфа. Он позволяет определять вклад регуляризации в общую сумму. С его помощью указывается приоритет — точность модели или минимальное количество используемых переменных. Несмотря на незначительность различия, свойства отличаются. Если в ridge-регрессии по мере роста альфа все коэффициенты приближаются к нулевым, то в LASSO-регрессии с ростом альфа все больше коэффициентов становятся нулевыми и перестают влиять на модель. Более значимые признаки продолжат вносить вклад, менее значимые — будут иметь нулевые коэффициенты [2,3,12].

Методы преобразования признакового пространства некоторым образом преобразуют исходные признаки в новые, все также описывающие

признаковое пространство датасета, но уменьшая его размерность и теряя в интерпретируемости данных, т.к. становится непонятно, за что отвечают новые признаки, даже, если известна формула их вычисления. Все методы этой группы можно разделить на **линейные** и **нелинейные**.

Одним из самых известных методов **линейного** выделения признаков является PCA (Principal Component Analysis, рус. *метод главных компонент*). Основной идеей этого метода является поиск такой гиперплоскости, на которую при ортогональной проекции всех признаков максимизируется дисперсия, являющаяся одной из основных метрик информативности признака. Такое преобразование может быть произведено, например, с помощью сингулярного разложения матриц и создает проекцию только на линейные многомерные плоскости, поэтому и метод находится в категории линейных [5].

ГЛАВА 3
ПРИМЕНЕНИЕ МЕТОДОВ ОЦЕНКИ ВАЖНОСТИ ПРИЗНАКОВ В РЕАЛЬНОЙ ЗАДАЧЕ

3.1 Описание задачи

Рассмотрим применение оценок важности признаков на датасете для задачи [8] – предсказания, зарабатывает ли человек больше определенного порога. Загрузим библиотеки и данные (число объектов в датасете – 32562, вместо текстовых значений целевой переменной используем метки 0 и 1: ‘<=50K’ = 0, ’>50K’ = 1), для удобства оставив только численные признаки, такие как:

- age – возраст человека
- fnlwgt (final weight) – оценка числа людей, которое представляет каждая строка данных
- educational-num – длительность получения образования
- capital-gain – прирост годового капитала
- capital-loss – потеря годового капитала
- hours-per-week – количество рабочих часов в неделю

3.2 Обучение моделей

Все кросс-валидации – 5-кратные перекрестные с сохранением приблизительной частоты появления классов в выборках и перемешиванием с жестко заготовленным случайным распределением для повторяемости [13]. Используемые далее обозначения: train_scores – оценки точности классификатора на каждой из тренировочных выборок, test_scores – на каждой из тестовых выборок, mean_score – математическое ожидание предыдущей величины. Для стабилизации дисперсии будем использовать класс-препроцессор данных PowerTransformer [14] по методу Йео-Джонсона (Yeo-Johnson) [15]. Коэффициенты линейных моделей нормируются на их сумму.

Оценка точности классификатора на кросс-валидации для случайного леса:
train scores = [0.99998025 0.99997231 0.99997693 0.9999678 0.99997414]
mean score = 0.99997 +/- 0.00000
test score = [0.82427915 0.82290796 0.83106668 0.8192637 0.83155106]
mean score = 0.82581 +/- 0.00478

Важность признаков для случайного леса (рисунок 3.1):

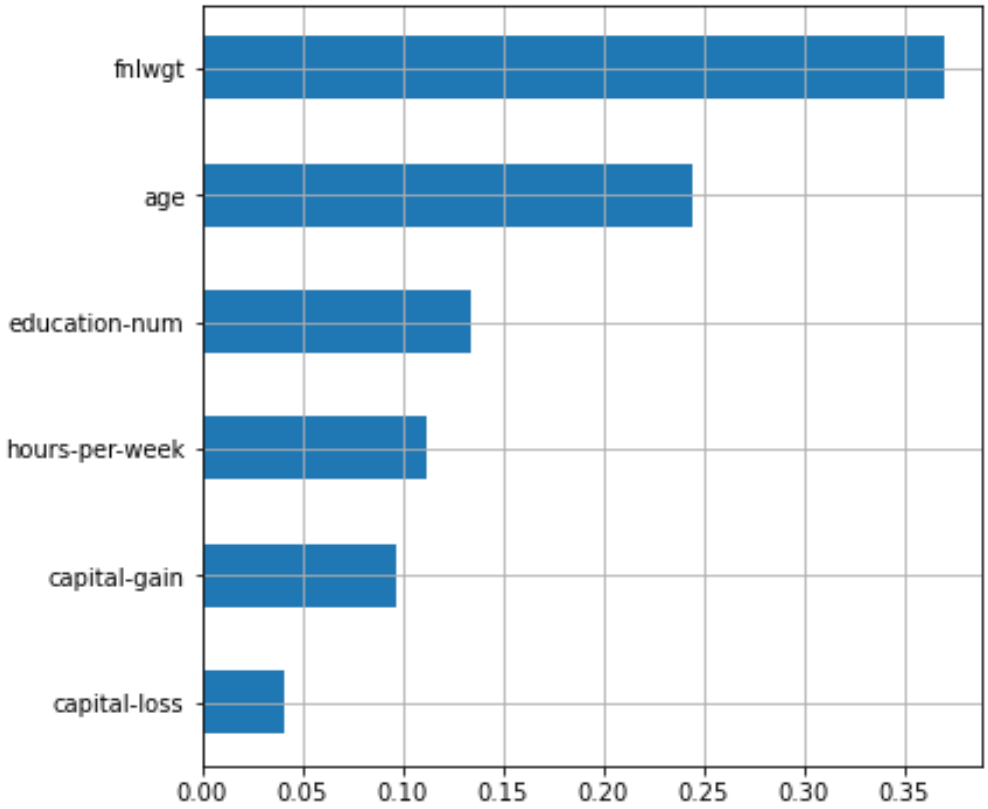

Рисунок 3.1. – Важность признаков случайного леса

Самым важным признаком для случайного леса [10] стал fnlwgt. Это можно интерпретировать как то, что самым важным признаком того, что человек зарабатывает больше определенного порога является количество людей с такими же характеристиками. Происходит это потому, что как деревья решений, так и случайные леса могут выдавать смещённую оценку важности признаков [6]. Притом, чем выше риск переобучения модели, тем выше риск получить более высокое смещение, поэтому доверять оценкам таких моделей надо с осторожностью. В данном случае видно, что модель переобучена (математическое ожидание точности классификатора случайного леса на тестовой выборке составило 99%, а на тестовой – всего 82%) и выдает смещенные оценки.

Рассмотрим оценку точности с помощью суррогатных моделей:

Оценка точности классификатора на кросс-валидации для суррогата случайного леса:

train scores = [0.99997586 0.99998316 0.9999789 0.99997561 0.99997187]
mean score = 0.99998 +/- 0.00000
test score = [0.86122731 0.86878913 0.87685501 0.86725615 0.86313271]
mean score = 0.86745 +/- 0.00543

Важность признаков для суррогата случайного леса (рисунок 3.2):

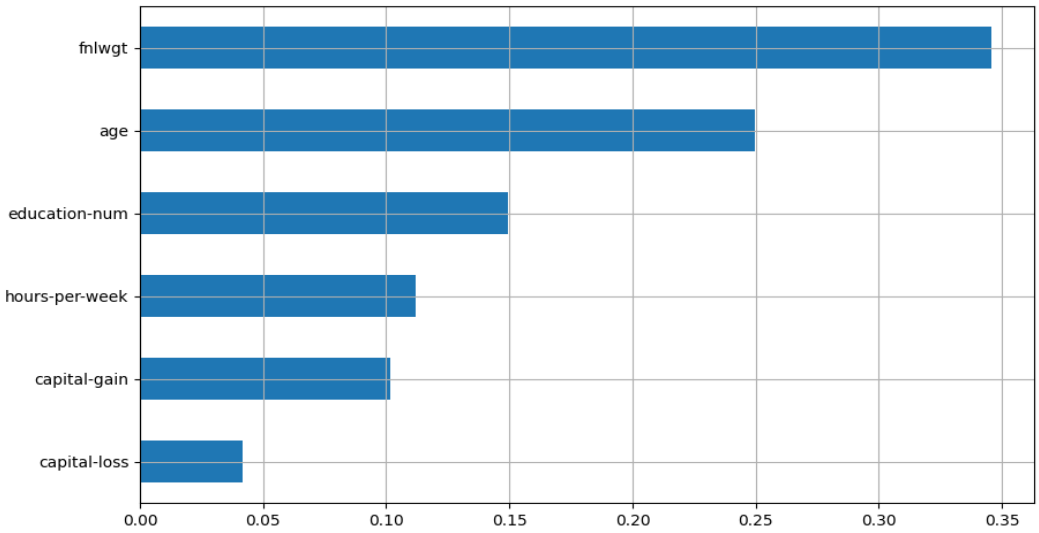

Рисунок 3.2 Оценка важности признаков суррогатной моделью случайного леса.

Заметим, что оценка параметра fnlwgt упала, а практически всех остальных – выросла, хотя их порядок и был сохранен. Это показывает хорошую согласованность оценок суррогатных моделей с оценками исходных моделей. Методом прямого последовательного отбора признаков отберем наилучшее подмножество из 5 признаков (рисунок 3.3):

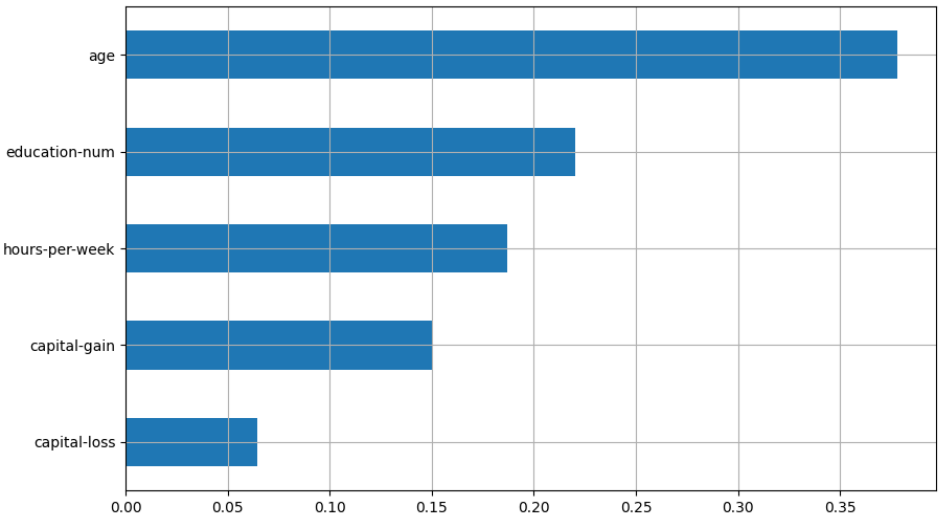

Рисунок 3.3. Важность признаков для метода прямого последовательного отбора признаков для случайного леса

Заметим, что наименее важным признаков оказался fnlwgt, а остальные признаки сохранили отношение порядка.

Оценим важность признаков методом перестановочной важности (рисунок 3.4):

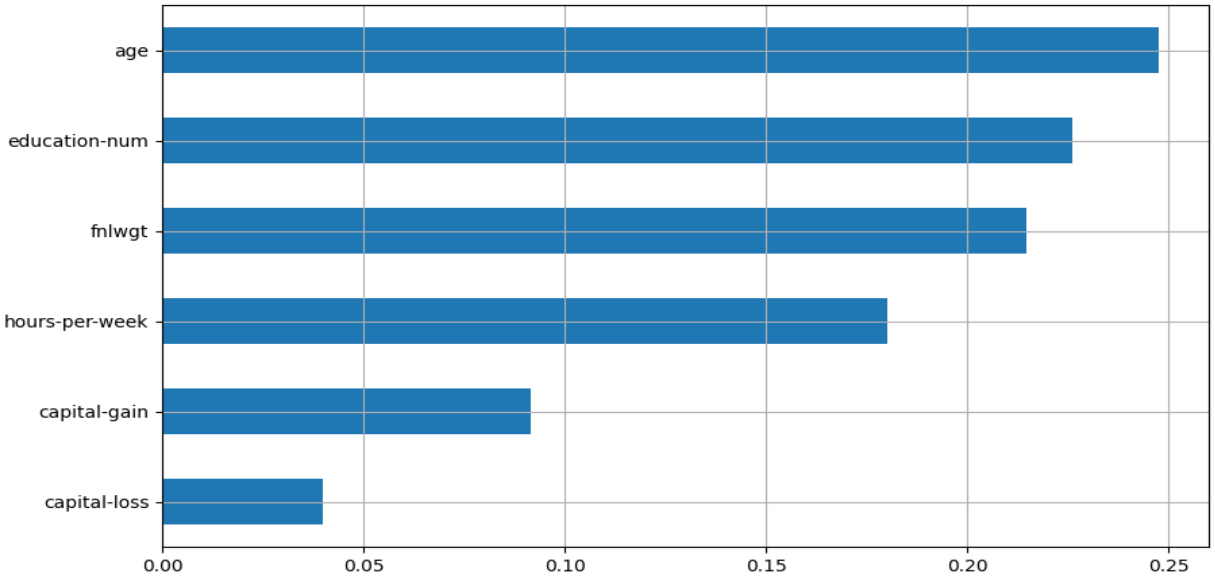

Рисунок 3.4 Оценка важности признаков методом перестановочной важности

Отметим, что параметр fnlwgt получил значительно меньшую оценку, чем при оценке случайным лесом, хоть и остался в тройке лидирующих по важности. Оставшиеся признаки также показывают сохранение отношения порядка, полученных иными методами.

Далее опишем предлагаемую модификацию метода определения важности признаков, основанную на определении важности с помощью перестановочной важности, рассмотренной ранее. В частности, вот так была изменена формула подсчета важности каждого отдельного признака:

$$\left(s - \frac{1}{K}\sum_{k=1}^{K} s_{k,j} -> s - \frac{1}{K}\sqrt{\sum_{k=1}^{K} {s_{k,j}}^2 - \sum_{k=1}^{K} {s_{k,j}}^2} \right)$$

Рассмотрим применение данного метода на практике (рисунок 3.5).

В отличие от обычной перестановочной важности признаков, её модифицированная версия склонна к увеличению важности параметров, которые были оценены низко другими метриками. Так же было сохранено отношение порядка, полученного оригинальным методом, что показывает согласованность данной разработки с показанными ранее методами.

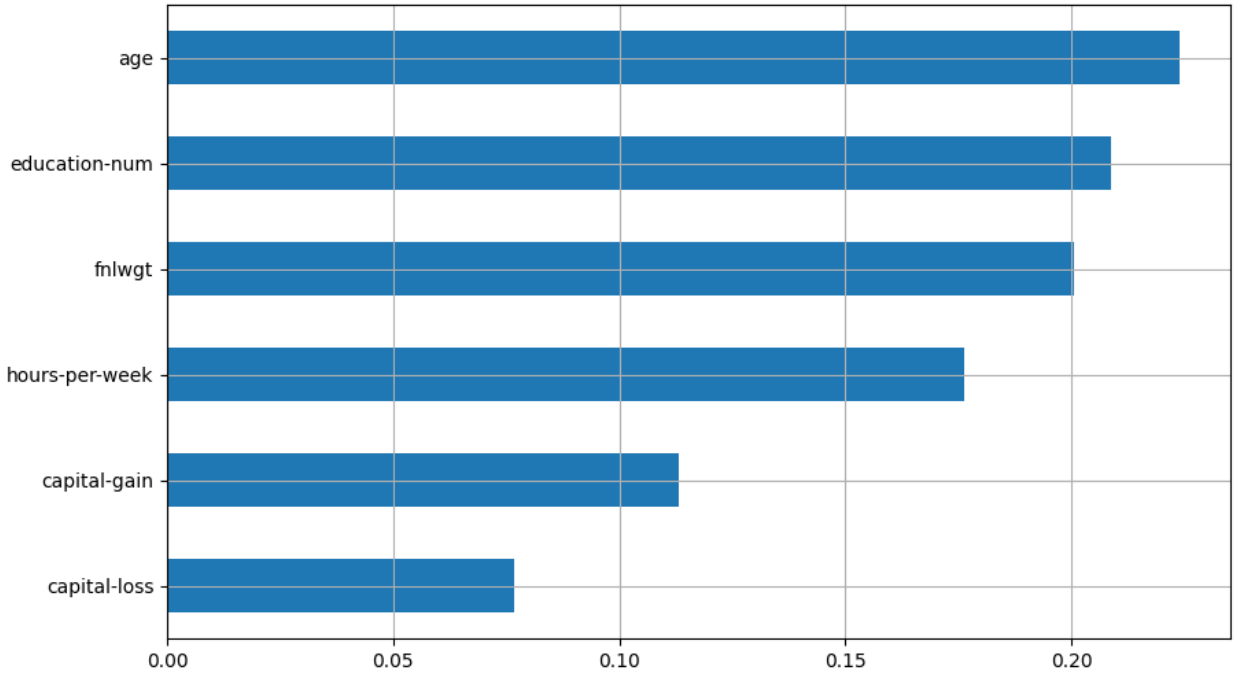

Рисунок 3.5 Важность признаков, оценённых модифицированным методом

Точность на кросс-валидации и коэффициенты регрессии для логистических регрессий:

Повторим процедуру для линейной модели (с L1-регуляризацией).

Точность на кросс-валидации для линейной модели с L1-регуляризацией:

train scores = [0.82988457 0.82727422 0.82610092 0.83029481 0.82600082]

mean score = 0.82791 +/- 0.00184

test score = [0.82034993 0.83000963 0.8348707 0.81787667 0.83548066]

mean score = 0.82772 +/- 0.00732

Для линейной модели с L2-регуляризацией:

Точность на кросс-валидации для линейной модели с L2-регуляризацией:

train scores = [0.82988494 0.82727349 0.82610074 0.83029575 0.82600163]

mean score = 0.82791 +/- 0.00184

test score = [0.82034877 0.83001208 0.83487096 0.81787925 0.83548092]

mean score = 0.82772 +/- 0.00732

Коэффициенты регрессии для линейной модели с L1-регуляризацией (рисунок 3.6):

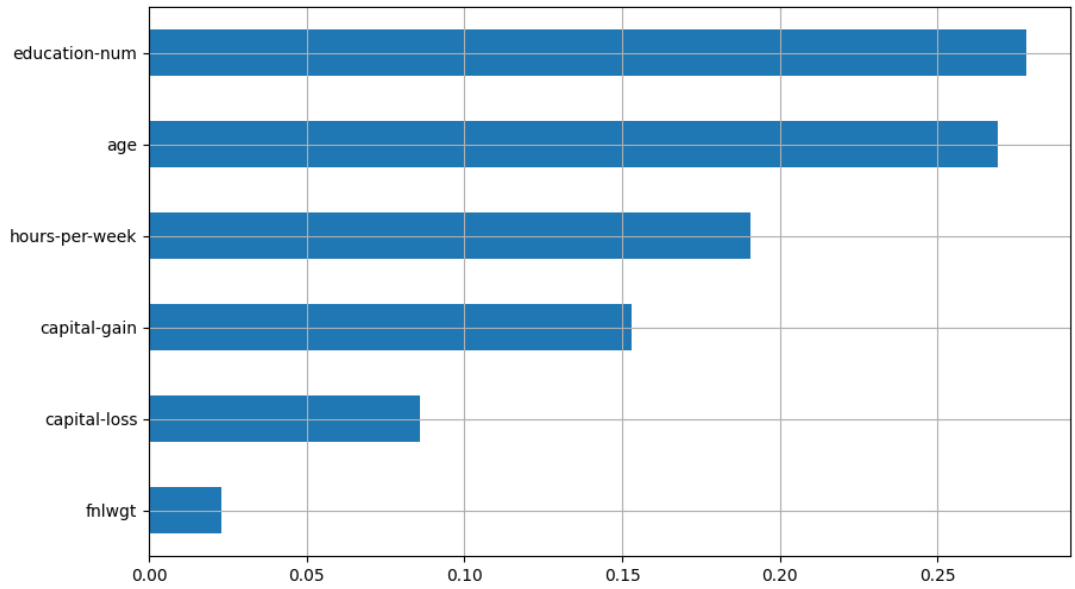

Рисунок 3.6. – Коэффициенты регрессии логистической регрессии с L1-регуляризацией

Коэффициенты регрессии для линейной модели с L2-регуляризацией (рисунок 3.7):

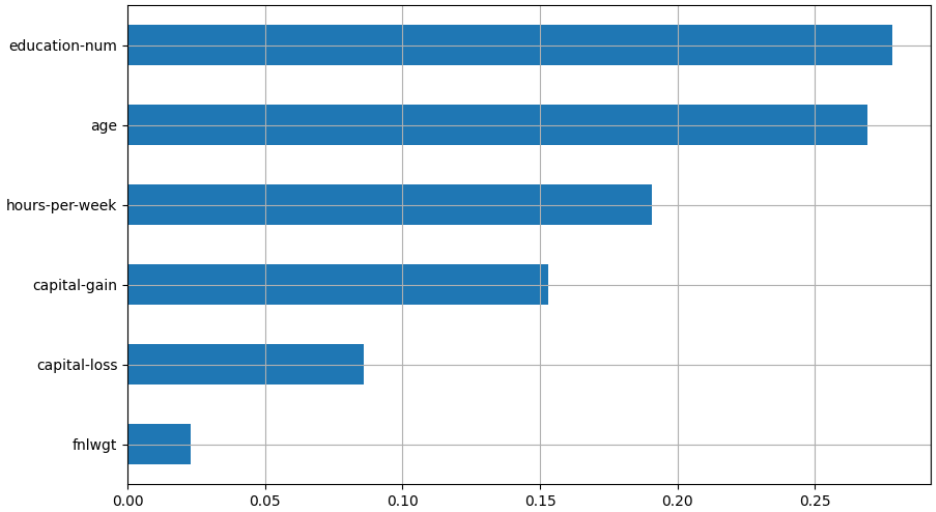

Рисунок 3.7. – Коэффициенты регрессии логистической регрессии с L2-регуляризацией

В отличие от случайного леса, регуляризация позволила сразу определить низкую важность параметра fnlwgt, что хорошо согласуется с выбором метода прямого последовательного отбора признаков. Отметим также, что модели оказались не переобучены (точности классификатора как на тестовой, так и на тренировочной выборках оказались равными 82,8%), а параметр education-num получил оценку важности, сходную с оценкой параметра age.

3.3 Внесение дополнительных шумовых признаков

Создадим 12 дополнительных шумовых признаков, элементами которых будут некоррелируемые случайные числа с нормальным, равномерным и Лапласовым распределениями. Параметры каждого из распределений берутся случайным независимым образом (рисунок 3.8).

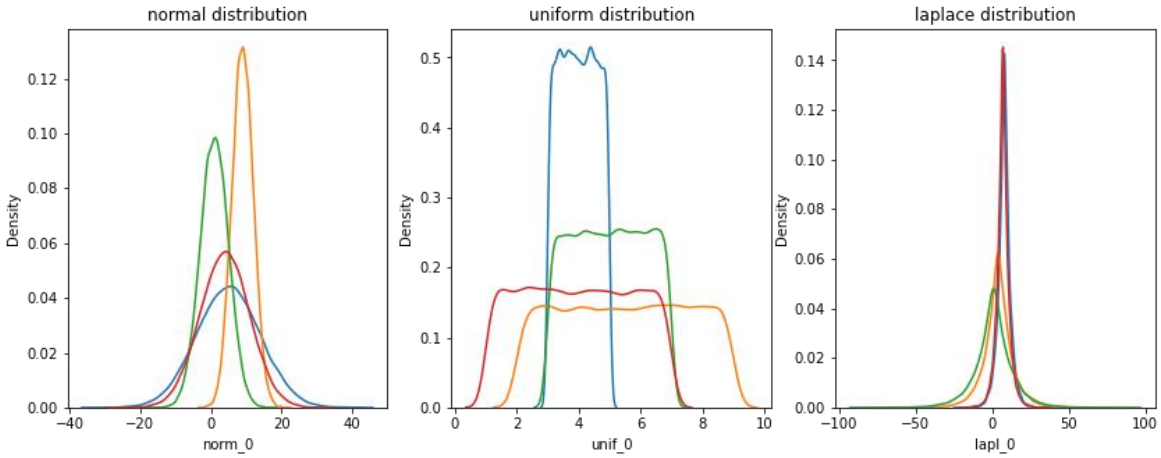

Рисунок 3.8. - Графики распределения плотности вероятностей шумовых признаков

Покажем, что шумовые признаки слабо коррелируют с целевой переменной с помощью функции взаимной информации (mutual information) (таблица 3.1):

Таблица 3.1. Взаимная информация признаков по сравнению с целевой переменной

feature_name	Mutual information
capital-gain	0.080221
age	0.065703
education-num	0.064743
hours-per-week	0.043655
capital-loss	0.033617
fnlwgt	0.033390
norm_3	0.003217
unif_3	0.002696
norm_0	0.002506
norm_2	0.002052
lapl_3	0.001201
unif_1	0.001144

Заметим, что основные признаки имеют на порядок большую оценку количества взаимной информации по сравнению с шумовыми.

Точность на кросс-валидации для случайного леса с внесенными шумовыми признаками:

train scores = [1. 1. 1. 1. 1.]

mean score = 1.00000 +/- 0.00000

test score = [0.8522425 0.85382173 0.86249657 0.84897581 0.85443027]

mean score = 0.85439 +/- 0.00447

Важность признаков для случайного леса (рисунок 3.8):

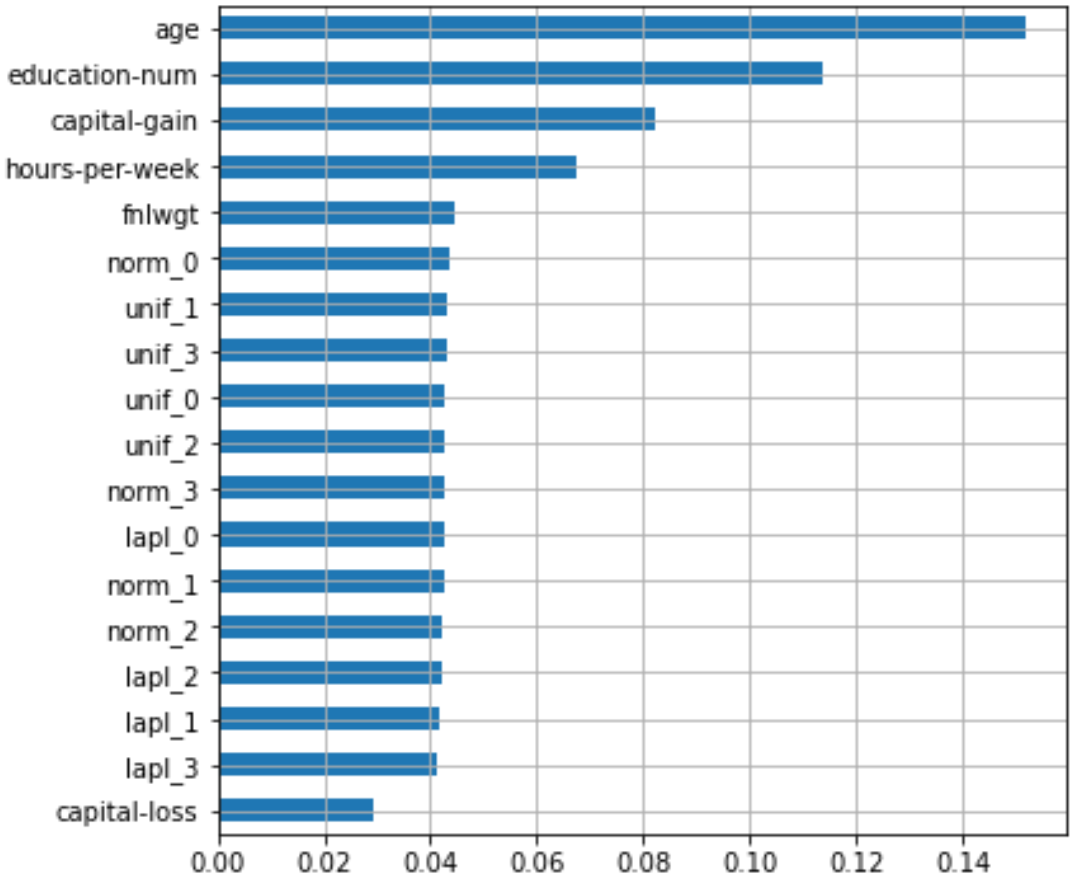

Рисунок 3.8. – Важность признаков случайного леса с шумовыми признаками

Несмотря на 12 добавленных шумовых признаков, точность модели при валидировании выросла как на каждой проверке, так и в среднем. Кроме того, все шумовые признаки имеют относительно высокую важность, сравнимую с двумя оригинальными. Очевидно, что наша модель переобучена (оценка математического ожидания точности классификатора не только достигла единицы на тренировочной выборке, но и на тестовой осталась на уровне 85%), однако в реальных задачах такие ситуации бывает очень сложно распознать исследователю, особенно когда при удалении некоторых признаков падает точность на кросс-валидации. Здесь мы видим отчетливый пример того, как влияет переобучение на важность признаков модели – значения важности

шумовых/маловажных признаков увеличивается. Поэтому часто бывает сложно подобрать постоянное пороговое значение важности признаков для исключения их из модели.

Посмотрим, как изменились оценки важности признаков для суррогата случайного леса.

Точность классификатора для суррогата случайного леса на зашумленных данных:

train scores = [1. 1. 1. 1. 1.]

mean score = 1.00000 +/- 0.00000

test score = [0.86590031 0.8665263 0.86342334 0.87133953 0.8588441]

mean score = 0.86521 +/- 0.00409

Оценка важности признаков для суррогата случайного леса (рисунок 3.9):

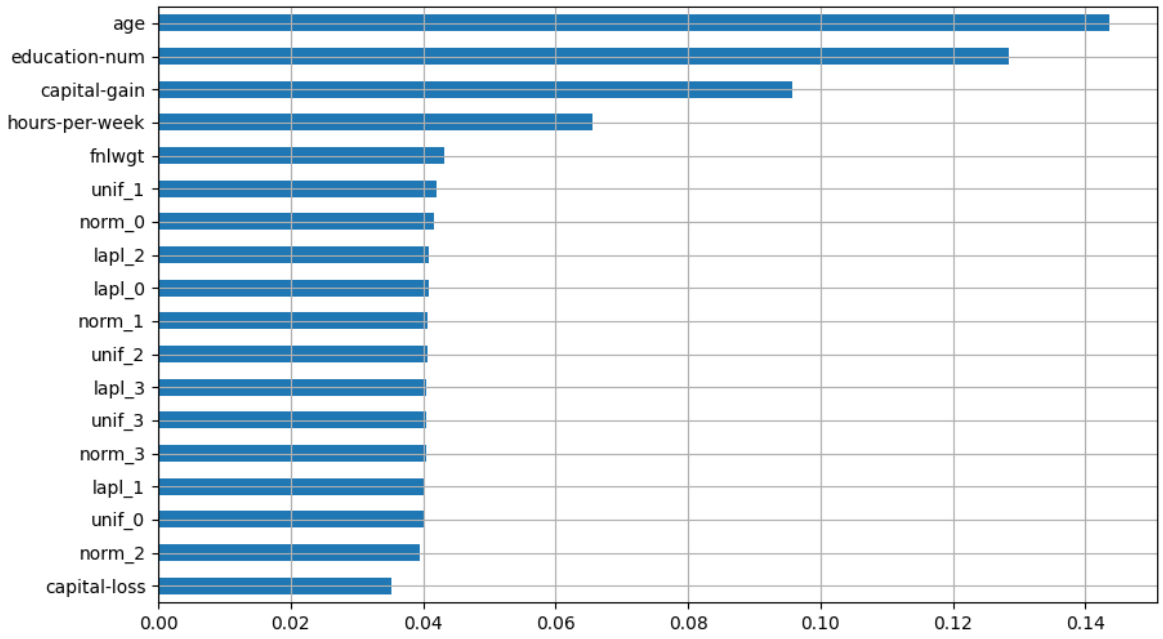

Рисунок 3.9 Важность признаков суррогата случайного леса на зашумленных данных

Характеристика education-num значительно выросла вместе с capital-gain, а вот оценка параметра age упала по сравнению с оригинальной. Отношение порядка сохранено.

Оценка важности признаков методом перестановки (рисунок 3.10):

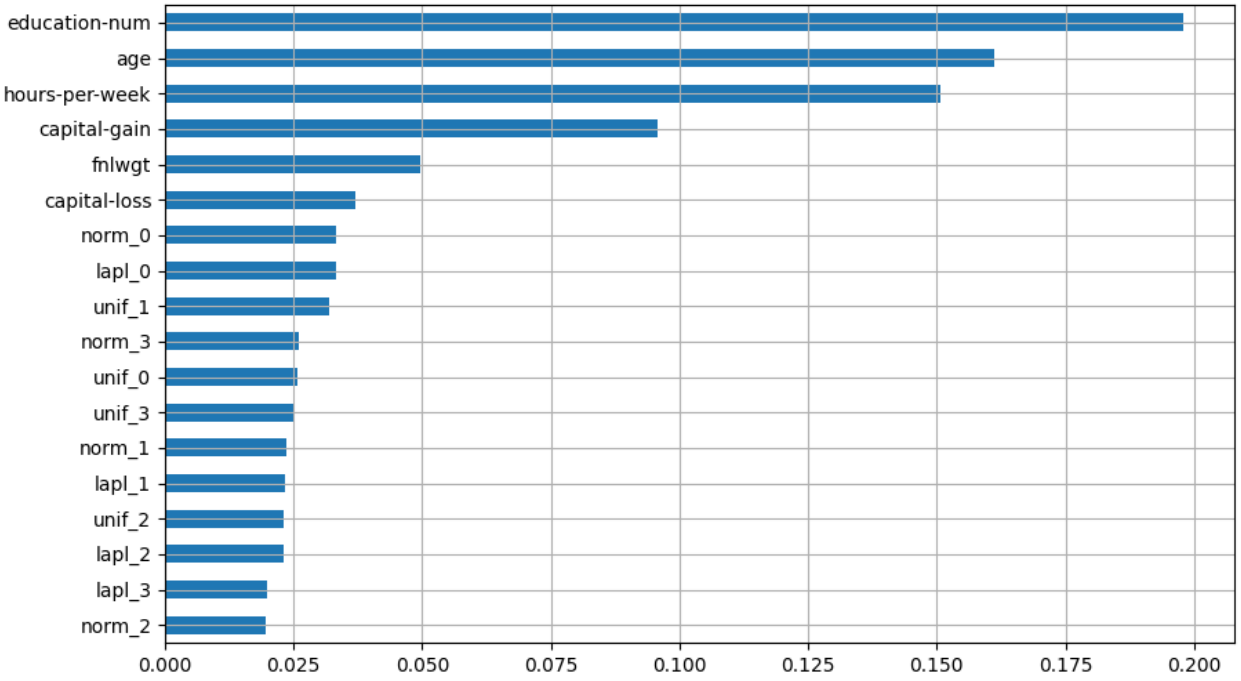

Рисунок 3.10 Оценки важности признаков методом перестановок

Шумовые признаки оценены как менее важные по сравнению с оригинальными.

Рассмотрим влияние разработанной модификации на оценки важности.

Оценка важности признаков модифицированным методом перестановки (рисунок 3.11)

Оценки шумовых признаков достигли значений, соответствующих значениям в случайном лесу, а признаки сохранили порядок, оценённый оригинальным методом.

Точность на кросс-валидации при зашумлённых данных для линейной модели с L1-регуляризацией:

train scores = [0.83019323 0.82751386 0.82642378 0.83058806 0.82619654]

mean score = 0.82818 +/- 0.00186

test score = [0.81993058 0.83005516 0.83446553 0.81763029 0.83543145]

mean score = 0.82750 +/- 0.00738

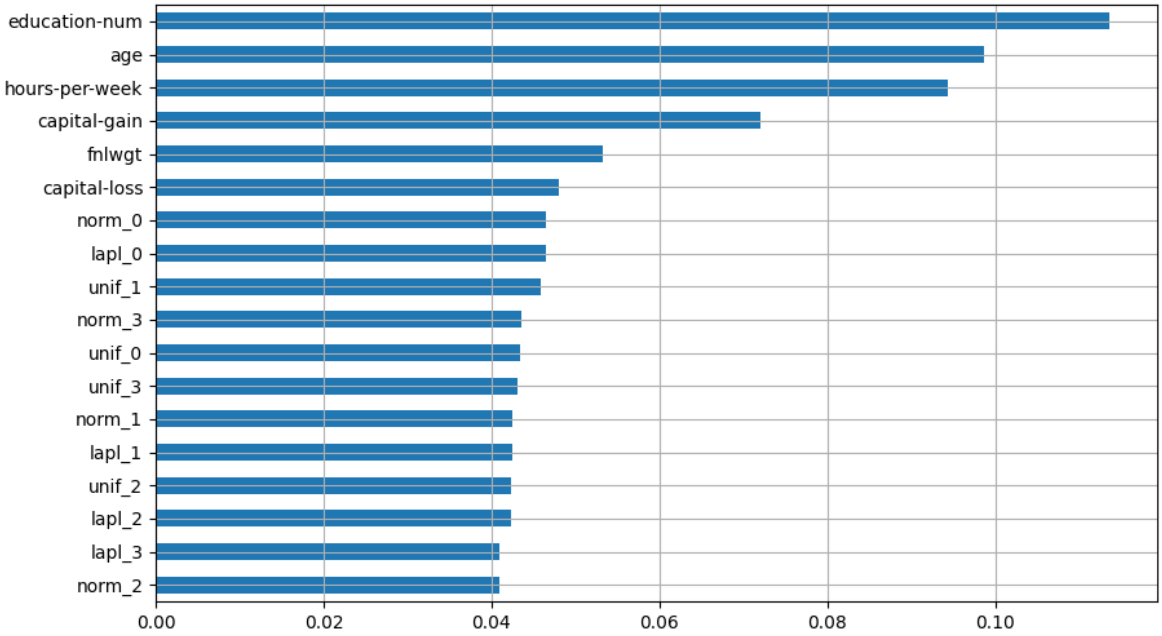

Рисунок 3.11 Оценки важности признаков модифицированным перестановочным методом.

Коэффициенты регрессии линейной модели с L1-регуляризацией (рисунок 3.12):

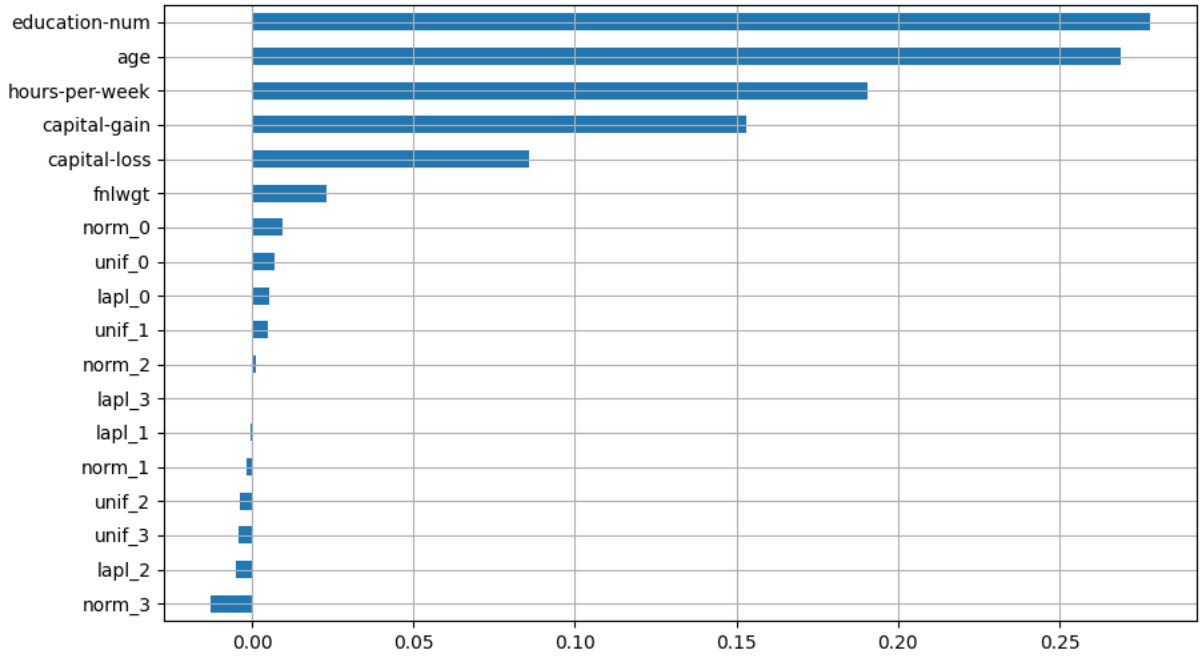

Рисунок 3.12. – Коэффициенты регрессии с шумовыми признаками при L1-регуляризации

Точность на кросс-валидации при зашумлённых данных линейной модели с L2-регуляризацией:

train scores = [0.83019479 0.827513 0.82642391 0.83058941 0.8261957]

mean score = 0.82818 +/- 0.00186

test score = [0.8199213 0.83005284 0.83446514 0.817625 0.8354299]

mean score = 0.82750 +/- 0.00739

После добавления дополнительных признаков модель не переобучилась, к тому же указанные выше признаки имеют более низкие коэффициенты, чем оригинальные. Отметим, что коэффициенты линейных моделей зависят от способа нормализации или масштабирования признаков (рисунок 3.13).

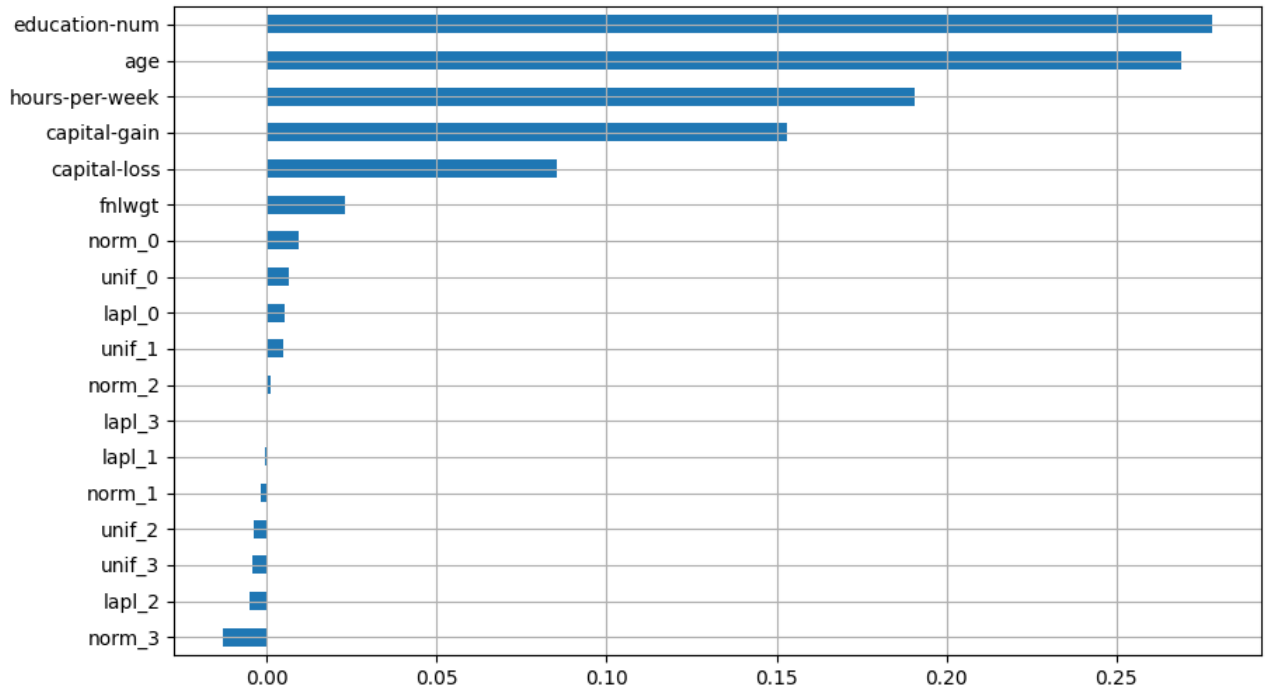

Рисунок 3.13. – Коэффициенты регрессии с шумовыми признаками при L2-регуляризации

3.4 Подбор гиперпараметров

Отбор признаков проводится статистическими методами, для чего был использован обобщённый вариант SelectKBest и SelectPercentile, который называется GenericUnivariateSelect из библиотеки sklearn[16], который принимает на вход 3 параметра – функцию оценки, режим отбора и его характеристики. В качестве функции оценки используется взаимная информация.

Сгенерированные признаки имеют низкое значение оценочной функции, поэтому в дальнейшем не используются [16].

В реальной задаче (когда количество шумовых признаков неизвестно) параметры GenericUnivariateSelect можно находить на кросс-валидации вместе с другими гиперпараметрами [17] модели. Посмотрим, как изменится точность классификаторов после подбора их гиперпараметров, а также количества признаков, используемых при обучении:

Точность на кросс-валидации для случайного леса с наилучшими гиперпараметрами модели (best_params):

train scores = [0.91956644 0.91766083 0.91975952 0.9197495 0.91516779]

mean score = 0.91838 +/- 0.00179

test score = [0.84414441 0.84622556 0.8541299 0.84239658 0.84981241]

mean score = 0.84734 +/- 0.00420

best params = {'randomforest_max_depth': 12, 'randomforest__max_features': 0.1, 'number_of_features_to_use': 5}

Значения точности классификатора на тренировочной и тестовых выборках значительно приблизились друг к другу, а лучший результат получился всего для 5 признаков (рисунок 3.14):

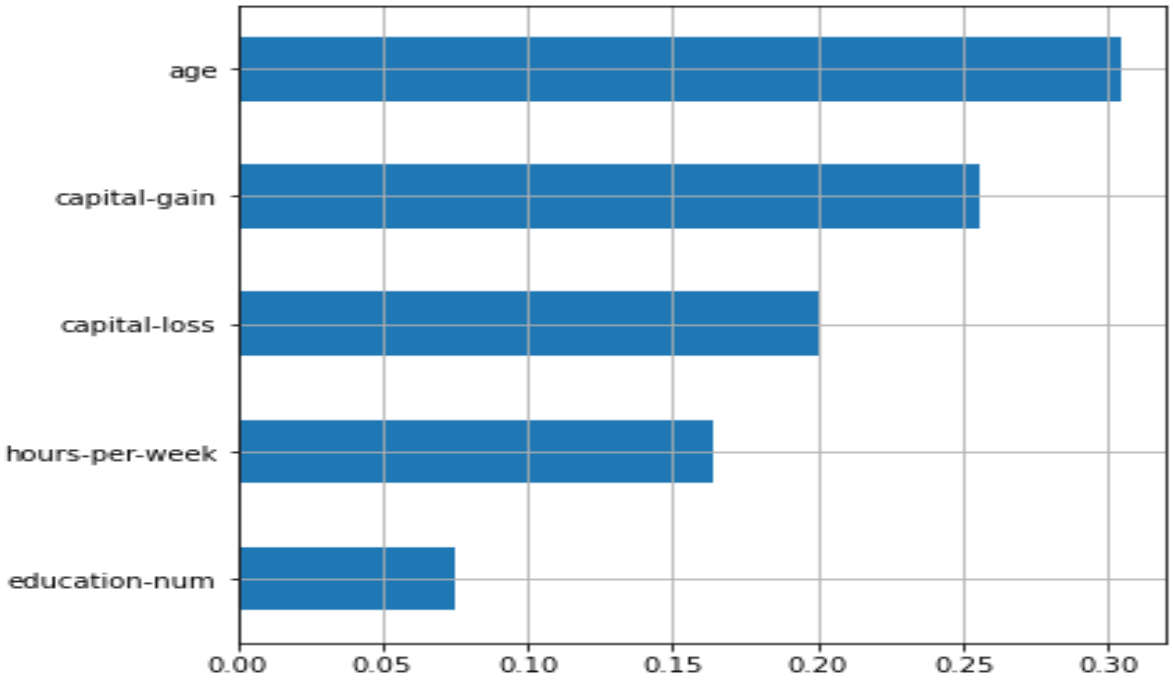

Рисунок 3.14. – Важность признаков случайного леса после фильтрации

Такой результат был показывает, что маловажными оказались шумовые признаки и признак fnlwgt, который при первоначальной оценке случайным лесом был самым значимым для модели. Однако из всех оригинальных признаков он имел наименьшее значение оценочной функции как для линейных моделей с регуляризацией, так и в GenericUnivariateSelect и в методе прямого последовательного отбора признаков. Результаты оценки важности признаков

после их отбора и настройки модели имеют более логичное значение – на целевую переменную влияют именно характеристики объекта, а не параметры выборки. Таким образом, статистический отбор признаков полезен для увеличения точности некоторых моделей и получения менее смещённой оценки при интерпретации их результатов.

Рассмотрим суррогатную модель случайного леса после подбора наилучших значений гиперпараметров модели.

Точности классификатора на кросс-валидации для суррогата случайного леса после подбора гиперпараметров (best_params):

best params = {'randomforest_max_depth': 12, 'randomforest__max_features': 0.1, 'number_of_features_to_use': 5}

train scores = [0.91185327 0.91131239 0.91662501 0.91089456 0.91029048]

mean score = 0.91220 +/- 0.00227

test score = [0.89389047 0.89511106 0.86838463 0.89805705 0.89928409]

mean score = 0.89095 +/- 0.01145

Оценки важности признаков суррогата случайного леса после подбора гиперпараметров модели (рисунок 3.15).

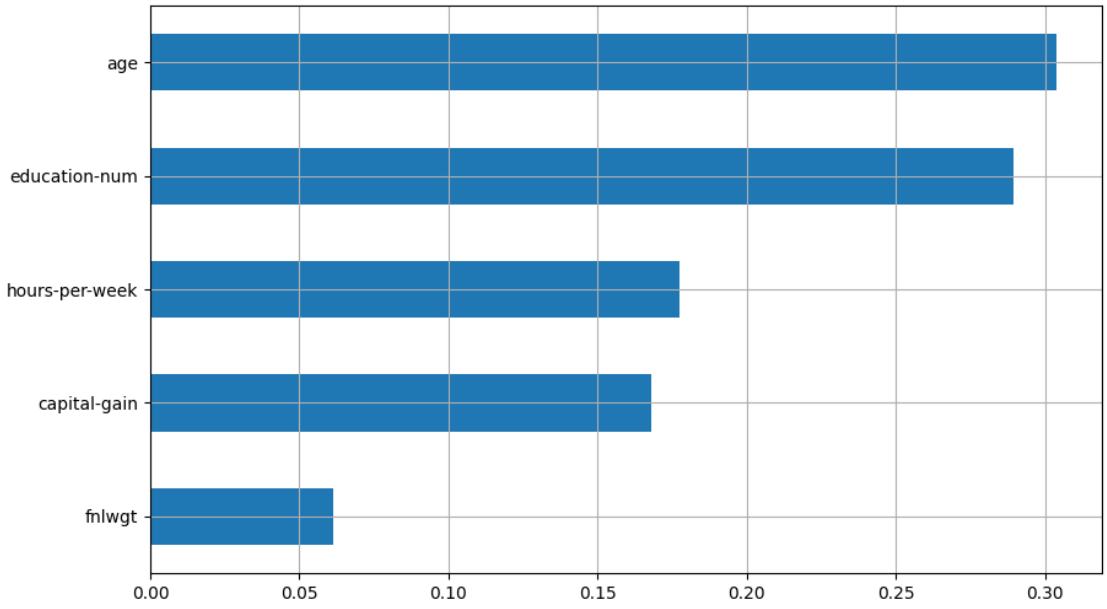

Рисунок 3.15 Оценки важности признаков суррогатной моделью случайного леса после подбора гиперпараметров модели.

Как можно заметить из таблицы 1, параметры fnlwgt и capital-loss имеют близкие показатели оценки взаимной информации, что показывает высокую вероятность подмены одного из них другим при оценках важности. То же самое и с оценками параметров age и education-num.

Оценка важности признаков методом перестановки после подбора гиперпараметров модели (рисунок 3.16):

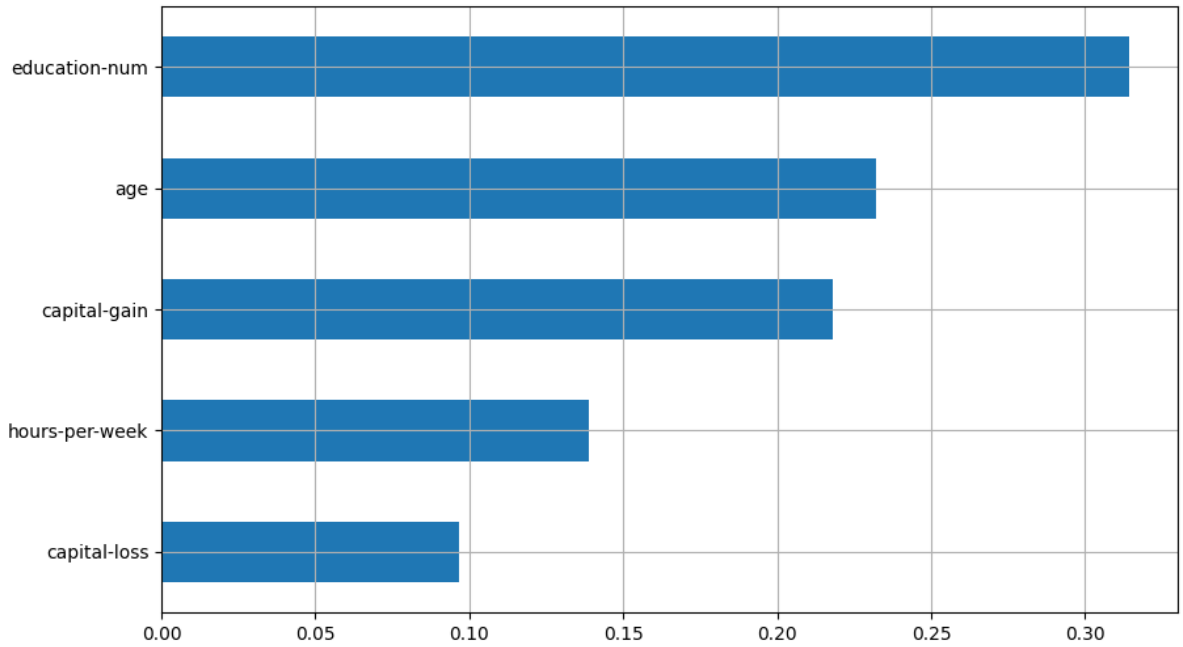

Рисунок 3.16 Оценки важности признаков методом перестановки после подбора гиперпараметров модели.

Оценка важности признаков методом перестановки после подбора гиперпараметров модели (рисунок 3.17):

Заметно, что признаки находятся в том же отношении порядка, что и при применении не модифицированного метода. Хоть оригинальный метод и расходится с оценкой важности случайного леса, такое же поведение проявляет и модифицированный метод.

Посмотрим, как изменится точность классификатора после подбора коэффициента регуляризации у логистической регрессии L1-регуляризации.

best params = {'regression__coefficient': 0.012}

train scores = [0.83006569 0.8272894 0.82627164 0.83047292 0.82605945]

mean score = 0.82803 +/- 0.00188

test score = [0.82030095 0.82989456 0.83462303 0.81777838 0.83549775]

mean score = 0.82762 +/- 0.00730

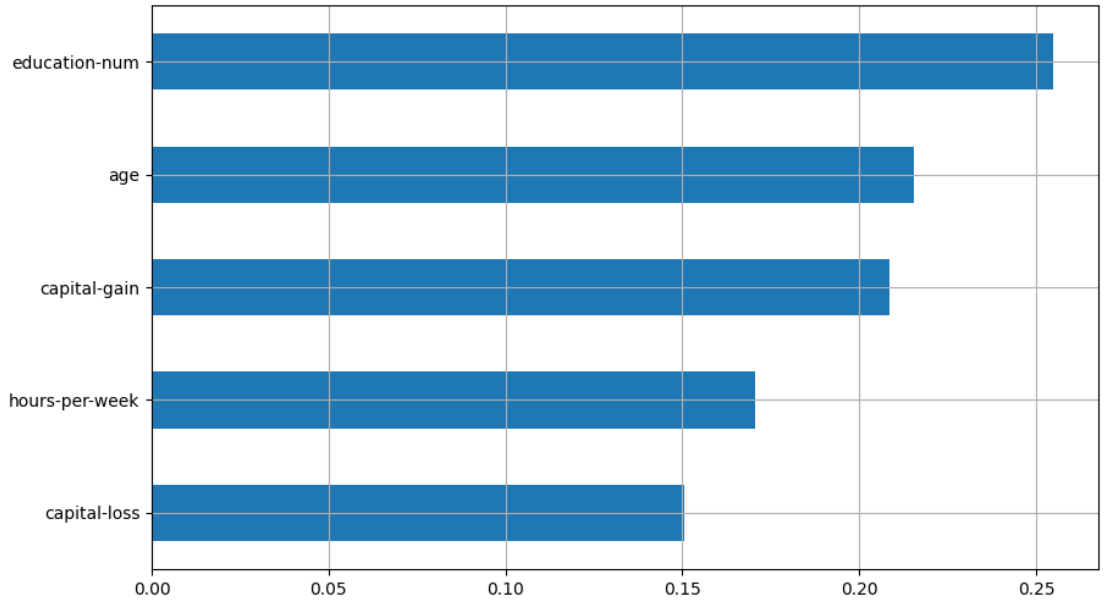

Рисунок 3.17 Оценки важности признаков модифицированным методом перестановки после подбора гиперпараметров модели.

После подбора гиперпараметров уменьшились коэффициенты оценки важности у шумовых признаков. Отметим, что сильная регуляризация (L1) может убрать большее, чем необходимо количество признаков (рисунок 3.18). Такое поведение, однако, не свойственно L2-регуляризации (рисунок 3.19).

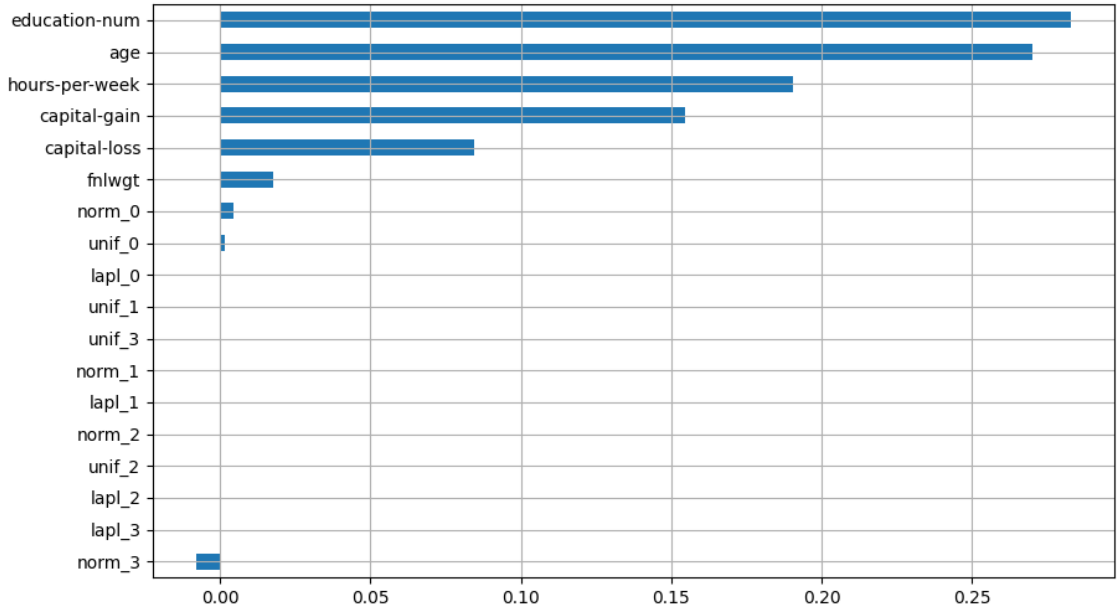

Рисунок 3.18. – Коэффициенты регрессии после подбора коэффициента L1-регуляризации

Точность классификатора после подбора коэффициента регуляризации у логистической регрессии L2-регуляризации:

train scores = [0.8302579 0.8275494 0.82648118 0.83063408 0.82626502]
mean score = 0.82824 +/- 0.00186
test score = [0.8198406 0.83012469 0.8347528 0.81769621 0.83543222]
mean score = 0.82757 +/- 0.00745
best params = {'lr__C': 0.002}

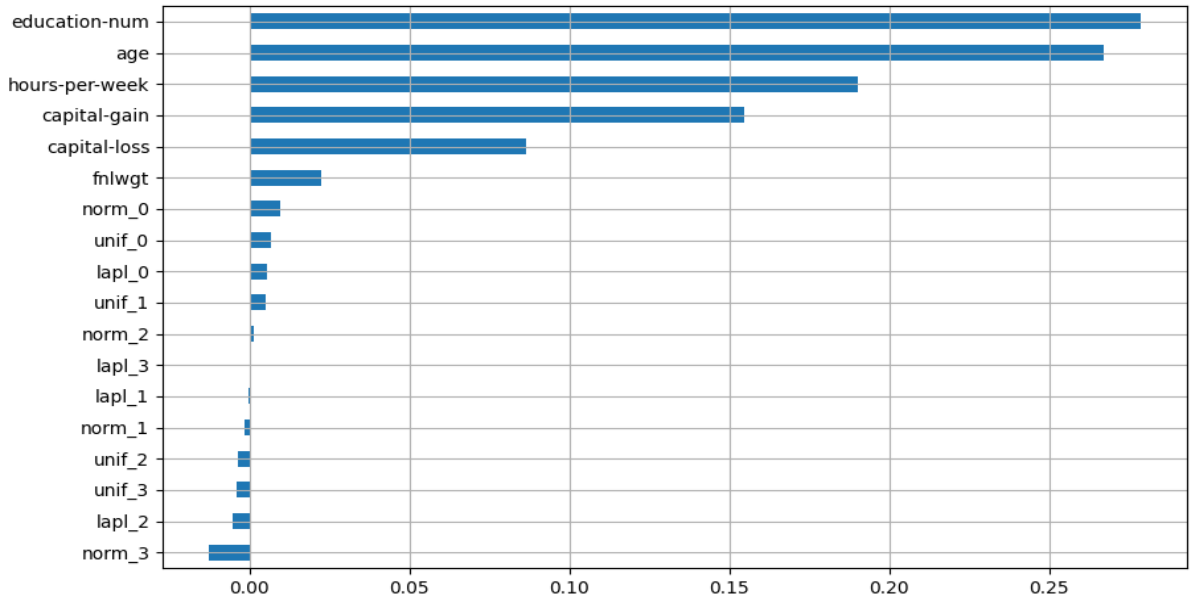

Рисунок 3.19. – Коэффициенты регрессии после подбора коэффициента L2-регуляризации

ЗАКЛЮЧЕНИЕ

В работе было рассмотрено использование методов оценки важности признаков для моделей машинного обучения с учителем.

Также в работе была рассмотрена классификация методов оценки важности признаков. Такие методы делятся на три категории: фильтры, обёрточные методы и встроенные методы.

Достоинства фильтров – стоимость вычислений линейно зависит от количества признаков, а методы интерпретации таких оценок хорошо исследованы. Недостатками фильтров является низкая степень свободы, поэтому такие методы не могут выявить более сложные зависимости в данных.

Такие методы хорошо подходят для большого количества признаков при малом количестве объектов (что встречается, например, в медицинских, или биологических исследованиях).

Также в работе были рассмотрены методы оценки важности признаков для моделей решающего дерева и случайного леса.

Достоинства решающего дерева – порождение четких правил классификации и быстрые процессы обучения и прогнозирования, а недостатки решающего дерева – чувствительность к шумам во входных данных, необходимость отсекать ветви дерева (pruning) или устанавливать минимальное число элементов в листьях дерева или максимальную глубину дерева для борьбы с переобучением.

Достоинства случайного леса - существование методов оценивания значимости отдельных признаков в модели, способность эффективно обрабатывать данные с большим числом признаков и классов. Недостатки случайного леса – увеличенная сложность интерпретации по сравнению с решающим деревом.

Также в работе была представлена модификация метода оценки важности признаков на основе перестановок. Такой метод показал более высокие значения важности признаков у тех признаков, которые были оценены более низкими значениями оригинальным методом, при этом сохраняя отношения порядка с оригинальным методом, а в некоторых случаях – еще и более высокую согласованность с иными методами оценки важности признаков, представленными в работе.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. R. Tibshirani. Regression Shrinkage and Selection via LASSO / R.Tibshirani // Journal of the Royal Statistical Society, – 1996 - V. 58, I. 1, pp. 267–288.
2. Practical part code [Electronic resource] / Ed. J.k. Reveal. – College Park M.D., 2019. – Mode of access: https://github.com/ghbdtncjctl/kkkkk. – Date of access: 24.05.2021.
3. Least Angle Regression / B. Efron, T. Hastie, I. Johnstone, R. Tibshirani. // Annals of Statistics. – 2004. – V. 32, No. 2. – pp. 407–499.
4. Ensemble methods (begging, busting, steking) [Electronic resource] / Ed. J.k. Reveal. – College Park M.D., 1996. – Mode of access: https://neurohive.io/ru/osnovy-data-science/ansamblevye-metody-begging-busting-i-steking/. – Date of access: 24.05.2021.
5. Feature Engeneering [Electronic resource] / El. J.K. Higual. – College Park M.K., 2001. – Mode of access: https://nagornyy.me/courses/data-science/feature_engineering/. – Date of access: 24.05.2021.
6. Random Forest Importance [Electronic resource] / Terence Parr, Kerem Turgutlu, Christopher Csiszar, and Jeremy Howard March 26, 2018 – Mode of access: https://explained.ai/rf-importance/. – Date of access: 24.05.2021.
7. Random Forest [Electronic resource] / Terofce Rorr, Kem Turlu, Cher Crenzar, and Jeremy Reveal. – March 26, 2018 – Mode of access: https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf. – Date of access: 24.05.2021.
8. Data Mining and Visualization [Electronic resource] / Ronny Kohavi and Barry Becker // Silicon Graphics. – Mode of access: https://archive.ics.uci.edu/ml/datasets/Adult. – Date of access: 24.05.2021.
9. The Elements of Statistical Learning / Hastie T., Tibshirani R., Friedman J. –2009.
10. Sklearn Library [Electronic resource] / Kerem Tibron, Christopher Higual, March 26, 2009 – Mode of access: ttps://mlcourse.ai/articles/topic5-part3-feature-importance/#3.-Sklearn-Random-Forest-Feature-Importance/. – Date of access: 24.05.2021.
11. Scikit-Learn Library [Electronic resource] / Kerem Tibron, Christopher Higual, March 26, 2009 – Mode of access: https://scikit-learn.org/stable/index.html. – Date of access: 24.05.2021.
12. Least Angle Regression for Stepwise regression [Electronic resource] / Vetrov Semen, Keron Hireal. – May 2, 2001. – Mode of access: http://www.machinelearning.ru/wiki/images/7/7e/VetrovSem11_LARS.pdf. – Date of access: 24.05.2021

13. Scikit-Learn Library – StratifiedKFold class [Electronic resource] / Kerem Tibron, Christopher Higual, March 26, 2009 – Mode of access: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html – Date of access: 24.05.2021.
14. Scikit-Learn Library – PowerTransformer class [Electronic resource] / Kerem Tibron, Christopher Higual, March 26, 2009 – Mode of access: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PowerTransformer.html – Date of access: 24.05.2021.
15. Yeo-Johnson Power Transformations / Sanford Weisberg Department of Applied Statistics, University of Minnesota, October 26, 2001 – Mode of access: https://www.stat.umn.edu/arc/yjpower.pdf – Date of access: 20.12.2021.
16. Scikit-Learn Library – GenericUnivariateSelect [Electronic resource] / Kerem Tibron, Christopher Higual, March 26, 2009 – Mode of access: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.GenericUnivariateSelect.html – Date of access: 24.05.2021.
17. Scikit-Learn Library – GridSearchCV class [Electronic resource] / Kerem Tibron, Christopher Higual, March 26, 2009 – Mode of access: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html – Date of access: 24.05.2021.
18. Scikit-Learn Library – Permutation feature importance [Electronic resource] / Kerem Tibron, Christopher Higual, March 26, 2009 – Mode of access: https://scikit-learn.org/stable/modules/permutation_importance.html – Date of access: 24.05.2021.
19. Surrogate-based modeling and optimization. / Koziel S., Leifsson L. – New York: Springer, 2013 – Date of access: 24.05.2021.
20. Stacked generalization / Wolpert, David H. / Neural networks 5.2 (1992): 241–259 – Date of access: 24.05.2021
21. Stacked regressions / Breiman, Leo. / Machine learning 24.1 (1996): 49-64– Date of access: 24.05.2021.
22. Bagging predictors / Breiman, Leo. / Machine learning 24.2 (1996): 123-140– Date of access: 24.05.2021.
23. Random Forests / Breiman, Leo. / Machine Learning, 45(1), 5-32, 2001– Date of access: 24.05.2021
24. A Short Introduction to Boosting / Freund, Yoav, Robert Schapire, and N. Abe. /Journal-Japanese Society for Artificial Intelligence 14.771-780 (1999): 1612 – Date of access: 24.05.2021
25. Stochastic gradient boosting / Friedman, Jerome H. / Computational Statistics and Data Analysis, 2002 –Date of access: 24.05.2021

26. Об алгебраическом подходе к решению задач распознавания или классификации / Журавлёв Ю. И. / Проблемы кибернетики. 1978 – Дата доступа: 24.05.2021
27. Limited bagging, boosting and the random subspace method for linear classifiers / Skurichina M., Duin R. P. W. / Pattern Analysis & Applications. 2002. Pp. 121 – 135. – Date of access: 24.05.2021
28. Feature-weighted linear stacking / Joseph Sill, Gabor Takacs, Lester Mackey, David Lin / arXiv preprint arXiv:0911.0460(2009). v Date of access: 24.05.2021
29. Troika–An improved stacking schema for classification tasks / Menahem, Eitan, Lior Rokach, and Yuval Elovici. / Information Sciences 179.24 (2009): 4097-4122. – Date of access: 24.05.2021
30. How to Make Stacking Better and Faster While Also Taking Care of an Unknown Weakness / Seewald, A. / Nineteenth International Conference on Machine Learning (pp. 554-561) (2002) – Date of access: 25.05.2021
31. Scalable stacking and learning for building deep architectures / Deng, Li, Dong Yu, and John Platt. / Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEEInternational Conference on. IEEE, 2012. Date of access: 26.05.2021
32. Hierarchical mixtures of experts and the EM algorithm / Jordan, Michael I., and Robert A. Jacobs. / Neural computation 6.2 (1994): 181-214. – Date of access: 27.05.2021
33. Scikit-Learn Library – Permutation feature importance [Electronic resource] / Kerem Tibron, Christopher Higual, March 26, 2009 – Mode of access: https://scikit-learn.org/stable/modules/generated/sklearn.inspection. permutation_importance.html – Date of access: 24.05.2021.
34. Методология построения суррогатных моделей для аппроксимации пространственно неоднородных функций / Бурнаев Е. В., Приходько П. В. / Труды МФТИ. 2013. №4 (20). Mode of access: https://cyberleninka.ru/article/n/metodologiya-postroeniya-surrogatnyh-modeley-dlya-approksimatsii-prostranstvenno-neodnorodnyh-funktsiy – Дата доступа: 24.05.2022).
35. Scikit-Learn Library – Feature selection [Electronic resource] / Kerem Tibron, Christopher Higual, March 26,2009–Mode of access: https://scikit-learn.org/stable/modules/feature_selection.html – Date of access: 24.05.2021.
36. Кобзарь А. И. Прикладная математическая статистика. — М.: Физматлит, 2006. — 626-628 с.
37. Лагутин М. Б. Наглядная математическая статистика. В двух томах. — М.: П-центр, 2003. — 343-345 с.
38. Лекция "Решающие деревья"/ Соколов Е.А. / Mode of access: https://www.hse.ru/mirror/pubs/share/215285956 — Дата доступа: 24.05.2022.