Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра прикладной математики и компьютерной безопасности

УТВЕРЖДАЮ
Заведующий кафедрой
_____ А.А. Кытманов
« _____» _____ 2019 г.

БАКАЛАВРСКАЯ РАБОТА

01.03.04 – Прикладная математика

Применение регрессионного анализа для расчета прогнозных значений популярности видео

Руководитель _____ доцент каф. ПМКБ, к.ф.-м.н Т.А. Кустицкая

Выпускник _____ А.И. Перебигайло

Красноярск 2019

РЕФЕРАТ

Выпускная квалификационная работа по теме «Применение регрессионного анализа для расчета прогнозных значений популярности видео» содержит 36 страниц текстового документа, 1 приложение, 18 использованных источников.

РЕГРЕССИОННЫЙ АНАЛИЗ, РЕШАЮЩИЕ ДЕРЕВЬЯ, РЕГРЕССИЯ, ВЫБРОС, ОБРАБОТКА ДАННЫХ.

Цель работы состоит в построении нескольких регрессионных моделей (линейных и с помощью метода случайного леса) для прогноза популярности видео на основе набора данных с медиахостингового сайта YouTube, их анализе и выборе наиболее точной модели.

Задачи работы следующие: изучение основной теории на тему регрессионного анализа, обработка данных, изучение и реализация основных моделей регрессионного анализа, устранение выбросов в наборе данных, оценка адекватности модели.

Выпускная квалификационная работа состоит из двух глав. В первой главе рассмотрена минимальная теория, требуемая для построения моделей регрессии и последующего их анализа. Вторая глава содержит практическую часть работы, в которой производятся вычисления коэффициентов регрессионного уравнения, выявление и устранение выбросов, оценка качества моделей.

В результате проведения исследования построены модели для прогнозирования популярности видео, и вычислены ошибки прогноза, сделан вывод о том, какая из моделей является наиболее точной.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Обработка статистических данных уже давно применяется в самых разнообразных видах человеческой деятельности. В качестве примера можно привести любую область знаний, которая имеет дело с обработкой и анализом огромных массивов информации. Всесторонний и глубокий анализ этой информации, так называемых статистических данных, предполагает использование различных специальных методов, важное место среди которых занимает регрессионный анализ данных, позволяющий решать задачу прогнозирования различных показателей.

Задачи прогнозирования решаются, в частности:

- в интернет-магазинах для выявления потребностей и вкусов покупателя;
- в медицине для выяснения болезней пациента;
- в бизнесе для построения стратегий и т.д.

Исследование прогноза популярности видео является важным и актуальным для человека доход, которого состоит из продажи рекламы. На видеохостинговой платформе YouTube при загрузке видео есть возможность вставить в него рекламу, которая будет приносить в дальнейшем доход, зависимый от количества просмотров данного видео, в свою очередь от количества положительных оценок будет зависеть, как часто видео будет отображаться видео в рекомендациях. Для решения задачи прогноза количества положительных оценок лучше всего подходят методы регрессионного анализа, которые изучаются и используются в данном исследовании.

Цель исследования: изучить общую постановку задачи регрессии, модели множественной линейной регрессии, решающих деревьев и случайного леса, кросс-валидации и LAD-регрессии. Применить эти методы к решению задачи предсказания популярности видео.

Основные задачи исследования:

1. Изучить научную литературу в предметной области на тему регрессионного анализа.
2. Выполнить первичную обработку данных.
3. Устранить выбросы в исследуемых данных.
4. Реализовать алгоритмы множественной линейной регрессии, случайного леса и LAD-регрессии для исследуемых данных на языке R.
5. Провести уточнение коэффициентов линейной регрессии методом кросс-валидации.
6. Оценить адекватность моделей.
7. Сравнить полученные модели с точки зрения их точности.

1. Теоретические основы регрессионного анализа

Регрессия — зависимость математического ожидания случайной величины от одной или нескольких других случайных величин (свободных переменных), то есть

$$E(y|x) = f(x), \tag{1}$$

где $y -$ зависимая переменная, $x -$ объясняющая переменная. Регрессионным анализом называется поиск такой функции f, которая наилучшим образом приближает данные. Критерием качества приближения (целевой функцией) обычно является среднеквадратичная ошибка:

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}{n}. \tag{2}$$

Регрессия может быть представлена в виде суммы неслучайной и случайной составляющих.

$$y = f(x) + v, \tag{3}$$

где f - функция регрессионной зависимости, а v - аддитивная случайная величина с нулевым матожиданием [12].

1.1 Линейная регрессия

В линейной регрессии по определению объясняемая переменная зависит от другой или нескольких других независимых переменных линейно.

$$y = \omega_0 + \sum_{i=1}^{n} \omega_i x_i. \tag{4}$$

Зададим модель следующим образом:

$$y = X\omega + \varepsilon, \tag{5}$$

где

- $y \in R^n$ – вектор с объясняемой (или целевой) переменной;

- $\omega = (1,\ \omega_1, \dots ,\ \omega_m)$ - вектор параметров модели (в машинном обучении эти параметры часто называют весами);
- X - матрица наблюдений и признаков размерности n строк на $m\ +\ 1$ столбцов (включая фиктивную единичную колонку слева) с полным рангом по столбцам: $rank(X) = m + 1$.
- ε - случайная переменная, соответствующая случайной непрогнозируемой ошибке модели.

Для отдельных наблюдений формула (5) выглядит следующим образом:

$$y_i = \sum_{j=0}^{m} \omega_j X_{ij}\ +\ \varepsilon_i \tag{6}$$

Также на модель накладываются следующие ограничения:

- математическое ожидание случайных ошибок равно нулю: $\forall i: E[\varepsilon_i] = 0$;
- дисперсия случайных ошибок одинакова и конечна: $Var(\varepsilon_i) =\ \sigma^2 < \infty$;
- случайные ошибки не коррелированы: $\forall i\ \neq j: Cov(\varepsilon_i\ , \varepsilon_j) = 0$.

Задача регрессии заключается в поиске оценок параметров модели ω_i, которые дают минимальные отклонения между фактическими значениями зависимой переменной и восстановленными [2].

1.1.1 Метод наименьших квадратов

Данный метод заключается в минимизации суммы квадратов отклонений (в регрессионном анализе они чаще называются остатками регрессии), относительно параметра ω.

Поскольку наша модель линейной регрессии задана формулой (5), вектор оценок объясняемой переменной $\hat{y}$ и вектор остатков регрессии e будут равны:

$$\hat{y} = X\omega, \qquad e = \vec{y} -\ \hat{y} = \vec{y} - X\omega. \tag{7}$$

Соответственно, сумма квадратов остатков регрессии будет равна:

$$RSS = e^T e = (\vec{y} - X\omega)^T (\vec{y} - X\omega). \tag{8}$$

Дифференцируя эту функцию по вектору параметров ω и приравняв производные к нулю, получим систему уравнений (в матричной форме):

$$(X^T X)\omega = X^T \vec{y}. \tag{9}$$

Отсюда найдем наш вектор ω:

$$\omega = (X^T X)^{-1} X^T \vec{y}. \tag{10}$$

При обучении весь набор данных делится на обучающую и проверочную выборки. Как следует из названия, на проверочной выборке производится проверка точности модели [12].

1.1.2 LAD-регрессия

Данная модель основана на методе наименьших модулей (least absolute deviation). Она используется для оценки неизвестных величин по результатам измерений, содержащих случайные ошибки, а также для приближенного представления заданной функции более простыми (аппроксимации). Похожа на линейную регрессию, но использует абсолютные величины вместо квадратов – в итоге, вместо оценивания условного математического ожидания (МНК), оценивается условная медиана.

Минимизируется абсолютное расстояние между фактическим значением зависимой переменной и предсказанным, выраженное следующим функционалом:

$$d[Y, f(X)] = \sum_{i=1}^{n} |y_i - f(x_i)|. \tag{11}$$

Если для метода наименьших квадратов требуется нормальное распределение остатков, то при использовании этой модели остатки должны иметь распределение Лапласа [15].

1.1.3 Разный масштаб признаков

Другой важной проблемой многомерной линейной регрессии является разнородность признаков. Если масштабы измерений признаков существенно (на несколько порядков) различаются, то появляется опасность, что будут учитываться только «крупномасштабные» признаки. Чтобы этого избежать, делается стандартизация (нормировка) матрицы:

$$x_{ij} = \frac{x_{ij} - x_{j_{\text{ср}}}}{\sigma_j}, \ \ j = 1 \dots m, \qquad i = 1 \dots n. \tag{12}$$

где, $x_{j_{\text{ср}}} = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$ – выборочное среднее, а $\sigma_j^2 = \frac{1}{n}\sum_{i=1}^{n}\left(x_{ij} - \overline{x_j}\right)^2$ - выборочная дисперсия. Причем после выполнения нормировки, ее следует применять ко всем объектам, посылаемым в качестве признака для построения предсказания [1].

1.1.4 Перекресная проверка (кросс-валидация)

В случае недостаточного количества данных при делении исходной выборки на обучающую и проверочную может возникнуть проблема, когда модель плохо обучается, что влечет за собой большие ошибки прогнозирования.

При использовании кросс-валидации фиксируется некоторое множество разбиений исходной выборки на две подвыборки: обучающую и контрольную. Для каждого разбиения выполняется настройка алгоритма по обучающей подвыборке, затем оценивается его средняя ошибка на объектах контрольной подвыборки. Оценкой кросс-валидации называется средняя по всем разбиениям величина ошибки на контрольных подвыборках.

Существует несколько способов реализации кросс-валидации, отличающиеся способом разбиения выборки, рассмотрим способ, использованный в исследовании. Изначальная выборка X разбивается на k примерно одинаковых по длине частей, таким образом, что: $X = X_1 \cup X_2 \cup \dots \cup X_k$. После этого модель начинает обучаться на $k - 1$ блоке, а не вошедший в обучающую выборку блок, становится проверочным, и так k раз. Обычно принято делить выборку на 10 частей.

Если выборка независима, то средняя ошибка кросс-валидации даёт несмещённую оценку вероятности ошибки. Это выгодно отличает её от средней ошибки на обучающей выборке, которая может оказаться смещённой (оптимистически заниженной) оценкой вероятности ошибки, что связано с явлением переобучения [13].

1.2 Метод случайного леса

1.2.1 Решающие деревья

Решающее дерево (Decision tree) — решение задачи обучения с учителем, основанный на том, как решает задачи прогнозирования человек. В общем случае — это k −ичное дерево с решающими правилами в нелистовых вершинах (узлах) и некотором заключении о целевой функции в листовых вершинах (прогнозом).

Схеме принятия решений соответствует связный ориентированный ациклический граф – ориентированное дерево. Дерево включает в себя корневую вершину, инцидентную только выходящим рёбрами, внутренние вершины (узлы), инцидентную одному входящему ребру и нескольким выходящим, и листья – концевые вершины, инцидентные только одному входящему ребру.

Каждый узел и корень содержат решающее правило.

Решающее правило — некоторая функция от объекта, позволяющее определить, в какую из дочерних вершин нужно поместить рассматриваемый объект. В листовых вершинах могут находиться разные объекты: класс, который нужно присвоить попавшему туда объекту (в задаче классификации), вероятности классов (в задаче классификации), непосредственно значение целевой функции (задача регрессии). Как вариант можно использовать следующее решающее правило: $\beta(x, j, t) = [x_j < t]$, где t некоторая константа [12].

Решающие правила разбивают, с помощью рекурсивного бинарного разбиения, пространство на некоторое количество непересекающихся подмножеств $\{J_1, J_2, J_3, \dots, J_n\}$, и в каждом подмножестве J_j выдает константный прогноз ω_j. Значит, соответствующий алгоритм можно записать аналитически:

$$f(x) = \sum_{j=1}^{n} \omega_i \, I(x \in J_j). \tag{13}$$

Алгоритм построения дерева:

1. Проверить критерий останова алгоритма. Если он выполняется, выбрать для узла выдаваемый прогноз, что можно сделать несколькими способами (Критерий останова может быть разнообразный: ограничение максимальной глубины дерева, ограничение минимального числа объектов в листе, ограничение максимального количества листьев в дереве и т.д.).
2. Иначе требуется разбить множество на несколько не пересекающихся. В общем случае в вершине t задаётся решающее правило $Q_t(x)$, принимающее некоторый диапазон значений. Этот диапазон разбивается на R_t непересекающихся множеств объектов, $S_1, S_2, \dots, S_{R_t}$, где R_t — количество потомков у вершины, а каждое S_i — это множество объектов, попавших в i-го потомка.
3. Множество в узле разбивается согласно выбранному правилу, для каждого узла алгоритм запускается рекурсивно [10].

1.2.2 Алгоритм случайного леса

В исследовании будет использоваться алгоритм являющийся надстройкой над решающими деревьями, а именно алгоритм случайного леса, один из лучших для прогнозирования, но имеющий следующий недостаток: увидеть явный вид обученной модели невозможно.

Основная идея заключается в использовании большого ансамбля решающих деревьев, каждое из которых само по себе даёт очень невысокое качество, однако при использовании большого количества деревьев качество повышается [14].

Существует много версий алгоритма, но выделяют две основных, это: для количественных переменных y_i, и для качественных переменных y_i. Рассмотрим алгоритм построения леса:

1. Выбирается подвыборка из обучающей выборки размера l (м.б. с возвращением) – по ней строится дерево (для каждого дерева — своя подвыборка);
2. Для построения каждого расщепления в дереве просматриваем k случайных признаков из всех (для каждого нового расщепления — свои случайные признаки);
3. Выбираем наилучший признак и расщепление по нему (по заранее заданному критерию, например по минимизирование RSS). Дерево строится, как правило, до исчерпания выборки (пока в листьях не останутся представители только одного класса);
4. Вычисляем итоговое предсказание $a(x) = \frac{1}{N}\sum_{i=1}^{N} b_i(x)$, где $b_i(x) -$ предсказание в $i -$ ом дереве.

Рекомендуется в задачах регрессии брать количество случайных признаков $k = \frac{3}{n}$, где n – количество всех признаков [10].

1.3 Удаление выбросов из набора данных

1.3.1 Точки с избыточным влиянием на модель

Говоря о необычных наблюдениях в контексте регрессионного анализа, можно выделить следующие три ситуации:

- Наблюдение представлено необычным сочетанием значений предикторов;
- наблюдение не согласуется с рассматриваемой моделью, т.е. является выбросом;
- наблюдение оказывает существенное влияние на оценки параметров модели; другими словами, удаление такого влиятельного

наблюдения из выборки приведет к значительному изменению предсказываемых моделью значений.

Необычные наблюдения могут оказывать существенное влияние на качество модели (как с точки зрения статистической значимости ее параметров, так и с точки зрения ее предсказательной силы), в связи, с чем выявление таких наблюдений является важной частью диагностики регрессионных моделей. Имеется возможность выразить потенциал воздействия количественно [16] . Распространенным подходом является расчет диагональных элементов, т.е. матрицы влияния, которые обозначаются как h_{ii}.

Данная матрица берется из линейной модели в общем виде:

$$y = X\omega + \varepsilon, \tag{14}$$

где, X – матрица модели, ω – вектор подлежащих оценке регрессионных коэффициентов, а ε – вектор остатков с нулевым математическим ожиданием.

Вектор с оценками регрессионных коэффициентов $\widehat{\omega}$ получают следующим образом:

$$\widehat{\omega} = (X^T X)^{-1} X^T y, \tag{15}$$

откуда предсказываемые значения можно записать следующим образом:

$$\hat{y} = X\widehat{\omega} = X(X^T X)^{-1} X^T y \tag{16}$$

или, в более простой форме, как:

$$\hat{y} = Hy \tag{17}$$

Матрица H (размером $n \times n$) как раз и является матрицей влияния, поскольку она позволяет выполнить линейное преобразование наблюдаемых значений y так, чтобы получить предсказываемые значения.

Оказывается, что сумма диагональных элементов матрицы проекции равна числу коэффициентов регрессионного уравнения, включая свободный член.

Соответственно, среднее значение h_{ii} можно рассчитать как p/n, где p – числу коэффициентов регрессионного уравнения, n – количество наблюдений. Отсюда вытекает эмпирическое правило, позволяющее судить о том, оказывает ли некоторое наблюдение существенное влияние на параметры модели - значения $h_{ii} > 2p/n$ являются достаточно большими, чтобы считать соответствующие наблюдения стоящими внимания.

1.3.2 Выбросы

Выброс - это наблюдение с большим остатком, возникающим из-за того, что соответствующее выборочное значение зависимой переменной y_i значительно отличается от предсказанного значения.

На практике работать с изначальными значениями выбросов, оказывается проблематично, поэтому прибегают к различным стандартизациям. В R для диагностики линейных моделей, чьи параметры оцениваются по методу наименьших квадратов, используются следующие два типа остатков:

Standardized residuals - стандартизованные остатки:

$$r_i = \frac{\varepsilon_i}{S_\varepsilon \sqrt{1 - h_{ii}}}, \tag{18}$$

где ε_i - остаток i - го наблюдения, S_ε - стандартное отклонение всех остатков модели, а h_{ii} - показатель потенциала воздействия i-го наблюдения на коэффициенты модели. Такие остатки имеют приближенно стандартное нормальное распределение [16].

Одним из важных недостатков стандартизованных остатков является тот факт, любое значение r_i и S_ε не являются независимыми, затрудняя формальную проверку статистической гипотезы о том, что некоторое i-е наблюдение не является выбросом.

Для устранения указанного недостатка используют Studentized residuals - стьюдентизированные остатки:

$$t_i = \frac{\varepsilon_i}{S_{\varepsilon_{(-t)}} \sqrt{1 - h_{ii}}} \tag{19}$$

где, $S_{\varepsilon_{(-t)}}$- стандартное отклонение, которое рассчитывается по остаткам модели, подогнанной после исключения из данных i-го наблюдения. Стьюдентизированные остатки имеют распределение Стьюдента с $n - p - 1$ степенями свободы. Соответственно, мы можем использовать квантили этого распределения для проверки того, насколько статистически значимо определенное наблюдение является выбросом. Если вычисленное значение $t_i \geq t_{\text{кр}}$, где $t_{\text{кр}}$ – это квантиль распределения Стьюдента с $n - p - 1$ степенью свободы и уровнем значимости $\alpha = 0.05$, то данное наблюдение рассматривается как выброс и подлежит удалению из выборки.

1.4 Оценка качества модели регрессии

1.4.1 Коэффициент детерминации

Данная оценка показывает, какая доля изменения исследуемого признака учтена в модели. Коэффициент детерминации R^2 может принимать значения от 0 до 1. Чем ближе коэффициент детерминации R^2 к единице, тем лучше качество модели.

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}{\sum_{i=1}^{n}\left(y_i - y_{cp}\right)^2} \quad (20)$$

где

$RSS = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n}(y_i - \widehat{y_i})^2$ – сумма квадратов регрессионных остатков,

$TSS = \sum_{i=1}^{n}\left(y_i - y_{cp}\right)^2$ – общая дисперсия,

$y_i, \widehat{y_i}$ – соответственно, фактические и расчетные значения объясняемой переменной,

$y_{\text{ср}} = \frac{1}{n}\sum_{i=1}^{n} y_i$ – выборочное среднее.

В случае линейной регрессии, $TSS = RSS + ESS$ является константой, где $ESS = \sum_{i=1}^{n}\left(\widehat{y_i} - y_{\text{ср}}\right)^2$ – объясненная сумма квадратов, отсюда получаем формулу, записанную в ином виде, которая выражает смысл данного коэффициента:

$$R^2 = \frac{ESS}{TSS}. \quad (21)$$

Основная проблема применения (выборочного) R^2 заключается в том, что его значение увеличивается (не уменьшается) от добавления в модель новых переменных, даже если эти переменные никакого отношения к объясняемой переменной не имеют. Для того чтобы увидеть это рассмотрим такой пример: предположим, что оценивается регрессионная зависимость y от x_1 и x_2, и строится уравнение вида: $\hat{y} = \omega_0 + \omega_1 x_1 + \omega_2 x_2$,

далее, предположим, что оценивается регрессионная зависимость y только от x_1, в результате получим: $\hat{y} = \omega_0 + \omega_1 x_1$, его можно переписать в виде: $\hat{y} = \omega_0 + \omega_1 x_1 + 0x_2$.

Если сравнить эти уравнения, то коэффициенты в первом из них свободно определялись с помощью метода наименьших квадратов на основе

данных для зависимости y, x_1 и x_2 при обеспечении наилучшего качества оценки. Однако во втором уравнении коэффициент при x_2 был произвольно установлен равным нулю, и оценивание не будет оптимальным, если только по случайному совпадению величина ω_2 не окажется равной нулю, когда оценки будут такими же. Следовательно, обычно коэффициент R^2 будет выше во втором уравнении, чем в первом, и он никогда не станет ниже. Конечно, если новая переменная на самом деле не относится к этому уравнению, то увеличение коэффициента R^2 будет, вероятно, незначительным.

Поэтому сравнение моделей с разным количеством факторов с помощью коэффициента детерминации, вообще говоря, некорректно. Для этих целей можно использовать альтернативные показатели.

Для того чтобы была возможность сравнивать модели с разным числом предикторов так, чтобы число предикторов не влияло на статистику R^2, обычно используется скорректированный коэффициент детерминации, в котором используются несмещённые оценки дисперсий:

$$R^2_{adj} = 1 - (1 - R^2)\left(\frac{n-1}{n-k}\right) . \quad (22)$$

где n — количество наблюдений, а k — количество параметров. Он даёт штраф за дополнительно включённые факторы.

Данный показатель всегда меньше единицы, но теоретически может быть и меньше нуля (только при очень маленьком значении обычного коэффициента детерминации и большом количестве факторов). Поэтому теряется интерпретация показателя как «доли». Тем не менее, применение показателя для сравнения моделей вполне обоснованно [14].

1.4.2 Функции ошибок

Среднеквадратическая ошибка

Точность подгонки модели под данные оценивается с помощью среднеквадратической ошибки по формуле (4).

Используется для выбора среди нескольких моделей, с последующим выбором модели с наименьшей данной ошибкой. Соответственно, алгоритм, рассчитывая регрессионную модель, стремится минимизировать этот коэффициент.

Средний модуль отклонения

Также были рассмотрены средние модули отклонения предсказанной величины от настоящего.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\widehat{y_i} - y_i| \quad (23)$$

В данной величине отображается усредненное значение отклонений фактических данных от предсказанных. Недостаток данной ошибки состоит в непонимании насколько велико значение относительно имеющихся данных.

Средняя абсолютная ошибка в процентах

Данная ошибка измеряется в процентах, что перекрывает недостаток ранее рассмотренной ошибки и упрощает интерпретацию. Коэффициент показывает процент отклонения предсказанных значений от фактических, он чувствителен к масштабу. Если действительные значения y_i близки к нулю, то в связи с делением на такие значения, ошибка сильно возрастает. Так же проблема возникает если одно из фактических значений равняется нулю.

$$MAPE = 100\% \cdot \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{|y_i|} \quad (24)$$

1.5 Проверка адекватности модели

Выделим основные предпосылки, которые должны выполняться для моделей линейной регрессии:

1. Математическое ожидание случайного отклонения ε_i равно нулю. $M(\varepsilon_i) = 0$ для всех наблюдений. Данное условие означает, что случайное отклонение в среднем не оказывает влияния на зависимую переменную.
2. Дисперсия случайных отклонений постоянна: $D(\varepsilon_i) = D(\varepsilon_j) = \sigma^2 = const$ для любых наблюдений i и j. Условие независимости дисперсии ошибки от номера наблюдения называется гомоскедастичностью. Невыполнимость этой предпосылки называется гетероскедастичностью.
3. Случайные отклонения ε_i и ε_j являются независимыми друг от друга для $i \neq j$. Невыполнимость этой предпосылки говорит о наличии автокорреляции.
4. Случайное отклонение должно быть независимо от объясняющих переменных. Данное условие предполагает выполнение следующего равенства:

$$\sigma_{\varepsilon_i, x_i} = M\left(\left(\varepsilon_i - M(\varepsilon_i)\right) \cdot \left(x_i - M(x_i)\right)\right) = M(\varepsilon_i, x_i) = 0.$$

5. Отсутствие мультиколлинеарности. Между объясняющими переменными отсутствует сильная линейная зависимость.
6. Случайные отклонения ε_i $(i = 1,2, \dots, n)$ имеют нормальное распределение.

Ниже рассмотрим некоторые предпосылки несколько подробней.

1.5.1 Распределение остатков

Одним из критериев адекватности модели линейной регрессии с помощью метода наименьших квадратов, является нормальное распределение остатков. В то время как для линейной регрессии с помощью метода наименьших модулей, распределение остатков должно иметь распределение Лапласа [13].

Рассмотрим несколько способов, позволяющих осуществить проверку гипотезы о нормальном распределении или распределении Лапласа.

Критерий с помощью QQ-Plot

Так называемый квантиль-квантиль график (QQ-Plot), это графический способ определения принадлежности эмпирической выборки к исследуемому закону распределения. График строится следующим образом:

- по оси x наносятся значения квантилей соответствующие предполагаемому закону распределения, построенные для каждого наблюдения;
- по оси y наносятся фактические значения квантилей.

Если обе выборки распределены по одному и тому же закону, то построенная линия должна соответствовать линии $y = x$ [6].

Критерий Колмогорова-Смирнова с поправкой Лиллиефорса

Данный критерий предназначен для проверки простых гипотез о принадлежности исследуемой выборки некоторому известному закону распределения. Рассмотрим изначально отдельно критерий Колмогорова-Смирнова.

Пусть X_n − выборка независимых одинаково распределенных величин, $F_n(x)$ − эмпирическая функция распределения, $F(x)$ − некоторая «истинная» функция распределения с известными параметрами. Статистика критерия определяется следующим образом:

$$D_n = \sup_x |F_n(x) - F(x)| \tag{25}$$

Через H_0 обозначим гипотезу о том, что выборка имеет рассматриваемое распределение. Тогда по теореме Колмогорова при справедливости проверяемой гипотезы:

$$\forall t > 0: \lim_{n\to\infty} P\left(\sqrt{n}\, D_n \le t\right) = K(t) = \sum_{j=-\infty}^{+\infty} (-1)^j e^{-2j^2t^2}. \quad (26)$$

Гипотеза H_0 отвергается, если статистика $\sqrt{n}\, D_n$ превышает квантиль распределения K_α заданного уровня значимости α, и принимается в ином случае.

В модернизированном методе проверка гипотезы проводится следующим образом:

- Оценивается выборочное среднее и дисперсия;
- находится максимальное отклонение между выборочной и теоретической интегральными функциями распределения;
- принимается решение, является ли статистически значимым наблюдаемое отклонение выборочной функции распределения от теоретической. В случае положительного ответа, нулевая гипотеза отвергается [13].

Критерий Крамера-Мизеса-Смирнова

В данном критерии используется следующая статистика:

$$S_\omega = n\,\omega^2 = \frac{1}{12n} + \sum_{i=1}^{n} \left(F(x_i,\theta) - \frac{2i-1}{2n}\right)^2, \quad (27)$$

где, n – объем выборки, $x_1, x_2, x_3, \dots, x_n$ – упорядоченные в порядке возрастания элементы выборки. $F(x_i,\theta)$ – основная гипотеза: исследуемое распределение подчиняется некоторому теоретическому закону. При объеме выборки $n > 40$ можно пользоваться квантилями распределения $n\omega^2$, приведенными в Таблице 1 [9]:

Таблица 1 – Квантили распределения $n\omega^2$

α	0,900	0,950	0,990	0,995	0,999
$n\omega^2(\alpha)$	0,3473	0,4614	0,7435	0,8694	1,1679

Критерий Андерсона-Дарлинга

При использовании этого критерия, вычисляется следующая статистика:

$$S = \sum_{i=1}^{n} \frac{2i-1}{n} \left(\ln\left(F(x_i)\right) + \ln\left(1 - F(x_{n+1-i})\right) \right), \quad (28)$$

Основная гипотеза: исследуемое распределение подчиняется некоторому теоретическому закону.

Критические значения для проверки нормальности распределения, при известных значениях среднего и дисперсии указаны в Таблице 2.

Таблица 2 – Критические значения теста Андерсона-Дарлинга при проверки на нормальность распределения

n	$\alpha = 0.15$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.025$	$\alpha = 0.01$
≥ 5	1.621	1.933	2.492	3.070	3.878

Критические значения для других распределений были приведены в трудах M. A. Stephens [17].

1.5.2 Мультиколлинеарность и методы ее устранения

Одной из основных проблем множественной линейной регрессии является тесная корреляционная взаимосвязь между объясняющими переменными, в матрице X это проявляется в виде линейной зависимости столбцов. Например, в задаче прогнозирования цены квартиры по ее параметрам линейная зависимость будет у признаков "площадь с учетом балкона" и "площадь без учета балкона". Формально для таких данных матрица $X^T X$ будет обратима, но из-за мультиколлинеарности у матрицы $X^T X$ некоторые собственные значения будут близки к нулю. Следовательно, в обратной матрице $(X^T X)^{-1}$ появятся экстремально большие собственные значения, т.к. собственные значения обратной матрицы – это $\frac{1}{\lambda_i}$, где λ_i – собственные значения матрицы $X^T X$. Итогом такого “шатания” собственных значений станет нестабильная оценка параметров модели, т.е. добавление нового наблюдения в набор тренировочных данных приведёт к совершенно другому решению [2].

Метод регуляризации Тихонова

Одним из способов регуляризации является регуляризация Тихонова, которая в общем виде выглядит как добавление нового члена к

среднеквадратичной ошибке. Часто матрица Тихонова выражается как произведение некоторого числа на единичную матрицу: $\Gamma = \lambda E$.

Следовательно, наше решение задачи будет выглядеть следующим образом:

$$\vec{\omega} = (X^T X + \lambda E)^{-1} X^T \vec{y}. \quad (29)$$

Такая регрессия называется гребневой регрессией (ridge regression). А гребнем является диагональная матрица, которую мы прибавляем к матрице $X^T X$, в результате получается такой операции гарантированно регулярная матрица. Такое решение уменьшает дисперсию, но становится смещенным, т.к. минимизируется также и норма вектора параметров [14].

Метод последовательного присоединения

1. Вычисляется матрица корреляций и выбирается регрессор, имеющий наибольшую корреляцию с выходной переменной;
2. К выбранному регрессору последовательно добавляются каждый из оставшихся регрессоров и вычисляются скорректированные коэффициенты детерминации для каждой из моделей. К модели присоединяется тот регрессор, который обеспечивает наибольшее значение скорректированного R^2;
3. Процесс присоединения завершается тогда, когда присоединение следующего регрессора не приводит к увеличению коэффициента детерминации по сравнению с предыдущим шагом. Это обстоятельство рассматривается как сигнал о том, что присоединение дополнительных регрессоров не увеличивает информативность уже включенных регрессоров.

Выявление и устранение мультиколлинеарности с помощью фактора инфляции дисперсии (VIF)

Фактор инфляции дисперсии (VIF) — мера мультиколлинеарности. Он позволяет оценить увеличение дисперсии заданного коэффициента регрессии, происходящее из-за высокой корреляции данных.

Чем выше фактор инфляции дисперсии для i-ой объясняющей переменной, тем сильнее линейная связь между этим и остальными объясняющими переменными. Показатель VIF часто используется в регрессионном анализе для выявления мультиколлинеарности и последующего исключения из модели тех предикторов, у которых VIF оказывается слишком высоким. Существуют разные мнения по поводу того, какое значение VIF считать пороговым. Обычно критическим считают значение $VIF = 5$, несколько реже $VIF = 10$. Некоторые исследователи также отмечают, что в биологических исследованиях (в частности, в

экологии), где наблюдаемые "сигналы" слабы, имеет смысл применять в качестве порогового значение $VIF = 2$.

Использование VIF включает следующие шаги:

Шаг 1. Для каждой объясняющей переменной X_i строится регрессионная модель его зависимости от остальных, например если $i = 1$:

$$X_i = \alpha_0 + \alpha_2 X_2 + \alpha_3 X_3 + \cdots + \alpha_n X_n + \varepsilon \quad (30)$$

Шаг 2. На основе полученной регрессионной модели для каждой X_i рассчитывается VIF по формуле:

$$VIF_i = \frac{1}{1 - R_i^2} \quad (31)$$

где $R_i^2 -$ коэффициент детерминации, модели построенной на первом шаге.

Шаг 3. Полученные для каждой объясняющей переменной значения VIF сравниваются с выбранным пороговым значением. Переменная с максимальным VIF, превышающем пороговое значение, исключается из анализа.

Шаги повторяются, пока в модели не останутся предикторы с низкими VIF [9].

2. Решение задачи предсказания просмотров видео методами регрессионного анализа

2.1 Описание набора данных и задачи исследования

Для анализа использовался набор данных, основанный на показателях с медиахостингового сайта YouTube. В набор входят следующие переменные: идентификатор видео, дата появления в разделе трендов, наименование, название канала выложившего видео, идентификатор категории, время публикации, поисковые тэги, количество лайков (likes), количество дизлайков (dislikes), количество комментариев, ссылку на видео, возможность оставлять комментарии, оценки, метка удалённого видео, описание видео.

Была поставлена задача, построить модели регрессии, которые объясняли бы как влияют на количество положительных оценок такие факторы как: количество просмотров, количество отрицательных оценок, количество комментариев. А так же выяснить влияние поисковых тэгов на количество положительных оценок видео.

Статистический анализ набора данных проводился средствами языка программирования R. Программный код представлен в Приложении.

2.2 Первичная обработка данных

Для начала были убраны все видео, у которых не было поисковых тэгов, и отключены возможности комментирования и оценивания, поскольку эти данные нам понадобятся для дальнейшей обработки. Так же, за ненадобностью были убраны: описание, время публикации, идентификатор категории и видео, время попадания в раздел трендов и ссылка на видео. Далее были выделены три тэга имеющие наибольшую популярность, а именно: «Путин», «Россия», «Юмор».

2.3 Построение регрессионных моделей

В первую очередь была построена корреляционная матрица, показывающая коэффициенты корреляции между предикторами: количеством положительных оценок (likes), количеством отрицательных оценок (dislikes), количеством комментариев (comment count), количеством использования поисковых тэгов (PUTIN, ROSSIYA, YUMOR).

	views	likes	dislikes	comment_count	PUTIN	ROSSIYA	YUMOR
views	1.00000000	0.85462137	0.488419881	0.764571041	-0.025868333	-0.027649215	0.019959984
likes	0.85462137	1.00000000	0.381143950	0.847083097	-0.028323919	-0.031782433	0.035014691
dislikes	0.48841988	0.38114395	1.000000000	0.488466317	0.073295371	0.008426511	0.008519472
comment_count	0.76457104	0.84708310	0.488466317	1.000000000	-0.002730951	-0.016448662	0.007725587
PUTIN	-0.02586833	-0.02832392	0.073295371	-0.002730951	1.000000000	0.425940868	-0.039776031
ROSSIYA	-0.02764922	-0.03178243	0.008426511	-0.016448662	0.425940868	1.000000000	0.002730040
YUMOR	0.01995998	0.03501469	0.008519472	0.007725587	-0.039776031	0.002730040	1.000000000

Рисунок 1 – Корреляционная матрица объясняющих и объясняемой переменных

Как можно заметить из данной матрицы, значительно коррелируют следующие две объясняющих переменных: количество просмотров и количество комментариев. Чтобы проверить их влияние были построены отдельные модели. В свою очередь теги, оказывают слишком малое влияние на количество положительных оценок, в связи с этим их можно исключить из моделей. Построим графики попарных зависимостей между переменными моделями.

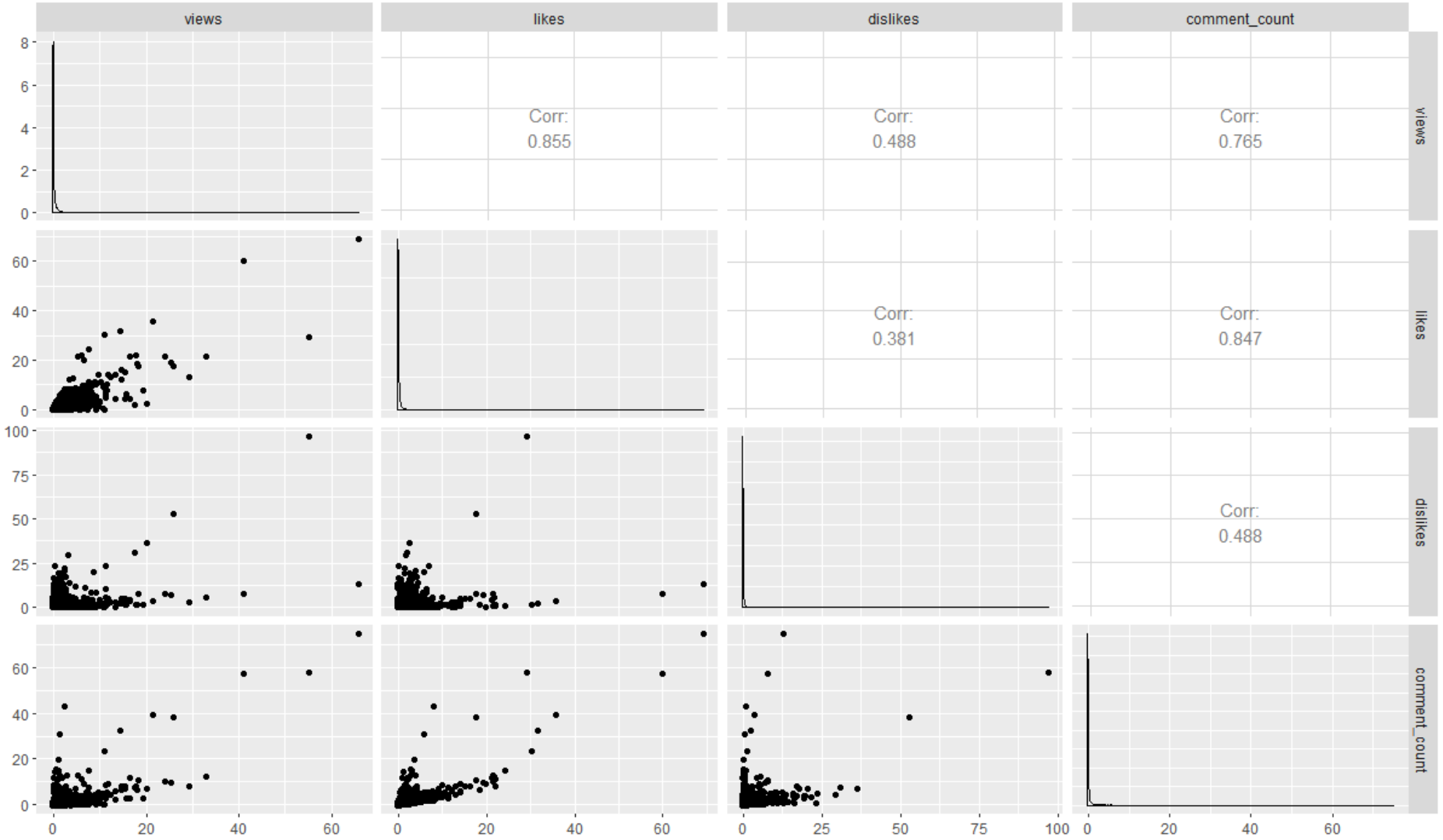

Рисунок 2 – Графики попарных зависимостей между переменными моделям

После нормировки данных можно переходить к построению моделей. Для начала проверим, имеет ли место мультиколлинеарность, для чего проведем отбор предикторов путем пошагового включения на основе значений коэффициентов детерминации. Построим три модели линейной регрессии в таком порядке:

1. $y_1 = \omega_0^1 + \omega_1^1 x_1;$
2. $y_2 = \omega_0^2 + \omega_1^2 x_1 + \omega_2^2 x_2;$
3. $y_3 = \omega_0^3 + \omega_1^3 x_1 + \omega_2^3 x_2 + \omega_3^3 x_3.$

где, $x_1, x_2, x_3 -$ количество положительных оценок, отрицательных оценок, комментариев, соответственно.

Рассмотрим их скорректированные коэффициенты детерминации:

- $R1^2_{adj} = 0.7218$
- $R2^2_{adj} = 0.7226$
- $R3^2_{adj} = 0.8344$

Скорректированный коэффициент детерминации с добавлением предикторов не уменьшается, что говорит об отсутствии мультиколлинеарности между предикторами.

Для большей уверенности проведем дополнительную проверку с помощью фактора инфляции дисперсии.

Получим следующие показатели:

- $VIF_{views} = 2.50$
- $VIF_{dislikes} = 1.38$
- $VIF_{comment} = 2.56$

Для всех трех предикторов фактор инфляции дисперсии не превышает критического значения $VIF = 5$. Данный критерий подтверждает отсутствие мультиколлинеарности.

Поскольку мультикорреляции между объясняющими переменными не возникает, будем использовать модель, в которую входят все три переменных.

В итоге нужно построить три модели, а именно:

1. Зависимость положительных оценок от просмотров, отрицательных оценок, комментариев с помощью линейной регрессии по методу наименьших квадратов (путем настройки на обучающей выборке и путем перекрестной проверки);
2. Зависимость положительных оценок от просмотров, отрицательных оценок, комментариев, с помощью алгоритма «Случайный лес»;
3. Зависимость положительных оценок от просмотров, отрицательных оценок, комментариев с помощью LAD-регрессии по методу наименьших модулей.

Линейная регрессия с помощью метода наименьших квадратов

С помощью стандартной функции языка R ($lm()$) получили первую модель линейной регрессии с помощью метода наименьших квадратов.

$$\widehat{y_1} = 0.0006501275 + 0.5051697562x_1 - 0.1247456773x_2 + 0.5591088919x_3.$$

Из уравнения видно, что комментарии сильнее всего влияют на количество положительных оценок. Знаки коэффициентов регрессии говорят о характере влияния на результативный признак. В нашем случае количество просмотров и комментариев увеличивают количество положительных оценок, а количество отрицательных оценок уменьшает число положительных.

Линейная регрессия с помощью перекрестной проверки

В исследовании было принято взять стандартное количество частей разделения выборки – 10 блоков.

Обозначим каждый из блоков разделения выборки K_i $(i = 1,2, \dots ,10)$. В первую очередь, была построена модель, обученная на множестве $K_2 \cup K_3 \cup \dots \cup K_{10}$, а в качестве проверочной выборки выбрано множество K_1. После этого построена модель, обученная на множестве $K_1 \cup K_3 \cup \dots \cup K_{10}$, а в качестве проверочной выборки выбрано множество K_2. Процесс повторяется 10 раз, и каждая из частей используется в качестве проверочной выборки. После чего мы получаем 10 моделей, каждую с собственными коэффициентами, которые после усредняются.

Получено следующее уравнение регрессии:

$$\widehat{y_2} = 0.004638021 + 0.507954x_1 - 0.01945x_2 + 0.5375936x_3.$$

Уравнение практически не отличается от модели, обученной на одной выборке. Это может быть связано с тем, что количество данных в нашем наборе велико и каждая из 10 моделей обучается практически одинаково.

Модель, построенная с помощью алгоритма «Случайный лес»

Данная модель является набором решающих правил и не может быть записана в виде функциональной зависимости. Рассмотрим одно решающее дерево (Рис.3).

Корень дерева разделяется по критерию количества положительных оценок, если их количество менее чем 164209 то мы идем к левому узлу, в ином случае к правому. В случае перехода к левому узлу дерево разбивается, опираясь на количество положительных оценок, и если рассмотреть случай, когда количество положительных оценок у видео менее 6229, то дерево

спрогнозирует среднее количество просмотров по данному множеству, т.е. 3751.025 просмотров. Это и есть один из листов дерева решений.

Линейная регрессия с помощью метода наименьших модулей

С помощью встроенной функции optim(), были вычислены коэффициенты $\omega_0, \omega_1, \omega_2, \omega_3$ выражения: $\omega_0 + \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_3$. Для их вычисления минимизировалась следующая величина: $\sum_{i=1}^{n} |\hat{y} - y|$.

Последняя модель имеет следующее уравнение:

$$\widehat{y_4} = -0.01810656 + 0.34124632x_1 - 0.01128834x_2 + 0.69574768x_3.$$

В этом случае видны различия в коэффициентах по сравнению с МНК. Из чего следует что скорость возрастания положительных оценок при увеличении количества просмотров в модели LAD регрессии меньше, чем в модели МНК-регрессии. Данные различия могут быть связаны с различными функционалами которые минимизируются в этих моделях.

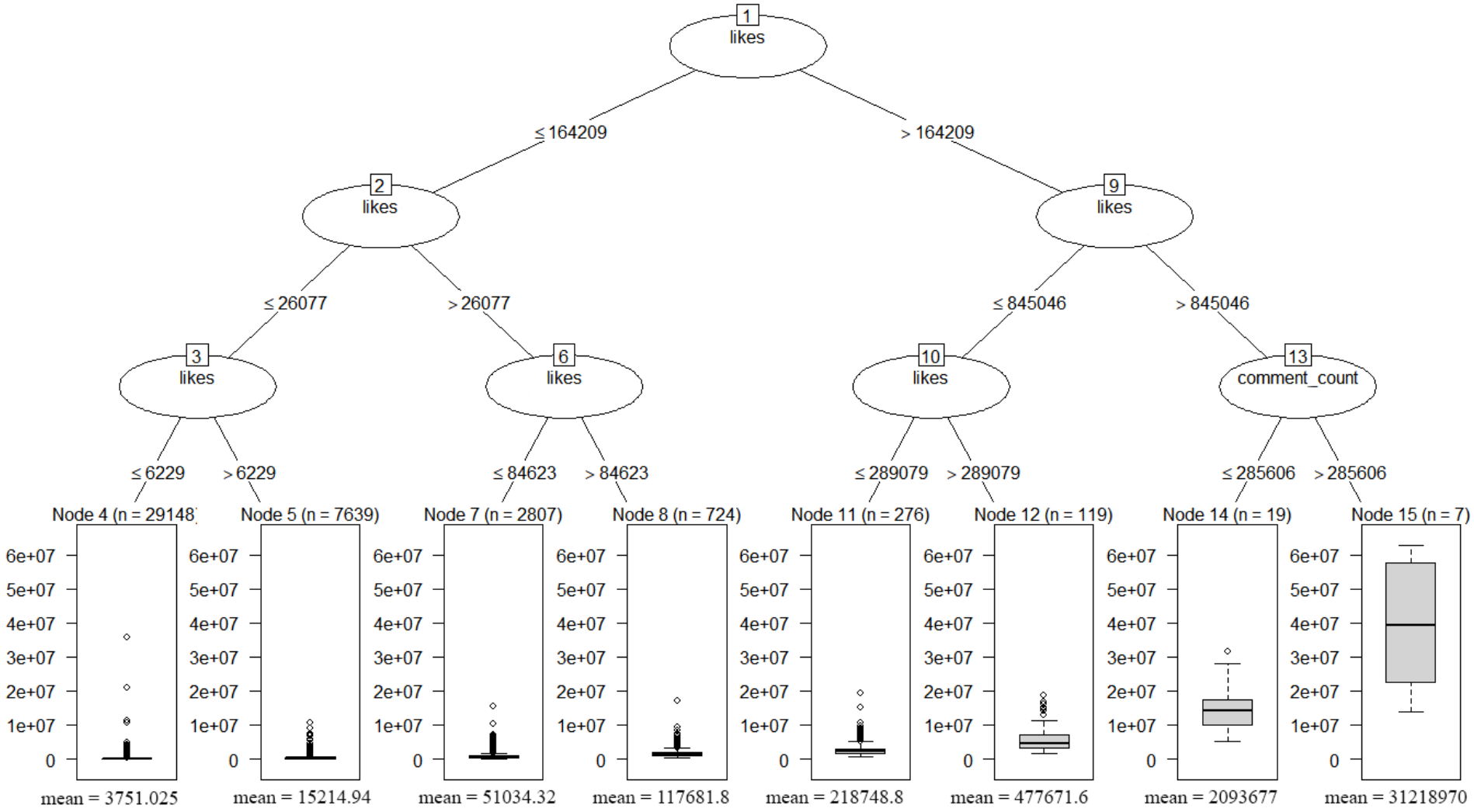

Рисунок 3 – Дерево решений для предсказания количества просмотров

2.4 Удаление выбросов из набора данных

Для того, что определить, имеются ли в наборе данных выбросы, способные повлиять на точность предсказания модели нужно вычислить и стандартизировать регрессионные остатки. Для стандартизации будем использовать стьюдентизированные остатки. Получим остатки следующим образом: $\varepsilon_i = \hat{y} - y$. После чего нужно получить стандартное отклонение, рассчитанное по остаткам модели, подогнанной после исключения из данных i-го наблюдения. Так же для вычисления стьюдентизированных остатков необходимо найти влияние точек на модель.

Построим полученные значения остатков для построенных моделей. Для визуального определения количества остатков можно построить графики с квантилями распределения Стьюдента, которые выступают в роли критических значений.

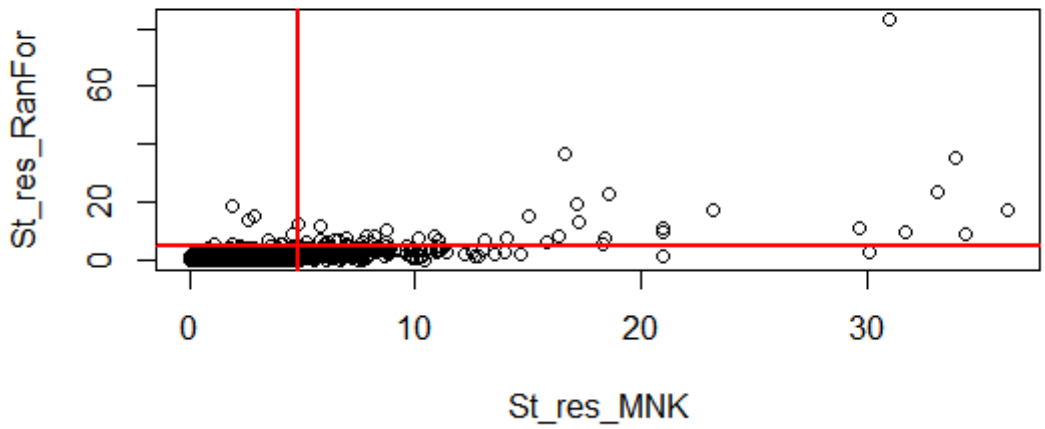

Рисунок 4 – Распределение остатков линейной регрессии с помощью МНК и алгоритма «Случайный лес» (по оси абсцисс – стьюдентизированные остатки, полученные из модели линейной регрессии построенной по методу МНК, по оси ординат – остатки, полученные из модели, построенной с помощью алгоритма «Случайный лес»), а линии являются квантилями распределения Стьюдента с уровнем значимости 0.05 (x = 4.822, y = 4.822)

Как видно из графика, все остатки, которые правее вертикальной линии и выше горизонтальной, являются выбросами. Вычислим и удалим все наблюдения, остатки которых превышают критическое значение равное приблизительно: $t_{\text{крит}} = 4{,}822$. Таких наблюдений получилось 226 записей из 35129.

После этого заново обучим наши модели, но уже на наборе без выбросов и определим, оказало ли удаление выбросов положительное влияние на качество моделей.

До удаления выбросов была получена линейная регрессионная модель со следующими коэффициентами:

$$\widehat{y_1} = 0.0006501275 + 0.5051697562x_1 - 0.1247456773x_2 + 0.5591088919x_3.$$

После удаления выбросов, получили следующее уравнение регрессии:

$$\widehat{y_1} = 0.00157941 + 0.43629213x_1 - 0.04126556x_2 + 0.69274141x_3,$$

где, x_1, x_2, x_3 – количество просмотров, отрицательных оценок и комментариев.

Сравним среднеквадратические ошибки (MSE).

Нумерация моделей идет соответственно порядку, описанному в Пункте 2.2.

Для предыдущего набора данных были получены следующие результаты:

- $MSE1_{train} = 0.1619,\ MSE1_{test} = 0.1942$
- $MSE2_{train} = 0.0493,\ MSE2_{test} = 0.1985$
- $MSE3_{train} = 0.1709,\ MSE3_{test} = 0.2112$
- $MSE4_{train} = 0.0451,\ MSE4_{test} = 0.0474$

После удаления выбросов значения стали равны:

- $MSE1_{train} = 0.0426,\ MSE1_{test} = 0.0431$
- $MSE2_{train} = 0.0079,\ MSE2_{test} = 0.0347$
- $MSE3_{train} = 0.0428,\ MSE3_{test} = 0.0436$
- $MSE4_{train} = 0.0450,\ MSE4_{test} = 0.0457$

Посмотрим на средний модуль отклонений. Значения, полученные до удаления выбросов:

- $MAE1_{train} = 0.1235, MAE1_{test} = 0.12$
- $MAE2_{train} = 0.0445, MAE2_{test} = 0.0876$
- $MAE3_{train} = 0.1119, MAE3_{test} = 0.1068$
- $MAE4_{train} = 0.1083, MAE4_{test} = 0.1041$

После удаления выбросов:

- $MAE1_{train} = 0.0855, MAE1_{test} = 0.0856$
- $MAE2_{train} = 0.0353, MAE2_{test} = 0.0728$
- $MAE3_{train} = 0.0860, MAE3_{test} = 0.0862$
- $MAE4_{train} = 0.0840, MAE4_{test} = 0.0838$

Удаление выбросов и точек, оказывающих значительное влияние на модель, уменьшило ошибки во всех моделях.

2.5 Проверка адекватности моделей линейной регрессии

Гетероскедастичность

Рассмотрим график зависимости отклонений от величины отклика.

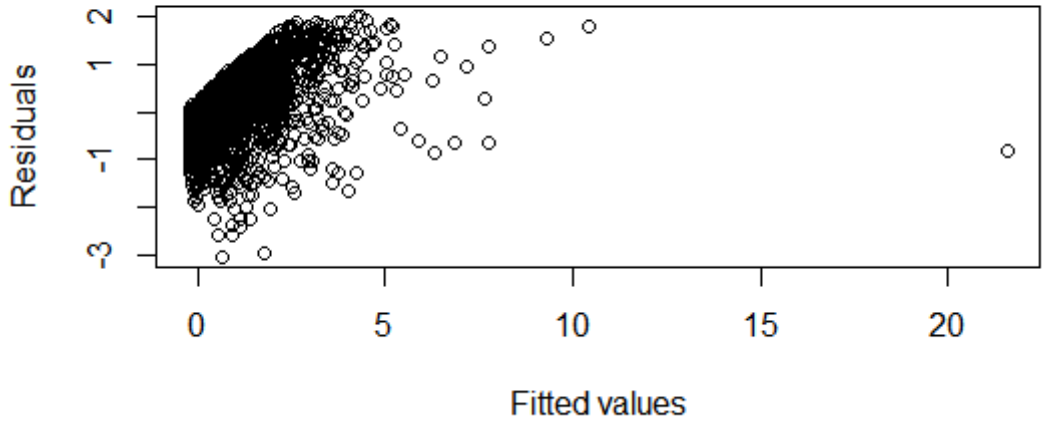

Рисунок 5 – График зависимости регрессионных остатков от величины отклика (по оси абсцисс – фактические значения, по оси ординат – регрессионные остатки)

Как видно из графика в выбранном наборе данных наблюдалась - гетероскедастичность. Гетероскедастичность – нарушение предположения о постоянстве дисперсии e_i (дисперсия возрастает с ростом значений y_i). Это одна из возможных проблем точности предсказания наших моделей.

Решить данную проблему можно путем преобразования исходных данных их производными, например, логарифмом, относительным изменением или другой нелинейной функцией.

Распределение регрессионных остатков

Рассмотрим распределения остатков моделей построенных с помощью метода наименьших квадратов и метода наименьших модулей. Для первого

распределение должно быть нормальным, тогда как для второго необходимо распределение Лапласа.

Проверим нормальность регрессионных остатков с помощью инструмента **QQ plot**.

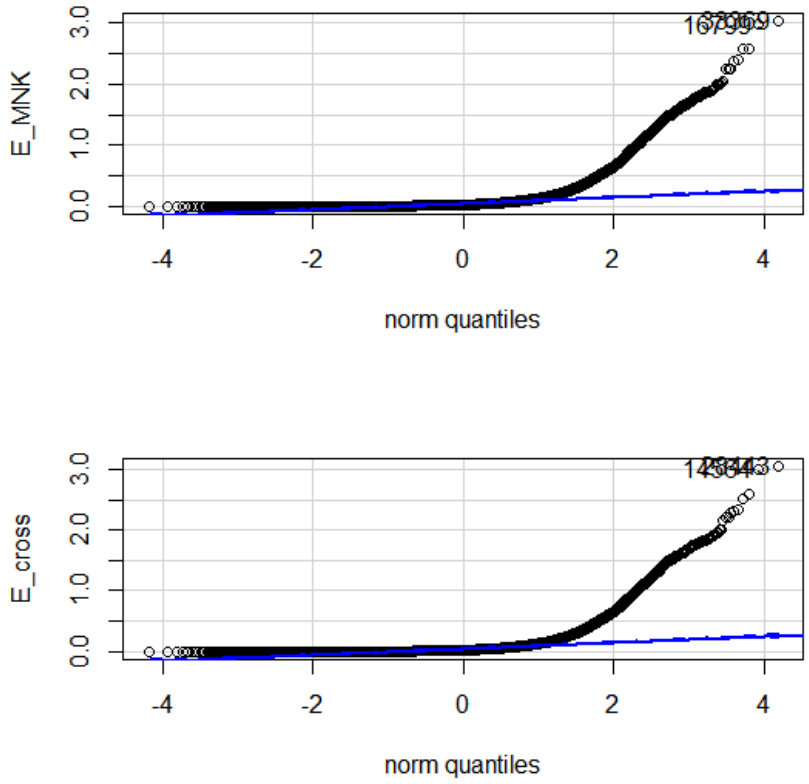

Рис. 6 QQ-Plot для стандартной линейной модели и модели кросс-валидации, соответственно (по оси абсцисс указаны теоретические квантили нормального распределения, по оси ординат – фактические значения квантилей)

Как видно из графиков, прямая далека от прямой $y = x$. Из чего можно сделать вывод, что по данному критерию гипотеза о нормальном распределении остатков отклоняется.

Критерий Колмогорова-Смирнова с поправкой Лилиефорса.

Для стандартной линейной модели и модели вычисленной с помощью кросс-валидации получим следующие значение $p - value$:

- $p - value_{MNK} = 2.2 \cdot 10^{-16}$
- $p - value_{cross} = 2.2 \cdot 10^{-16}$

Основная гипотеза о нормальности распределения остатков отклоняется.

Критерий Крамера-Мизера-Смирнова.

- $p - value_{MNK} = 0.3551$
- $p - value_{cross} = 0.3555$
- $p - value_{lad} = 0.348$

В данном критерии рассмотрены остатки для моделей МНК, кросс-валидации и LAD-регрессии, соответственно, с основной гипотезой: остатки имеют нормальное распределение (для МНК и кросс-валидации), распределение Лапласа (для LAD-регрессии). Причем критерий показывает достаточно высокие значения, для того чтобы принять основные гипотезы. Однако рассмотрим еще один.

Критерий Андерсона-Дарлинга

- $p - value_{MNK} = 1.72 \cdot 10^{-8}$
- $p - value_{cross} = 1.72 \cdot 10^{-8}$
- $p - value_{lad} = 1.72 \cdot 10^{-8}$

Рассматриваемый критерий также отвергает основную гипотезу для моделей, обученных с помощью МНК, кросс-валидации и LAD-регрессии, соответственно.

Можно сделать общий вывод по распределениям остатков. С большей долей вероятности остатки распределены не нормально для МНК и не имеют распределение Лапласа для LAD-регрессии, что как минимум можно видеть из графиков распределения остатков, и что доказывают все тесты кроме одного.

Невыполнение вышеописанных предпосылок затрудняет статистический анализ данных моделей на устойчивость.

Проверка качества регрессионных моделей

Для анализа качества моделей вычислим метрики качества на обучающей и тестовой выборках. Значения метрик на тестовой выборке должны быть приблизительно равны значениям на тестовой выборке.

Для начала был рассмотрен скорректированный коэффициент детерминации. Коэффициенты для моделей МНК, алгоритма «Случайный лес», кросс-валидации и LAD-регрессии, соответственно:

- $TrainR1^2_{adj} = 0.7730,\ TestR1^2_{adj} = 0.8118$
- $TrainR2^2_{adj} = 0.9580,\ TestR2^2_{adj} = 0.7630$
- $TrainR3^2_{adj} = 0.7722,\ TestR3^2_{adj} = 0.8099$
- $TrainR4^2_{adj} = 0.7602,\ TestR4^2_{adj} = 0.8003$

Если опираться на коэффициент детерминации, то лучше всего описывает исследуемые данные модель, построенная с помощью алгоритма «Случайный лес» у которого коэффициент детерминации на обучающей выборке получился максимальным.

Рассмотрим среднеквадратичную ошибку на обучающей и проверочной выборках, для моделей МНК, алгоритма «Случайный лес», кросс-валидации и LAD-регрессии, соответственно:

- $MSE1_{train} = 0.0426,\ MSE1_{test} = 0.0431$
- $MSE2_{train} = 0.0079,\ MSE2_{test} = 0.0347$
- $MSE3_{train} = 0.0428,\ MSE3_{test} = 0.0436$
- $MSE4_{train} = 0.0450,\ MSE4_{test} = 0.0457$

Как и в предыдущих пунктах рассмотрим отклонение для обучающей и проверочной выборки

- $MAE1_{train} = 0.0855, MAE1_{test} = 0.0856$
- $MAE2_{train} = 0.0353, MAE2_{test} = 0.0728$
- $MAE3_{train} = 0.0860, MAE3_{test} = 0.0862$
- $MAE4_{train} = 0.0840, MAE4_{test} = 0.0838$

Из рассмотренных ошибок, можно сделать следующие выводы: алгоритм «Случайный лес» предсказывает данные на обучающей и проверочной выборке с меньшей ошибкой, нежели иные модели. Также данный алгоритм лучше обучился на проверочной выборке, однако на проверочной выборке он показывает результаты не намного лучше, чем линейная регрессия.

Помимо описанных выше ошибок, можно было бы рассмотреть среднюю абсолютную ошибку в процентах, для тех же моделей. Однако данная ошибка плохо работает на нашем нормированном наборе данных:

- $MAPE1_{train} = 3259{,}27\ , MAPE1_{test} = 194{,}83$
- $MAPE2_{train} = 1341{,}91\ , MAPE2_{test} = 218{,}30$
- $MAPE3_{train} = 3065{,}64\ , MAPE3_{test} = 199{,}79$
- $MAPE4_{train} = 3231{,}98\ , MAPE4_{test} = 181{,}70$

В нашем случае такие большие значения ошибок связаны со спецификой исследуемого набора данных: большое число значений результирующего признака (33939 наблюдения) в нашем наборе имеют значения, по модулю меньшие, чем 1, а как было сказано в Главе 1, данная ошибка очень чувствительна к малым значениям, которые получаются в исследуемом наборе после нормировки данных.

ЗАКЛЮЧЕНИЕ

В исследовании была рассмотренная одна из актуальнейших задач анализа данных – задача прогнозирования, одним из основных методом решения которой является регрессионный анализ. Поставленная в данном исследовании цель - с помощью регрессионного анализа построить модели пригодные для прогнозирования популярности видео – была достигнута в полном объеме путем решения следующих задачи:

1. Изучение теории регрессионного анализа.
2. Обработка статистических данных для последующего анализа.
3. Обучение регрессионных моделей.
4. Оптимизация моделей под средством удаления выбросов в наборе данных.
5. Оценка адекватности моделей.

В ходе исследования были изучены методы линейной регрессии, а также метод регрессии, основанный на решающих деревьях. Выполнено обучение регрессионных моделей на обучающих выборках. После чего произведено удаление выбросов из набора данных для получения более точного результата.

Проведена оценка адекватности построенных линейных моделей путем проверки выполнения предпосылок линейной регрессии. В результате чего выяснилось наличие гетероскедастичности остатков. Кроме этого распределение остатков не соответствовало требуемым распределениям. В связи с этим, затрудняется статистический анализ данных моделей на устойчивость.

С помощью анализа значений различных функционалов ошибок для каждой из построенных моделей, сделан вывод о том, что алгоритм “Случайный лес” выдает наиболее точный прогноз на исследованном наборе данных.

Результаты данного исследования могут быть использованы для предсказания популярности видео по количеству просмотров, количеству отрицательных оценок, количеству комментариев. В последующем, с использованием полученных моделей может быть рассчитана прибыль, приносимая рекламодателю при внедрении в видео рекламы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Айвазян, С.А. Прикладная статистика и основы эконометрики / С.А. Айвазян, В.С. Мхитарян - М.: ЮНИТИ, 1998. — 1000 с.
2. Дрейпер, Н. Прикладной регрессионный анализ / Н. Дрейпер, Г.Смит. - М.: Вильямс, 2016. - 912 с.
3. Ефимова, М.Р. Общая теория статистик [Текст] / М.Р. Ефимова, Е.В. Петрова, В.Н. Румянцев. – М.: Инфра-М, 2004. – 416с.
4. Зарядов И. С. Введение в статистический пакет R: типы переменных, структуры данных, чтение и запись информации, графика. – М.: Издательство Российского университета дружбы народов, 2010. – 207 с.
5. Крыштановский, А.О. Ограничения метода регрессионного анализа [Электронный ресурс]. – http://socioline.ru/pages/ao-kryshtanovskij-ogranicheniya-metoda-regressionnogo-analiza
6. Курс лекций по машинному обучению на русскоязычном блоге Habr.com [Электронный ресурс]. – https://m.habr.com/company/ods/blog/323890/
7. Кустицкая, Т.А. Презентация с семинара по анализу данных. Линейная регрессия [Электронный ресурс]. – http://ikit.sfu-kras.ru/files/ikit/Representativness-_Kustitskaya.pdf
8. Магнус, Я.Р. Эконометрика. Начальный курс [Текст] / Я.Р. Магнус, П.К. Катышев, А.А. Пересецкий. - Учеб. — 6- изд., перераб. доп. - М.: Дело, 2004. - 576 с.
9. Материалы по диагностика регрессионных моделей [Электронный ресурс] – http://creativity.ipras.ru/texts/AspSem/lobanov_10_03_11.pdf
10. Михальченко Г.Е. Деревья принятия решений [Электронный ресурс]. – http://ikit.sfu-kras.ru/files/ikit/Derevya_-_Mihalchenko.pdf
11. Набор данных с платформы Kaggle [Электронный ресурс] – https://www.kaggle.com/datasnaek/youtube-new
12. Профессиональный информационно-аналитический ресурс MachineLearning.ru [Электронный ресурс]. – http://machinelearning.ru
13. Фёрстер Э. Методы корреляционного и регрессионного анализа / Э.Фёрстер, Рёнц Б. - "Финансы и статистика", 1983 г.- 304 с.
14. Хасти, Т. Введение в статистическое обучение с примерами на языке R [Текст] / Т.Хасти, Р. Тибширани, Г. Джеймс, Д. Уиттон. - ДМК Пресс, 2016 г, - 450 с.
15. Gonzalo, R. A. A Maximum Likelihood Approach to Least Absolute Deviation Regression / R.A. Gonzalo, Yinbo Li - [Электронный ресурс] - https://pdfs.semanticscholar.org/2b26/f91ab6028ff6ec73c309198c25804007a118.pdf
16. Hastie, T. The Elements of Statistical Learning / T. Hastie, R. Tibshirani, J. Friedman - Springer; 2nd edition, 2016 – 745 с.

17. Stephens, M. A. “Tests Based on EDF Statistics,” In: R. B. D’Agostino and M. A. Stephens, Eds., Goodness-of -Fit Techniques, Marcel Dekker, New York, 1986, 193 с.

18. Zuur, A. A protocol for data exploration to avoid common statistical problems [Электронный ресурс]. - https://doi.org/10.1111/j.2041-210X.2009.00001.

ПРИЛОЖЕНИЕ А

Код написанной программы на языке программирования R

```
library(stringr)
library(ggplot2)
library(GGally)
library(randomForest)
library(tsensembler)
library(party)
library(car)
library(corrplot)
library(nortest)
library(L1pack)
library(goftest)
Data <- read.csv("RUvideos.csv")
Data <- Data[Data$tags != "[none]",]
Data$description <- NULL
Data$publish_time <- NULL
Data$category_id <- NULL
Data$video_id <- NULL
Data$trending_date <- NULL
Data$thumbnail_link <- NULL
Data <- Data[Data$comments_disabled == FALSE,]
Data <- Data[Data$ratings_disabled == FALSE,]
Data$title <- as.character(Data$title)
Data$channel_title <- as.character(Data$channel_title)
Data$tags <- as.character(Data$tags)
Encoding(Data$title) <- "UTF-8"
Encoding(Data$channel_title) <- "UTF-8"
Encoding(Data$tags) <- "UTF-8"
Data$tags <- str_replace_all(Data$tags, "[[:punct:]]", "")
# Функции для перевода на транслит с русского
translate <- function(t) do.call(paste0, as.list(
  do.call(paste0, lapply(strsplit(toupper(t), ""),
                         function(s){s.temp <-lat[s];
                         s.temp[which(is.na(s.temp))]<-s[is.na(s.temp)];    unlist(s.temp)
}))))

translate <- function(t) do.call(paste0, lapply(unlist(strsplit(toupper(t), "")),
function(s)lat[s]))
```

```
        lat <- c("A","B","V","G","D","E","YO","ZH","Z","I","J","K","L","M","N","O","P","R","S","T","U","F","KH","C","CH","SH","SHH","``","Y","`","E","YU","YA", " ",",","A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q","R","S","T","U","V","W","X","Y","Z","0","1","2","3","4","5","6","7","8","9")
        rus <- c("А","Б","В","Г","Д","Е","Ё","Ж","З","И","Й","К","Л","М","Н","О","П","Р","С","Т","У","Ф","Х","Ц","Ч","Ш","Щ","Ъ","Ы","Ь","Э","Ю","Я", " ","|","A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q","R","S","T","U","V","W","X","Y","Z","0","1","2","3","4","5","6","7","8","9")
        names(lat)<-rus
        Tags <- NULL
        #Переводим транслитом все тэги
        for(i in 1:nrow(Data)){
          Data$tags[i] <- translate(Data$tags[i])
        }
        #Получаем датафрейм с наименованием тега и его частотой
        AllTags <- data.frame(table(unlist(strsplit(as.character(Data$tags[1:nrow(Data)]), ','))))
        #Выделяем 3 - частовстречаемых тегов
        AllTags <- AllTags[AllTags$Freq>100,]
        AllTags <- AllTags[order(-AllTags$Freq),]
        TopTags <- as.character(AllTags[1:3,1])
        InTag <- NULL
        for(i in 1:3){
          InTag[i] <- data.frame(as.numeric(str_detect(Data$tags, TopTags[i])))
        }

        # Формируем бинарную таблицу для данных тегов и нашего основного набора данных
        InTag <- data.frame(matrix(unlist(InTag), nrow = nrow(Data), byrow = F))
        names(InTag) <- TopTags

        #Объеденим основной датафрейм и получившийся бинарный
        Data <- cbind(Data,InTag)
        #Конец функции работы с тэгами
        rm(InTag,AllTags,Tags,TopTags,rus,lat,i,translate)
        #Посмотрим на графики, добавим на графики линии линейной регрессии для наглядности
        my_fn <- function(data, mapping, ...){
          p <- ggplot(data = data, mapping = mapping) +
            geom_point() +
```

```
    geom_smooth(method=lm, fill="blue", color="blue", ...)
  p
}
ggpairs(Data, columns = c(4:7))
corel <- data.frame(cor(Data[,c(4:7)],Data[,c(4:7)]))
g = ggpairs(Data,columns = 4:7, lower = list(continuous = my_fn))
g
# Нормировка данных
sd_views <- sd(Data$views)
sd_likes <- sd(Data$likes)
sd_dislikes <- sd(Data$dislikes)
sd_comment <- sd(Data$comment_count)
mean_views <- mean(Data$views)
mean_likes <- mean(Data$likes)
mean_dislikes <- mean(Data$dislikes)
mean_comment <- mean(Data$comment_count)

Data$views <- (Data$views - mean(Data$views))/sd(Data$views)
Data$likes <- (Data$likes - mean(Data$likes))/sd(Data$likes)
Data$dislikes <- (Data$dislikes - mean(Data$dislikes))/sd(Data$dislikes)
Data$comment_count <- (Data$comment_count - mean(Data$comment_count))/sd(Data$comment_count)
# Выборка для обучения и проверки
trainsample <- sample(nrow(Data), 0.75*nrow(Data))

#Строим линейную модель регрессии с коррелирующими регрессорами
modelMNK <- lm(data = Data, subset = trainsample, likes ~ views  + dislikes + comment_count)
# Общая информация о данной модели
summary (modelMNK)
# Отдельно коэф. при переменных в МНК
coef(modelMNK)
yhat <- predict(modelMNK, Data)
r_squared(Data$likes, yhat)
# Случайный лес
modelRanFor <- randomForest(likes ~ views + dislikes + comment_count, data = Data, subset = trainsample)
summary(modelRanFor)
mean(modelRanFor$rsq)
yhat <- predict(modelRanFor, Data)
r_squared(Data$likes, yhat)
```

```
# Среднеквадратические ошибки на обучающей выборке и на тестовой выборке для первой и второй модели
MSE_train_1 <- mean((Data$likes-predict(modelMNK,Data))[trainsample]^2)
MSE_test_1 <- mean((Data$likes-predict(modelMNK, Data))[-trainsample]^2)
MSE_train_2 <- mean((Data$likes-predict(ModelMnkWithTags,Data))[trainsample]^2)
MSE_test_2 <- mean((Data$likes-predict(ModelMnkWithTags, Data))[-trainsample]^2)
MSE_train_3 <- mean((Data$likes-predict(modelRanFor,Data))[trainsample]^2)
MSE_test_3 <- mean((Data$likes-predict(modelRanFor, Data))[-trainsample]^2)
# Среднее отклонение модулей
MAE_train_MNK <- mae(Data$likes[trainsample], predict(modelMNK,Data)[trainsample])
MAE_test_MNK <- mae(Data$likes[-trainsample], predict(modelMNK,Data)[-trainsample])
MAE_train_MNK2 <- mae(Data$likes[trainsample], predict(ModelMnkWithTags,Data)[trainsample])
MAE_test_MNK2 <- mae(Data$likes[-trainsample], predict(ModelMnkWithTags,Data)[-trainsample])
MAE_train_RanFor <- mae(Data$likes[trainsample], predict(modelRanFor,Data)[trainsample])
MAE_test_RanFor <- mae(Data$likes[-trainsample], predict(modelRanFor,Data)[-trainsample])
# Выбросы
# Стандатизируем остатки следующим образом:
E_MNK = predict(modelMNK, Data) - Data$likes
E_RanFor = predict(modelRanFor, Data) - Data$likes
sd_MNK = 0
sd_RanFor = 0
# Вычислим стандратное отклонение каждого остатка без его самого:
for (i in 1:length(E_MNK)){
  temp = E_MNK[-i]
  SD = sd(temp)
  sd_MNK = c(sd_MNK, SD)
}
sd_MNK = sd_MNK[-1]
for (i in 1:length(E_RanFor)){
  temp = E_MNK[-i]
  SD = sd(temp)
  sd_RanFor = c(sd_RanFor, SD)
```

```
}
sd_RanFor = sd_RanFor[-1]
rm(temp,SD,i)
# Вычислим стьюдентезированные остатки
h_MNK = sqrt(1 - sum(hat(predict(modelMNK, Data)))/length(E_MNK))
h_RanFor = sqrt(1 - sum(hat(predict(modelRanFor, Data)))/length(E_RanFor))
St_res_MNK = abs(E_MNK / (sd_MNK * h_MNK))
St_res_RanFor = abs(E_RanFor / (sd_RanFor * h_RanFor))
Qt_MNK = abs(qt(0.05/(length(E_MNK)*2), length(E_MNK)-sum(hat(predict(modelMNK, Data))-1)))
Qt_RanFor = abs(qt(0.05/(length(E_RanFor)*2), length(E_RanFor)-sum(hat(predict(modelRanFor, Data))-1)))
plot(St_res_MNK,St_res_RanFor)
abline(h=Qt_RanFor,col="red",lwd=2)
abline(v=Qt_MNK,col="red",lwd=2)
# Удаление элементов превышащих критическое значение квантиля с уровнем значимости 0.05
# И поправкой Бонферрони
delete_str = which(St_res_MNK > Qt_MNK | St_res_RanFor > Qt_RanFor)
Data_without_outliers <- Data[-delete_str,]
trainsample_without_outliers <-  setdiff(trainsample, delete_str)
Data_for_train <- Data_without_outliers[trainsample_without_outliers,]
#Появились строки с NA
delete_str_2 <- which(is.na.data.frame(Data_for_train$likes))
Data_for_train <- Data_for_train[-delete_str_2,]
trainsample_without_outliers <- setdiff(trainsample_without_outliers, delete_str_2)
rm(delete_str_2)
#Проведем заново все действия с таким набором без "выбросов"
# Линейная регрессия с помощью МНК
modelMNK_without_outliers <- lm(data = Data_for_train, likes ~ views + dislikes + comment_count)
# Линейная регрессия с помощью алгоритма случайного леса
modelRanFor_without_outliers <- randomForest(likes ~ views + dislikes + comment_count, data = Data_for_train)
#Ошибки
#MSE
MSE_train_without_outliers_MNK <- mean((Data_for_train$likes-predict(modelMNK_without_outliers,Data_for_train))^2)
MSE_test_without_outliers_MNK <- mean((Data_without_outliers$likes-predict(modelMNK_without_outliers,Data_without_outliers))[-trainsample_without_outliers]^2)
```

```
MSE_train_without_outliers_RanFor <- mean((Data_for_train$likes-predict(modelRanFor_without_outliers,Data_for_train))^2)
MSE_test_without_outliers_RanFor <- mean((Data_without_outliers$likes-predict(modelRanFor_without_outliers,Data_without_outliers))[-trainsample_without_outliers]^2)
#MAE
MAE_train_MNK_without_outliers <- mae(Data_for_train$likes, predict(modelMNK_without_outliers,Data_for_train))
MAE_test_MNK_without_outliers <- mae(Data_without_outliers$likes[-trainsample_without_outliers], predict(modelMNK_without_outliers,Data_without_outliers)[-trainsample_without_outliers])
MAE_train_RanFor_without_outliers <- mae(Data_for_train$likes, predict(modelRanFor_without_outliers,Data_for_train))
MAE_test_RanFor_without_outliers <- mae(Data_without_outliers$likes[-trainsample_without_outliers], predict(modelRanFor_without_outliers,Data_without_outliers)[-trainsample_without_outliers])
# Разница между моделями с выбросами и без
Delta_MSE_train_MNK = MSE_train_1 - MSE_train_without_outliers_MNK
Delta_MSE_test_MNK = MSE_test_1 - MSE_test_without_outliers_MNK
Delta_MAE_train_MNK = MAE_train_MNK - MAE_train_MNK_without_outliers
Delta_MAE_test_MNK = MAE_test_MNK - MAE_test_MNK_without_outliers
Delta_MSE_train_RanFor = MSE_train_3 - MSE_train_without_outliers_RanFor
Delta_MSE_test_RanFor = MSE_test_3 - MSE_test_without_outliers_RanFor
Delta_MAE_train_RanFor = MAE_train_RanFor - MAE_train_RanFor_without_outliers
Delta_MAE_test_RanFor = MAE_test_RanFor - MAE_test_RanFor_without_outliers
#Cross-Validation
cross_model <- lm(data = Data, likes ~ views + dislikes + comment_count)
E_MNK = predict(cross_model, Data) - Data$likes
sd_MNK = 0
# Вычислим стандратное отклонение каждого остатка без его самого:
for (i in 1:length(E_MNK)){
  temp = E_MNK[-i]
  SD = sd(temp)
  sd_MNK = c(sd_MNK, SD)
}
```

```
sd_MNK = sd_MNK[-1]
rm(temp,SD,i)
# Вычислим стьюдентезированные остатки
h_MNK = sqrt(1 - sum(hat(predict(modelMNK, Data)))/length(E_MNK))
St_res_MNK = abs(E_MNK / (sd_MNK * h_MNK))
Qt_MNK = abs(qt(0.05/(length(E_MNK)*2), length(E_MNK)-sum(hat(predict(modelMNK, Data))-1)))

# Удаление элементов превышающих критическое значение квантиля с уровнем значимости 0.05
# И поправкой Бонферрони
delete_str = which(St_res_MNK > Qt_MNK)
Data_cross <- Data[-delete_str,]
#Появились строки с NA
delete_str_2 <- which(is.na.data.frame(Data_cross$likes))
if(length(delete_str_2) > 0){
  Data_cross <- Data_cross[-delete_str_2,]
}
rm(delete_str_2, E_MNK, St_res_MNK, Qt_MNK, delete_str, h_MNK, cross_model)
coefs <- data.frame()
MSE_cross_delta <- data.frame()
MSE_cross_lin <- data.frame()
folds <- cut(seq(1,nrow(Data_cross)),breaks=10,labels=FALSE)
for(i in 1:10){
  trainIndexes <- which(folds==i,arr.ind=TRUE)
  trainData <- Data_cross[trainIndexes, ]
  testData <- Data_cross[-trainIndexes, ]
  model <- lm(data = Data_cross, subset = trainIndexes, likes ~ views + dislikes + comment_count)
  coefs <- rbind(coefs, coef(model))
# MSE_cross_delta <- rbind(MSE_cross_delta, mse(Data_cross$likes[-trainIndexes], predict(model, Data_cross)[-trainIndexes]))
# MSE_cross_lin <- rbind(MSE_cross_lin, mse(Data_cross$likes[-trainIndexes], predict(modelMNK,Data_cross)[-trainIndexes]))
}
MSE_cross_delta <- cbind(MSE_cross_delta, MSE_cross_lin)
rm(trainIndexes,trainData,testData,model, MSE_cross_lin)
coef1 <- mean(coefs[,1])
coef2 <- mean(coefs[,2])
coef3 <- mean(coefs[,3])
coef4 <- mean(coefs[,4])
```

```
y_hat_cross <- (coef1 + coef2 * Data_without_outliers$views + coef3 * Data_without_outliers$dislikes + coef4 * Data_without_outliers$comment_count)
y_hat_cross_train <- (coef1 + coef2 * Data_for_train$views + coef3 * Data_for_train$dislikes + coef4 * Data_for_train$comment_count)
# Ошибки в кросс-валидации
#MSE
MSE_test_cross <- mean((Data_without_outliers$likes[-trainsample_without_outliers]-y_hat_cross[-trainsample_without_outliers])^2)
MSE_train_cross <- mean((Data_for_train$likes-y_hat_cross_train)^2)
#MAE
MAE_train_cross <- mae(Data_for_train$likes, y_hat_cross_train)
MAE_test_cross <- mae(Data_without_outliers$likes[-trainsample_without_outliers], y_hat_cross[-trainsample_without_outliers])
# Разность кросс-валидации и модели без выбросов (Если знак плюс, то кросс хуже)
MSE_Delta_cross_withoutOut <- MSE_test_cross - MSE_test_without_outliers_MNK
MAE_Delta_cross_withoutOut <- MAE_test_cross - MAE_test_MNK_without_outliers
# LAD регрессия
lad <- function(p){
  pred <- p[1] + p[2]*Data_without_outliers$views + p[3]*Data_without_outliers$dislikes + p[4]*Data_without_outliers$comment_count
  mean(sum(abs(pred - Data_without_outliers$likes)))
}
opt <- optim(fn = lad, par = c(0,0,0,0))
coefs_lad <- opt$par
y_hat_lad <- coefs_lad[1] + coefs_lad[2] * Data_without_outliers$views + coefs_lad[3] * Data_without_outliers$dislikes + coefs_lad[4] * Data_without_outliers$comment_count
y_hat_lad_train <- coefs_lad[1] + coefs_lad[2] * Data_for_train$views + coefs_lad[3] * Data_for_train$dislikes + coefs_lad[4] * Data_for_train$comment_count
#MSE
MSE_train_lad <- mse(Data_for_train$likes, y_hat_lad_train)
MSE_test_lad <- mse(Data_without_outliers$likes[-trainsample_without_outliers], y_hat_lad[-trainsample_without_outliers])
#MAE

MAE_train_lad <- mae(Data_for_train$likes, y_hat_lad_train)
```

```
MAE_test_lad <- mae(Data_without_outliers$likes[-trainsample_without_outliers], y_hat_lad[-trainsample_without_outliers])
#MAPE для всех моделей
MAPE_train_1 <- 100 * mean((abs(Data_for_train$likes-predict(modelMNK, Data_for_train))/abs(Data_for_train$likes)))
MAPE_test_1 <- 100 * mean((abs(Data_without_outliers$likes-predict(modelMNK_without_outliers, Data_without_outliers))/abs(Data_without_outliers$likes))[-trainsample_without_outliers])
MAPE_train_2 <- 100 * mean((abs(Data_for_train$likes-predict(modelRanFor_without_outliers, Data_for_train))/abs(Data_for_train$likes)))
MAPE_test_2 <- 100 * mean((abs(Data_without_outliers$likes-predict(modelRanFor_without_outliers, Data_without_outliers))/abs(Data_without_outliers$likes))[-trainsample_without_outliers])
MAPE_train_3 <- 100 * mean((abs(Data_for_train$likes-y_hat_cross_train)/abs(Data_for_train$likes)))
MAPE_test_3 <- 100 * mean((abs(Data_without_outliers$likes-y_hat_cross)/abs(Data_without_outliers$likes))[-trainsample_without_outliers])
MAPE_train_4 <- 100 * mean((abs(Data_for_train$likes-y_hat_lad_train)/abs(Data_for_train$likes)))
MAPE_test_4 <- 100 * mean((abs(Data_without_outliers$likes-y_hat_lad)/abs(Data_without_outliers$likes))[-trainsample_without_outliers])
# Остатки
plot(abs(Data_without_outliers$likes - predict(model_lad,Data_without_outliers)))
E_MNK <- abs(Data_without_outliers$likes - predict(modelMNK_without_outliers,Data_without_outliers))
E_RanFor <- abs(Data_without_outliers$likes - predict(modelRanFor_without_outliers,Data_without_outliers))
E_cross <- abs(Data_without_outliers$likes - y_hat_cross)
E_lad <- abs(Data_without_outliers$likes - y_hat_lad)
# Проверка на нормальность остатков линейных моделей
# Построим Q-Q plot (Quantile-Quantile test) если линия близка к прямой, то все хорошо
qqPlot(E_MNK)
qqPlot(E_cross)
# Тест Колмогорова-Смирнова
# Vодифицированный вариант критерия Колмогорова-Смирнова, специально для проверки нормальности.
# Он реализован с помощью функции lillie.test() – критерий Лиллифорса
lillie.test(E_MNK)
```

```
lillie.test(E_cross)
#Тест Крамера фон Мизеса
cvm.test(E_MNK, "pnorm", 0, var(E_MNK))
cvm.test(E_cross, "pnorm", 0, var(E_MNK))
cvm.test(E_lad, "plaplace", 0,sqrt(2/var(E_lad)))
# Тест Андерсона- Дарлинга
ad.test(E_MNK, "pnorm", 0, var(E_MNK))
ad.test(E_cross, "pnorm", 0, var(E_MNK))
ad.test(E_lad, "plaplace", 0,sqrt(2/var(E_lad)))
# Денормировка
y_denorm <- data.frame()
y_denorm <- rbind(y_denorm, Data_without_outliers$likes * sd_likes + mean_likes)
y_denorm <- rbind(y_denorm, predict(modelMNK, Data_without_outliers) * sd_likes + mean_likes)
y_denorm <- rbind(y_denorm, predict(modelMNK, Data_without_outliers) * sd_likes + mean_likes)
y_denorm <- rbind(y_denorm, predict(modelRanFor, Data_without_outliers) * sd_likes + mean_likes)
y_denorm <- rbind(y_denorm, predict(modelMNK_without_outliers, Data_without_outliers) * sd_likes + mean_likes)
y_denorm <- rbind(y_denorm, predict(modelRanFor_without_outliers, Data_without_outliers) * sd_likes + mean_likes)
y_denorm <- rbind(y_denorm, predict(Non_linear_model) * sd_likes + mean_likes)
y_denorm <- rbind(y_denorm, y_hat_cross * sd_likes + mean_likes)
y_denorm <- rbind(y_denorm, y_hat_lad * sd_likes + mean_likes)
y_denorm <- data.frame(t(y_denorm))

names(y_denorm) <- c('Real','modelMNK', 'RanFor', 'modelMNK_without_outliers', 'modelRanFor_without_outliers', 'Non_linear_model', 'model_crassvol', 'lad-regression')
```

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра прикладной математики и компьютерной безопасности

УТВЕРЖДАЮ
Заведующий кафедрой
_____ А.А. Кытманов
« 8 » июля 2019 г.

БАКАЛАВРСКАЯ РАБОТА
01.03.04 – Прикладная математика
Применение регрессионного анализа для расчета прогнозных значений популярности видео

Руководитель _____ доцент каф. ПМКБ, к.ф.-м.н. Т.А. Кустицкая
04.07.2019

Выпускник _____ А.И. Перебигайло
01.07.2019

Красноярск 2019