**Министерство образования и науки РФ**

**Федеральное государственное автономное образовательное учреждение**

**Высшего образования**

**«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**ВЫСШАЯ ШКОЛА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ИНФОРМАЦИОННЫХ СИСТЕМ**

Специальность: 09.03.03. – Прикладная информатика

Отчет по курсу

**«**Основы построения систем связи и передачи данных**»**

Выполнили:

Студенты 3 курса

Группы 11-405

Р.Р. Гилязова

К.Р. Аязгулова

С.В. Ширгалеев

Проверил:

П.А. Новиков

г. Казань, 2016 г.

Содержание

1. [Описание набора данных 3](#_Toc471675564)
2. [Преобразования, которые были совершены с данными 4](#_Toc471675565)
3. [Описание методов, которые были использованы 7](#_Toc471675566)
4. [Оценка качества работы алгоритмов 8](#_Toc471675567)
5. [Вывод 9](#_Toc471675568)
6. [Приложение №1. Листинг программы 11](#_Toc471675569)

# Описание набора данных

Так часто мы сталкиваемся с тем, что после просмотра трейлера к фильму мы с нетерпением ждем его полнометражной версии, однако он оказывается совершенно провальным. В силу этого возникает вопрос: можем ли мы предугадать успешность фильма до его выхода на экраны?

Перед нами встала задача создать такую модель, которая предсказывала бы успешность фильма до его релиза по набору определенных критериев. Для этого мы использовали набор данных с сайта IMDB, включающий в себя более 5000 фильмов. Набор данных состоит из фильмов, снятых на протяжении последних 100 лет в более чем 60 странах мира. В нем представлены работы 2399 режиссеров, а также тысячи актеров и актрис.

Для каждого из фильмов описаны 28 признаков: “название фильма”, “количество критиков, написавших рецензию”, “количество лайков фильма на facebook”, “продолжительность”, “имя режиссера”, “количество лайков режиссера на facebook”, “имя третьего актера”, “количество лайков третьего актера на facebook”, “имя второго актера”, “количество лайков второго актера на facebook”, “имя первого актера”, “количество лайков первого актера на facebook”, “бюджет фильма”, “рейтинг содержимого”, “количество проголосовавших пользователей”, “общее количество лайков актерского состава на facebook”, “количество лиц на постере”, “ключевые слова сюжета”, “ссылка на фильм на сайте IMDB”, “количество пользователей, написавших отзыв”, “язык”, “страна”, “доход фильма”, “год выпуска”, “рейтинг на сайте IMDB”, “соотношение сторон экрана”, “жанр”.

Наша задача относится к классу задач регрессии, т.е. результат (рейтинг фильма), который нужно предугадать – количественный, является переменной величиной и зависит от других переменных. Данный тип задач решается с использованием регрессионного анализа - поиск функции ***f***, которая описывает эту зависимость. Задачи данного класса используются при прогнозировании, когда необходимо предсказать будущий результат на основе предыдущих значений.

Проанализировав выборку, мы убедились, что большинство признаков является числовыми. Поэтому цель нашего проекта – выяснить, можно предугадать рейтинг фильма, используя в качестве входных данных только его числовые признаки.

# Преобразования, которые были совершены с данными

Качественные данные — это необходимое условие для создания эффективных моделей прогнозирования. Реальные данные собираются для последующей обработки из разных источников и процессов. Они могут содержать ошибки и повреждения, негативно влияющие на качество набора данных. Многие алгоритмы машинного обучения плохо работают с неподготовленными наборами данных, поэтому, перед тем как приступить к обучению той или иной модели, следует преобразовать данные.

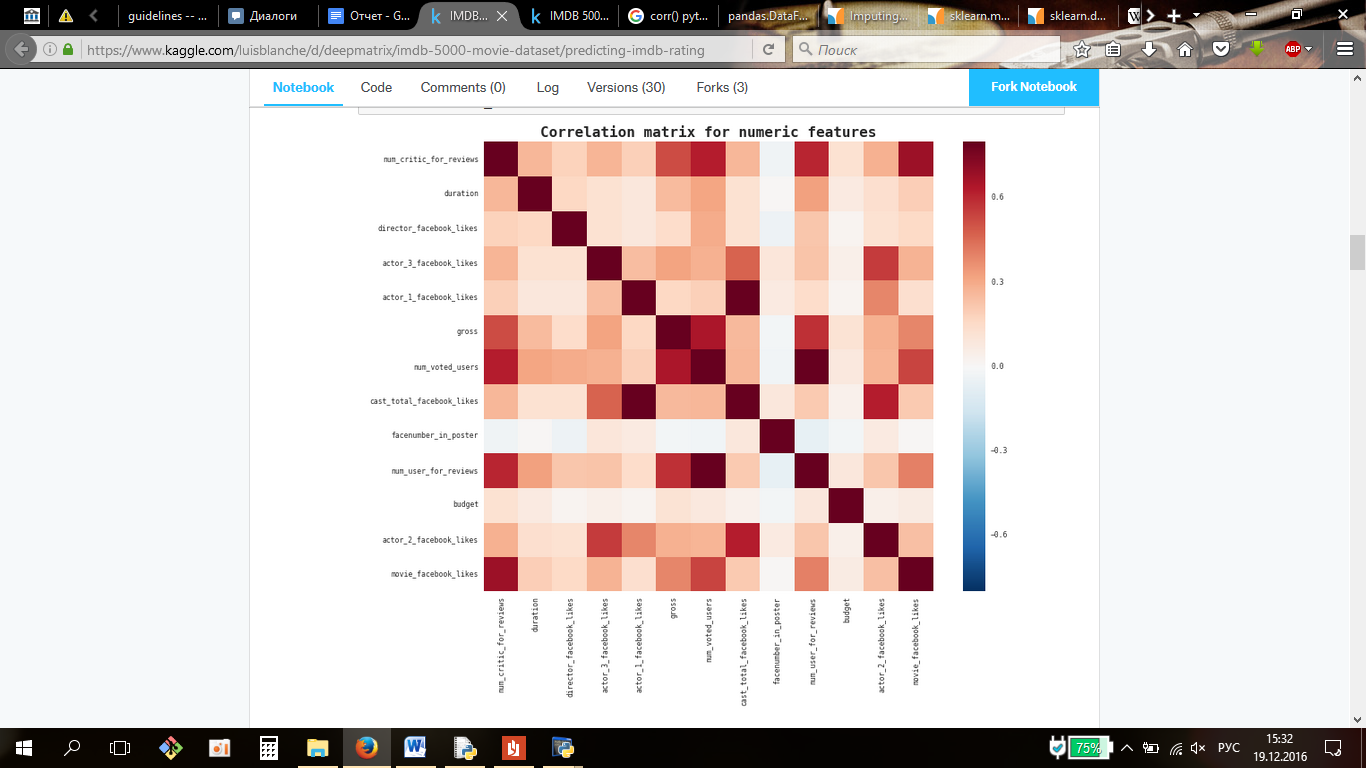
В первую очередь следует проверить наличие пропущенных значений в наборе данных. Если таковые имеются, то очевидная стратегия - это отбрасывать строки или столбцы с пустыми значениями в наборе данных. Однако данных способ может привести к потере полезных и значимых данных. Наилучшая стратегия - приписывать недостающие значения, методом вывода этих значений по известным данным. Для этого мы воспользовались классом Imputer, который предоставляет способы вставки пропущенных значений в выборку, используя среднее значение, медиану или наиболее часто встречающееся значение из строки или столбца, в котором недостающие значения расположены.

Более того, многие алгоритмы машинного обучения чувствительны к масштабированию данных. Обычно для масштабирования используется стандартизация данных – придание значениям каждого атрибута свойств стандартного нормального распределения: нулевое математическое ожидание, единичное среднеквадратичное отклонение. Новые значения выборки считаются по следующей формуле:

где µ - математическое ожидание, – среднеквадратичное отклонение.

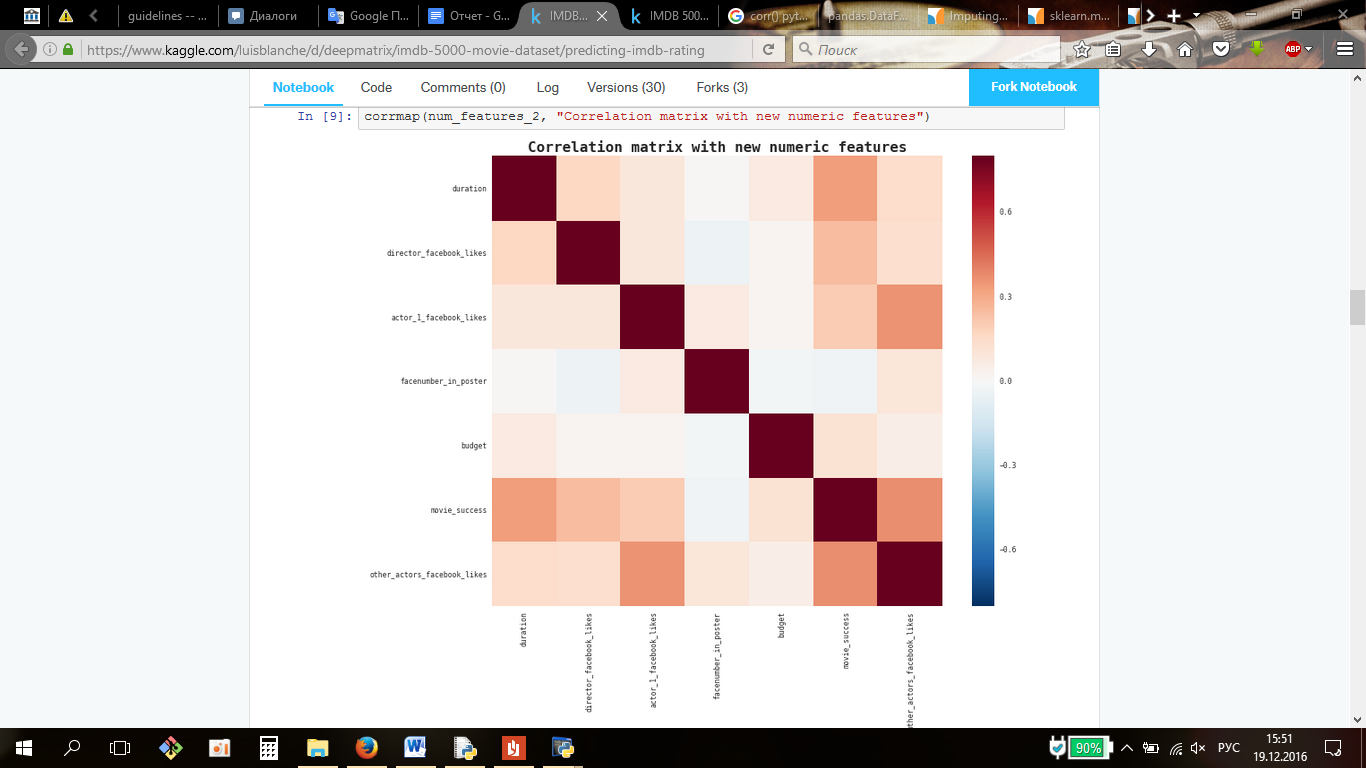
Для стандартизации нашего набора данных мы воспользовались классом StandardScaler библиотеки scikit-learn.

Для дальнейшей работы с данными нам нужно определить зависимость между двумя наборами данных: массивом количественных признаков для каждого фильма и массивом рейтинга, для этого мы использовали функцию pearsonr(), входящую в библиотеку scipy. Данная функция считает коэффициент корреляции Пирсона, который измеряет линейную зависимость между двумя наборами данных. Строго говоря, коэффициент корреляции Пирсона требует, чтобы каждый набор данных имел нормальное распределение. Коэффициент колеблется в диапазоне от -1 до 1, 0 не подразумевает никакой корреляции. Если корреляция положительная - при увеличении первого набора данных, увеличивается и второй, при отрицательной корреляции увеличение первого набора данных приводит к уменьшению второго. Коэффициент Пирсона,  равный -1 или 1, определяет точную линейную зависимость между двумя выборками.

Однако определение коэффициента корреляции между количественными признаками и результатом не достаточно для полного анализа данных. Рассчитаем корреляционную матрицу попарно между собой для каждого количественного признака.

Визуализировав корреляционную матрицу на тепловой диаграмме, можно заметить, что параметр “общее количество лайков актерского состава на Facebook” сильно коррелирует с общим количеством лайков на Facebook. Это естественно, так как второй параметр является суммой остальных параметров, связанных с лайками. Так что мы можем просто удалить данные критерии и добавить 2 новых: “количество лайков первого актера на Facebook” и “количество лайков других актеров на Facebook”. Более того, существует большая зависимость между признаками “количество проголосовавших пользователей”, “количество критиков, написавших рецензии”, “количество пользователей, написавших рецензии”, “доход фильма”, “количество лайков фильма на Facebook”, поэтому все эти признаки мы решили объединить в один “успех фильма”.

У нас получился новый набор данных, обладающий 7 признаками. Применим ранее использованные нами функции для вычисления коэффициента корреляции между ними.



Наша задача относится к задачам обучения с учителем, т.е. известна конечная совокупность прецедентов с известными результатами – обучающая выборка. Именно на основе этих данных строится алгоритм, способный предсказывать результат. Для этого нашу выборку мы разделили на обучающую и тестовую, используя метод train\_test\_split библиотеки scikit-learn, в соотношении 70% и 30% соответственно.

# Описание методов, которые были использованы

После того, как данные были подготовлены, мы можем приступить непосредственно к обучению модели. Для этого были выбраны методы, наиболее подходящие для решения нашей задачи:

* Метод наименьших квадратов (LinearRegression)
* Случайный лес (RandomForest)
* Метод ближайших соседей (KNeighborsRegressor)

**Метод наименьших квадратов -** математический метод, применяемый для решения различных задач, основанный на минимизации суммы квадратов отклонений некоторых функций от искомых переменных.

Сущность метода: Пусть имеется ***n*** значений некоторой переменной ***y*** (это могут быть результаты наблюдений, экспериментов и т. д.) и соответствующих переменных ***x***. Задача заключается в том, чтобы взаимосвязь между ***y*** и ***x*** аппроксимировать некоторой функцией ***f(x,b)***, известной с точностью до некоторых неизвестных параметров ***b***, то есть фактически найти наилучшие значения параметров ***b***, максимально приближающие значения ***f(x,b)*** к фактическим значениям ***y***.

**Случайный лес -** алгоритм машинного обучения, который строит ансамбль случайных деревьев, каждое из которых обучается на выборке, полученной из исходной с помощью процедуры изъятия с возвращением. К достоинствам данного алгоритма можно отнести то, что он способен эффективно обрабатывать данные с большим числом признаков и классов, однако данный алгоритм чувствителен к переобучению.

Случайный лес устроен так, что раз мы не можем построить одно оптимальное дерево, то построим 1000 не оптимальных, но так как каждое не оптимально по-своему, они будут уравновешивать недостатки друг друга.

Каждое полученное дерево будет плохим, не оптимальным, может быть вообще неэффективным. Но вместе они образуют работающий лес, который нейтрализует недостатки и позволяет нам нормально обучать модель.

**Метод ближайших соседей –** простейший метод машинного обучения, основанный на оценивании сходства объектов. Функция регрессии в данном методе вычисляется путем расчета расстояния до каждого из объектов обучающей выборки, далее отбирается k объектов выборки с минимальным расстоянием, и предсказываемая величина вычисляется на основе полученных объектов.

Алгоритм способен выделить среди всех наблюдений k известных объектов (k-ближайших соседей), похожих на новый неизвестный ранее объект. На основе результатов ближайших соседей выносится решение касательно нового объекта. Важной задачей данного алгоритма является подбор коэффициента k – количество записей, которые будут считаться похожими.

# Оценка качества работы алгоритмов

Качествоммоделирегрессии называется адекватность построенной модели по исходным (наблюдаемым) данным.

Для оценки качества модели регрессии используются специальные показатели, для проверки качества работы наших моделей мы решили сравнить для каждого алгоритма его среднеквадратическую и среднюю абсолютную погрешности в условиях кросс-валидации.

Кросс-валидация (перекрёстная проверка) - это способ оценить умение модели работать на гипотетическом тестовом наборе, когда такой набор в явном виде получить невозможно. Процесс перекрестной проверки заключается в разбиении обучающей выборки на подвыборки для настройки алгоритма и оценки его средней ошибки на объектах контрольной подвыборки. Данная методика позволяет избежать переобучения модели, случая, когда модель подстраивается именно под конкретную выборку, а на других данных алгоритм будет выдавать неверные результаты.

Среднеквадратической погрешностью называется среднеквадратическое отклонение выборочного распределения статистических данных. Другими словами, ее можно использовать для оценки точности выборочного среднего значения. Среднеквадратическая погрешность показывает, насколько близка функция регрессии к множеству значений. Данный параметр вычисляется по формуле:

где – математическое ожидание, n – количество элементов выборки.

Средняя абсолютная погрешность является оценкой абсолютной ошибки измерения, т.е. данная величина определяет, насколько предсказание близко к точному результату.

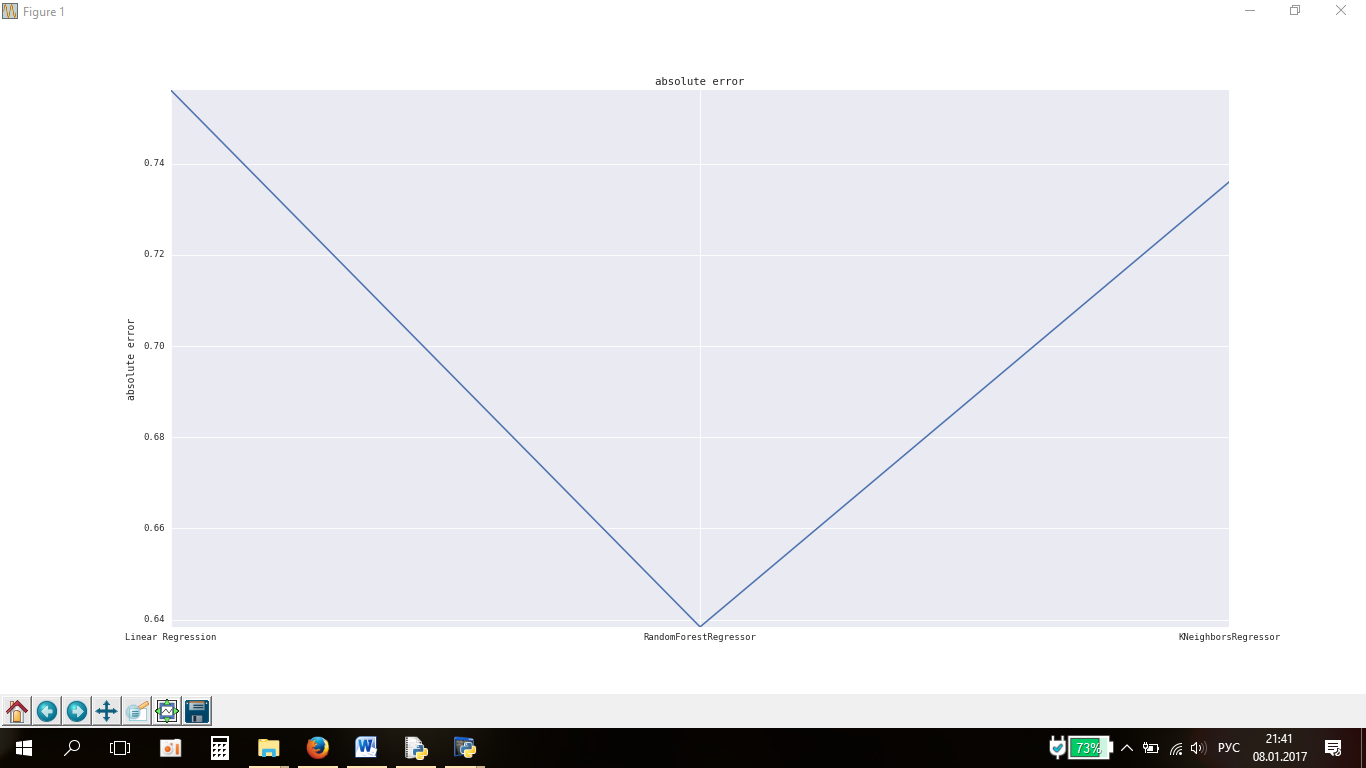
где *xi - x* – абсолютная погрешность.

Чем меньше значения среднеквадратической и средней абсолютной погрешностей, тем обучающая модель является более эффективной.

# Вывод

Целью нашего проекта являлось обучить модель предсказывать рейтинг фильма до его показа, используя его числовые характеристики (количество лайков на Facebook, количество рецензий критиков, бюджет, продолжительность фильма и т.д.). Проанализировав и изменив данные, мы обучили нашу модель. Для более качественного обучения модели мы отобрали 3 наиболее подходящих для нашей задачи алгоритма и проверили их эффективность, используя значения MSE и MAE алгоритмов.

Как видно из графиков, метод RandomForestRegressor обладает минимальными значениями погрешностей, следовательно, является более эффективным для решения нашей задачи.



Однако если сравнивать предсказанные значения с предполагаемым результатом, используя алгоритм RandomForestRegressor, то мы получим довольной большой процент ошибки на обучающей выборке (≈ 35%). Следовательно, мы можем сделать вывод, что для обучения нашей модели недостаточно использовать только числовые признаки объекта. На наш взгляд, для более точного результата следует проанализировать все характеристики в совокупности и, применяя методы по уменьшению размерности, обучать модель с использованием метода RandomForestRegressor.

# Приложение №1. Листинг программы

