Отчет по лабораторной работе № 2

**Анализ многомерных случайных величин**

Выполнено:

Барбаков Илья Олегович,

Группа J4150

Санкт-Петербург

2022

**Оглавление**

[1. Обоснование выбора данных 1](#_Toc118447056)

[2. Постановка задачи 1](#_Toc118447057)

[3. Анализ случайных величин 2](#_Toc118447058)

[4. Многомерная корреляция 5](#_Toc118447063)

[5. Построение регрессионной модели 7](#_Toc118447067)

[6. Построение байесовской сети 21](#_Toc118447095)

[7. Реализация задачи регрессии с помощью БС 23](#_Toc118447100)

[Исходный код 25](#_Toc118447103)

# Обоснование выбора данных

Для задачи многомерного анализа случайных величин выбран датасет, содержащий статистические данные города Бостон. Он имеет 506 записей, 11 предикторов и целевую переменную:

* CRIM - количество преступлений на душу населения
* ZN - процент жилых участков площадью больше 25 тыс. кв. футов
* INDUS - процент площадей под оптовую торговлю
* CHAS - протекает ли река
* NOX - концентрация оксидов азота
* RM - среднее число комнат в здании
* AGE - доля зданий, построенных до 1940 года
* RAD - индекс доступности скоростных магистралей
* TAX - уровень налогов
* PTRATIO - среднее число учащихся на одного преподавателя
* LSTAT - процент граждан с низким уровнем жизни
* MEDV (целевой) - медианная стоимости домов в районе

Задачей работы является построение регрессионной модели, и данный датасет отлично подходит для этого, целевой переменной выступает значение медианной стоимости домов в районе.

# Постановка задачи

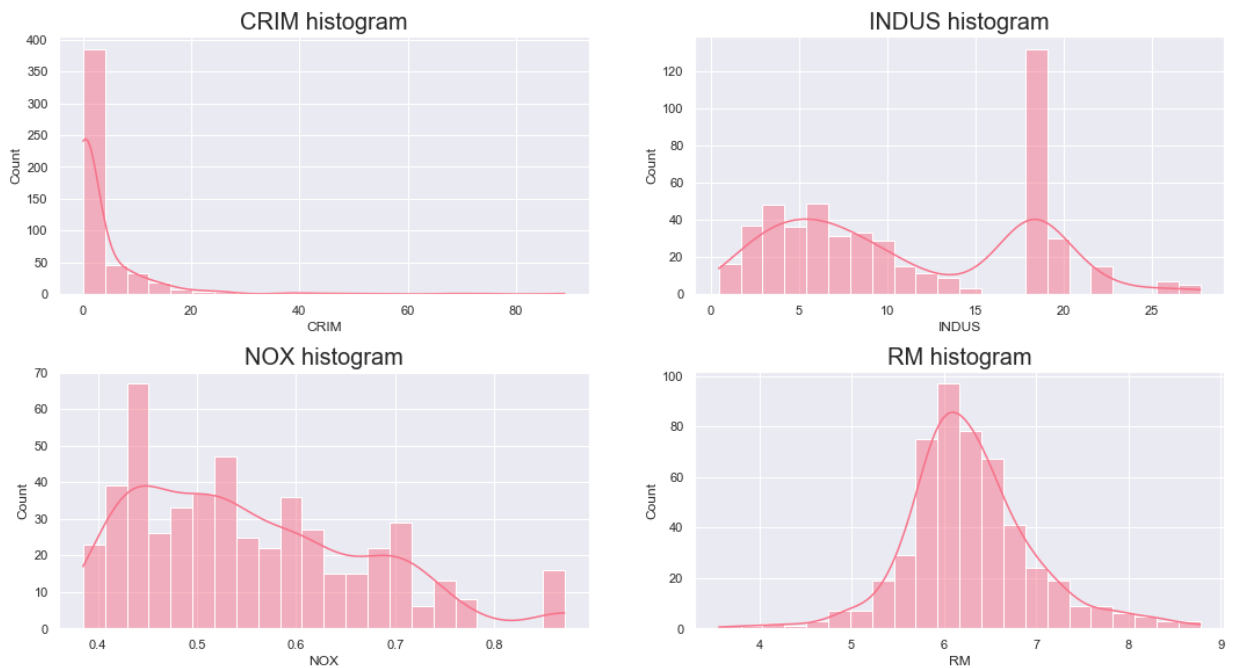
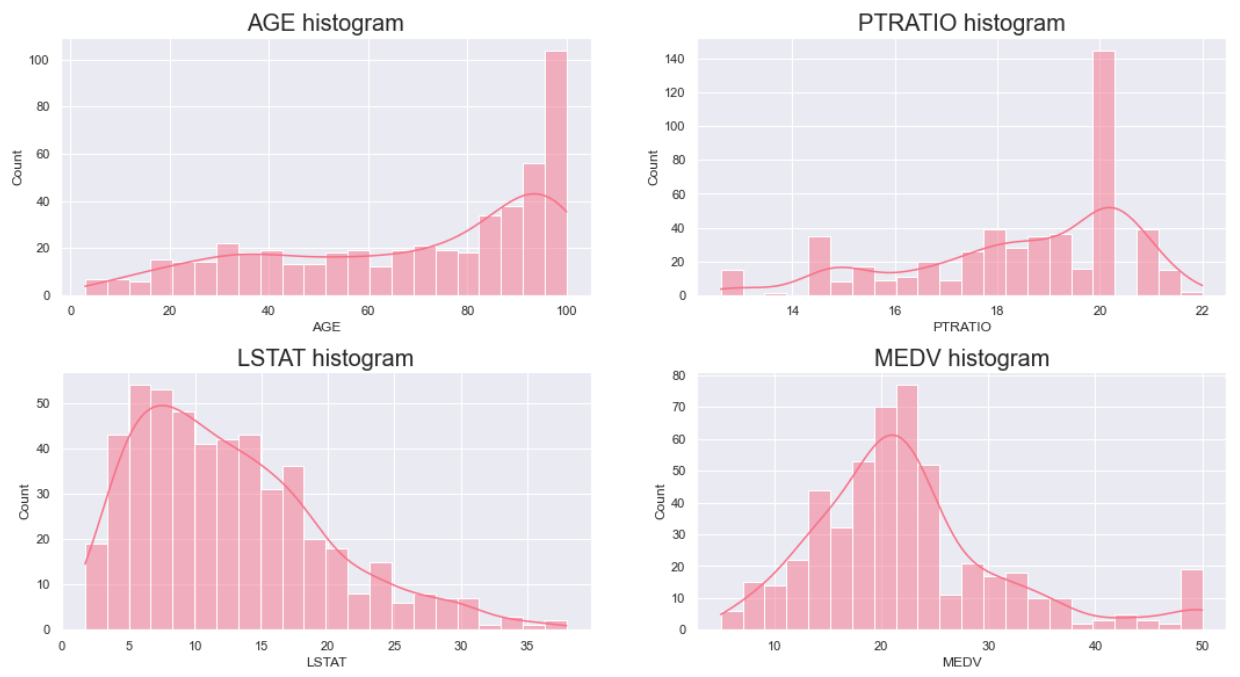
Задачей данной работы является минимизация среднеквадратичной ошибки регрессии прогноза медианной стоимости домов в районе.

# Анализ случайных величин

Первым этапом выступает анализ одномерных случайных величин датасета. Построим гистограммы, количество столбцов которых рассчитаем с помощью формулы m = 1 + 3.32 lg n. Для непрерывных величин также визуализируем ядерное сглаживание, основанное на Гауссовой функции.

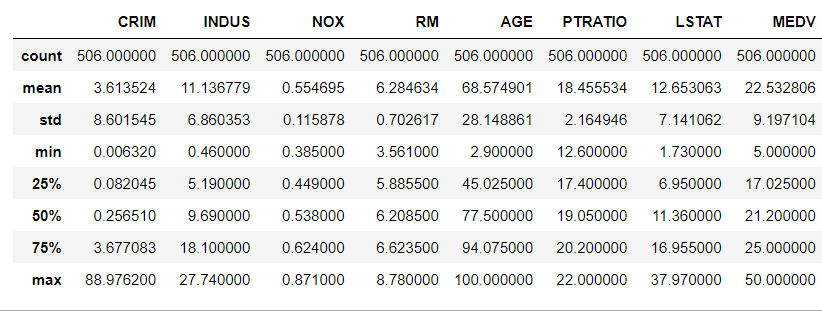


## Рисунок 1 - Гистограммы дискретных распределений



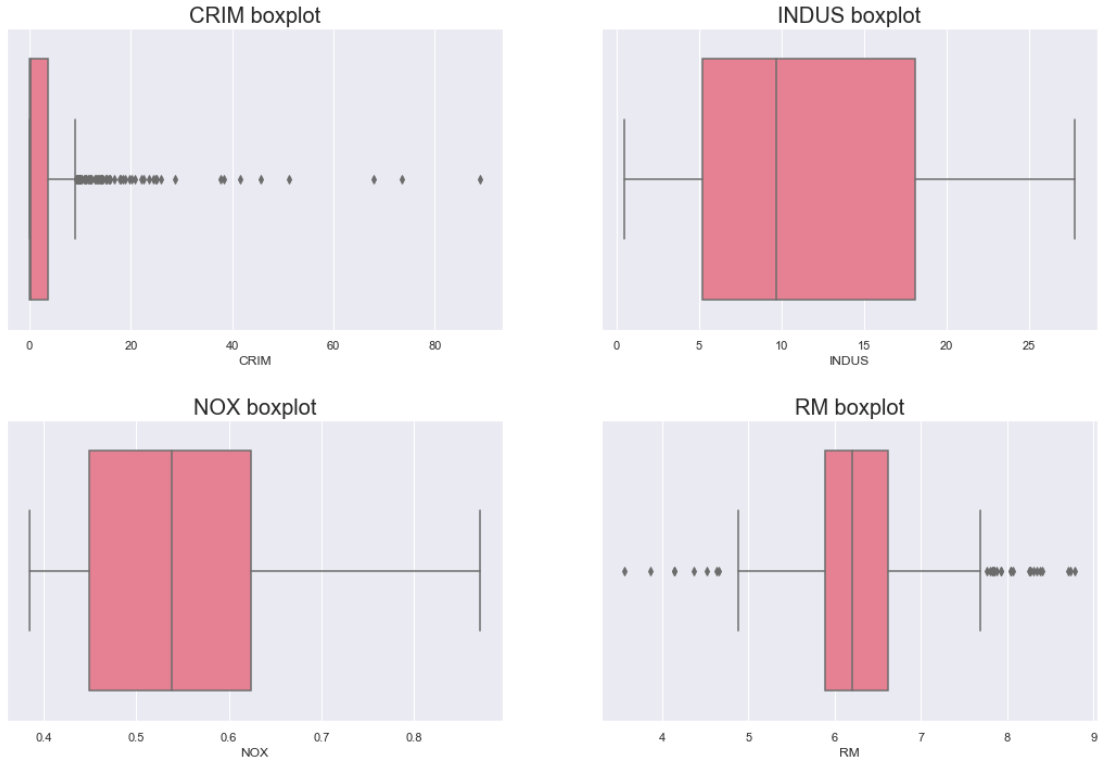
## Рисунок 2 - Гистограммы непрерывных распределений

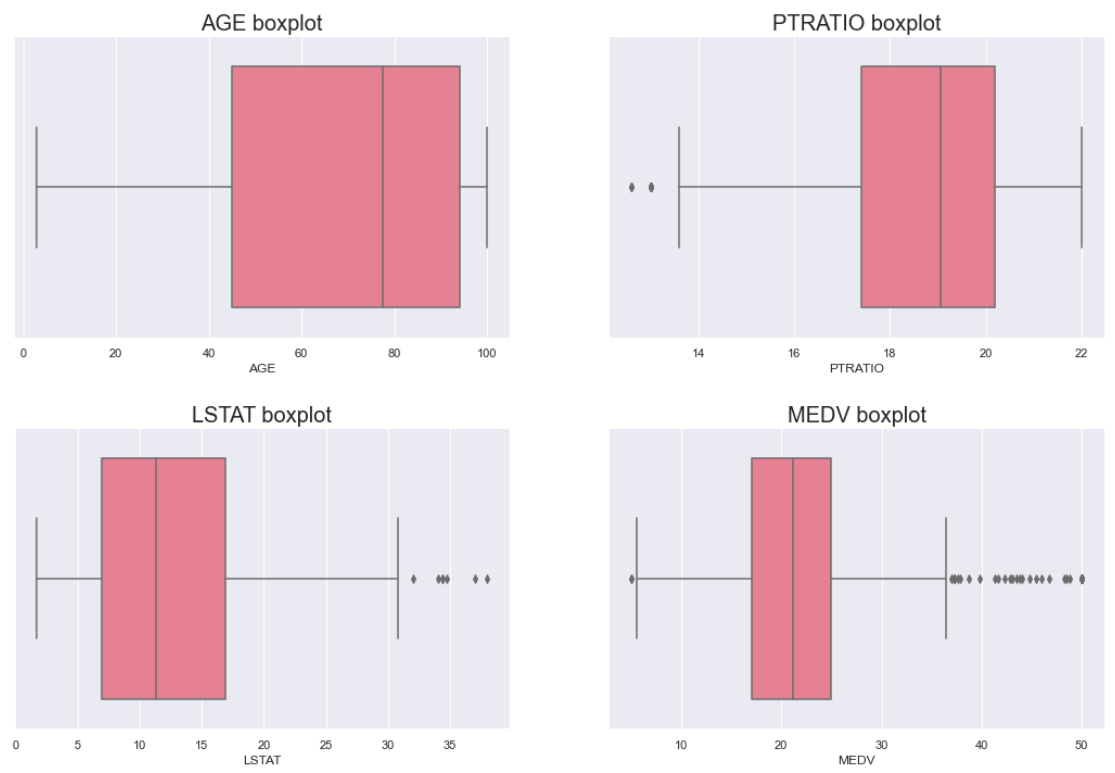
Далее проанализируем порядковую статистику непрерывных величин.



## Рисунок 3 - Порядковая статистика

Визуализируем порядковую статистику непрерывных случайных величин с помощью «ящика с усами».

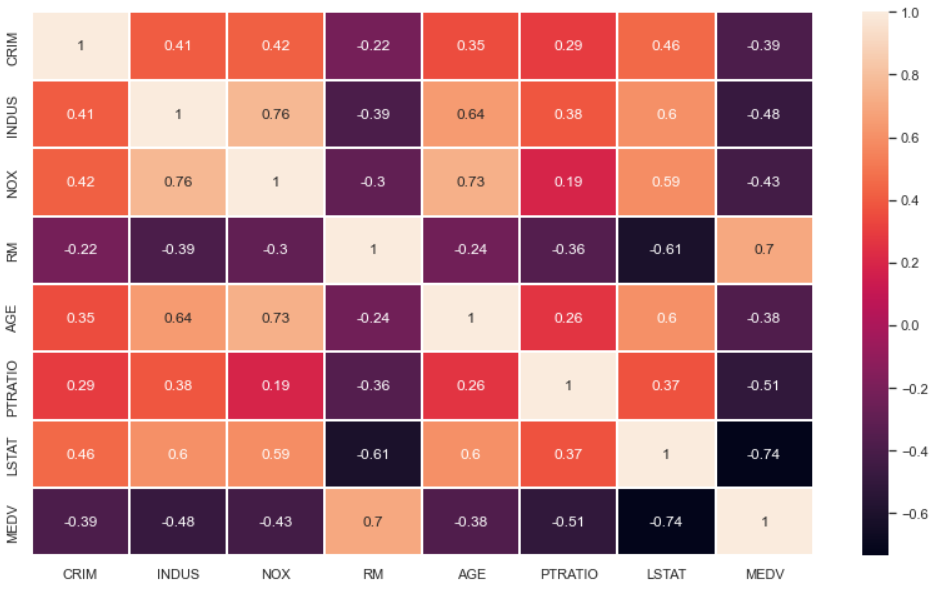




## Рисунок 4 - Визуализация порядковой статистики

# Многомерная корреляция

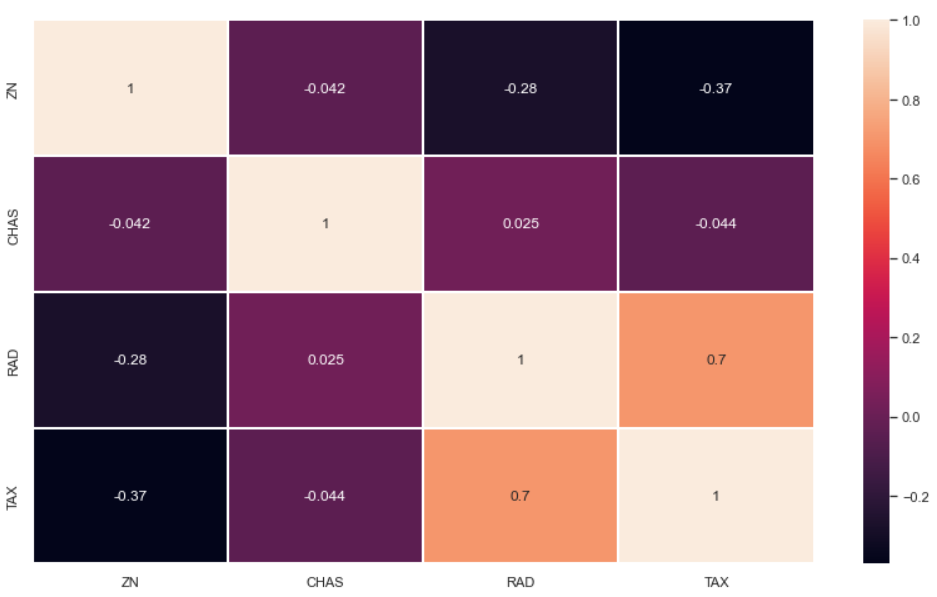
Корреляция – нормированная от -1 до 1 мера линейной зависимости переменных. Чем коэффициент ближе к 0, тем меньше зависимости, а чем он по модулю ближе к 1, тем сильнее зависимы переменные. Суть многомерной корреляции заключается в попарном анализе корреляций признаков датасета. Построим многомерную корреляцию для непрерывных величин.



## Рисунок 5 - Многомерная корреляция для непрерывных СВ

Из графика можно наблюдать, что между предикторами нету сильной линейной зависимости, таким образом отсутствует проблема мультиколлиниарности. На целевую переменную наименьшее линейное влияние имеет показатель доли старых зданий, а наибольшее - процент граждан с низким уровнем жизни, коэффициент равен -0.74, что показывает отрицательную линейную зависимость.

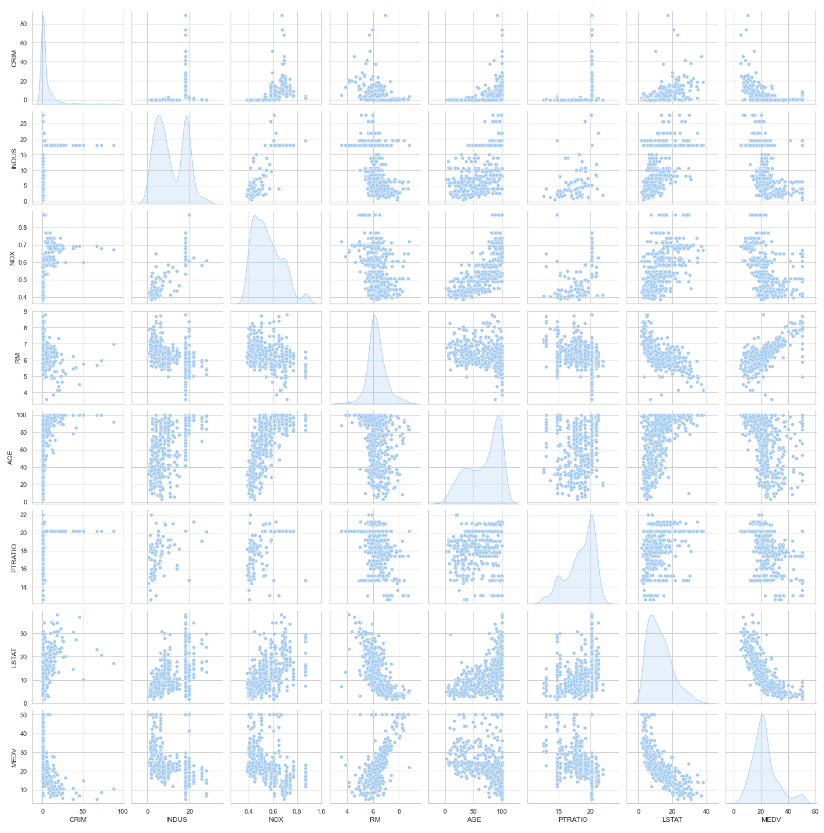
Для дискретных случайных величин существует ранговая корреляция Спирмена.



## Рисунок 6 - Многомерная корреляция для дискретных СВ

Показатели «RAD» и «TAX» достаточно хорошо коррелируют друг с другом.

Далее визуализируем попарные графики зависимостей переменных. В библиотеке seaborn такой функционал реализован с помощью pairplot.

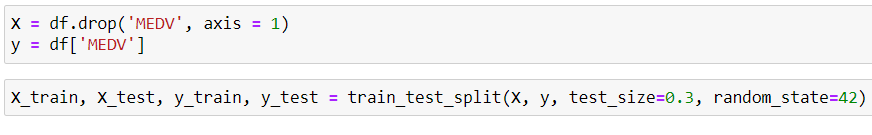


## Рисунок 7 - Графики зависимостей переменных

Используя визуальный анализ, можно сделать вывод, что предикторы «RM» и «LSTAT» имеют хорошую зависимость с целевой переменной.

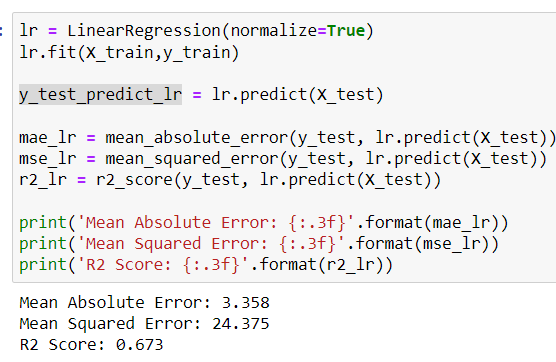
# Построение регрессионной модели

Далее необходимо спрогнозировать медианную стоимость домов в районе с помощью построения регрессионной модели. Первым этапом подготавливаем данные: выделяем целевую переменную и разделяем датасет на тренировочную и тестовую.



## Рисунок 8 - Подготовка данных для обучения

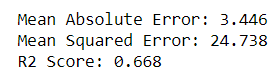
В качестве baseline модели используем обычную линейную регрессию. В качестве метрик сразу будем выводить значение средней абсолютной ошибки, среднеквадратичной и коэффициент детерминации.



## Рисунок 9 - Оценки линейной регрессии

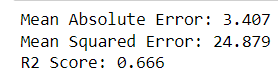
Показатели достаточно низкие, поэтому ищем более оптимальную модель регрессии.

Анализируя корреляции между предикторами, было замечена небольшая линейная зависимость. Хотя это недостаточно для утверждения мультиколлинеарности, на всякий случай проведем регуляризацию регрессионной модели и посмотрим результаты. Первым способом выступает Lasso-регрессия, которая использует L1-регуляризацию. Сначала реализуем небольшую регуляризацию с коэффициентом альфа = 0.15.



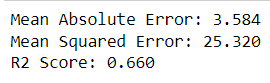
## Рисунок 10 - Оценки Lasso-регрессии

Оценки стали хуже, возможно из-за неправильного подобранного коэффициента альфа. С помощью кросс-валидацинной проверки получаем параметр равный 0.06 и применяя его прогоняем еще раз Lasso-регрессию.



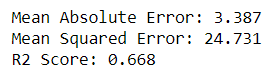
## Рисунок 11 - Оценки Lasso-регрессии с подобранными параметрами

Оценки Lasso-регрессии стали чуть лучше, но все равно хуже, чем у линейной регрессии. Мы можем сделать вывод, что L1-регуляризация не подходит для данной модели и не несет смысла. Далее применим L2-регуляризацию с помощью Ridge-регрессии. В качестве параметра альфа принимаем 100.



## Рисунок 12 - Оценки Ridge-регрессии

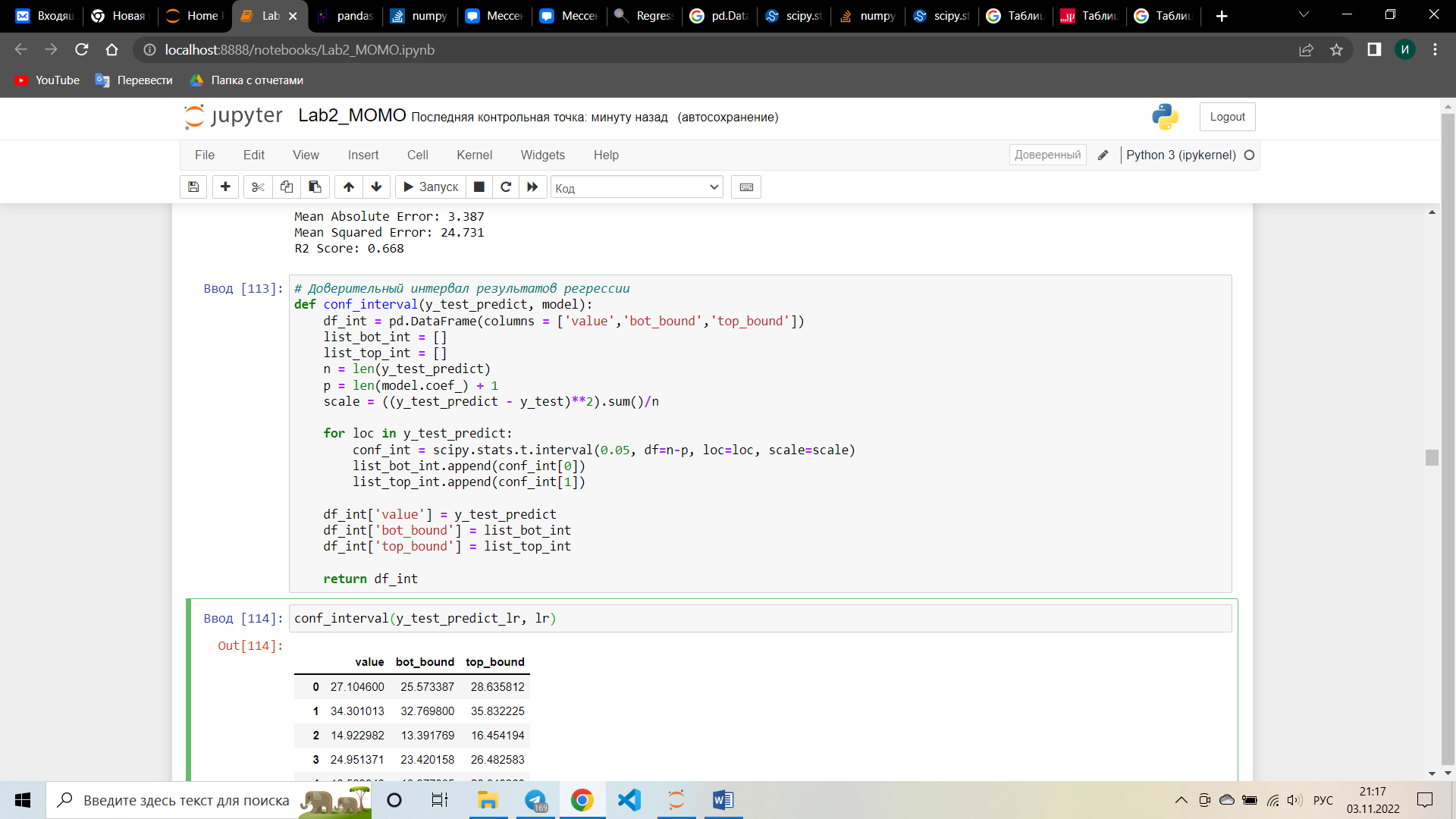
Также, как и для Lasso-регрессии, подберем параметр альфа с помощью кросс-валидации, он равен 3.57.



## Рисунок 13 - Оценки Ridge-регрессии с подобранными параметрами

Основные оценки регрессионной модели стали чуть лучше, но не сильно, поэтому можно сделать вывод, что в данных отсутствует мульколлинеарность и регуляризация не принесла хороших результатов.

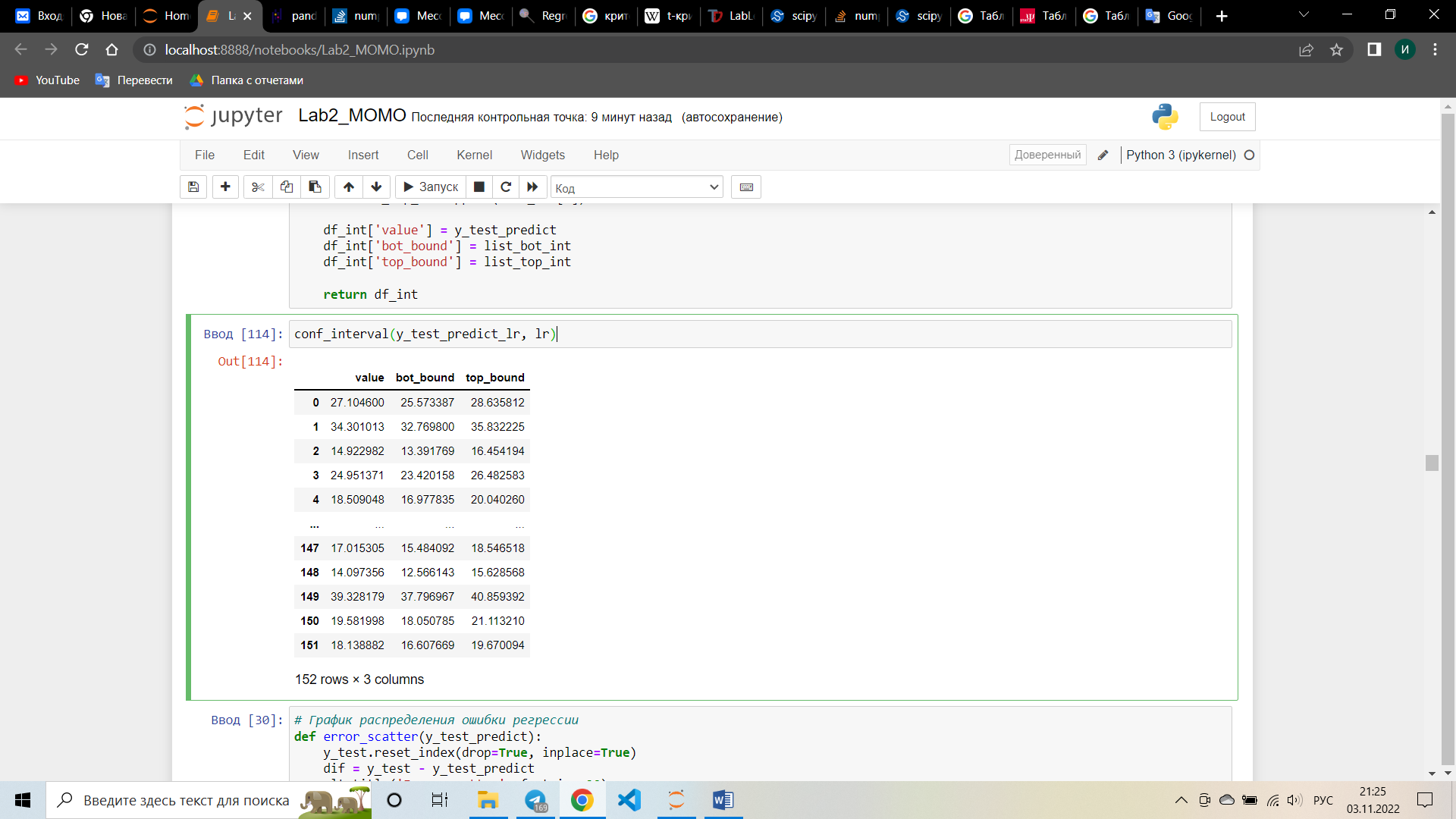
Для линейной регрессии построим доверительные интервалы результатов с помощью t критерия Стьюдента.



## Рисунок 14 - Доверительный интервал результатов регрессии

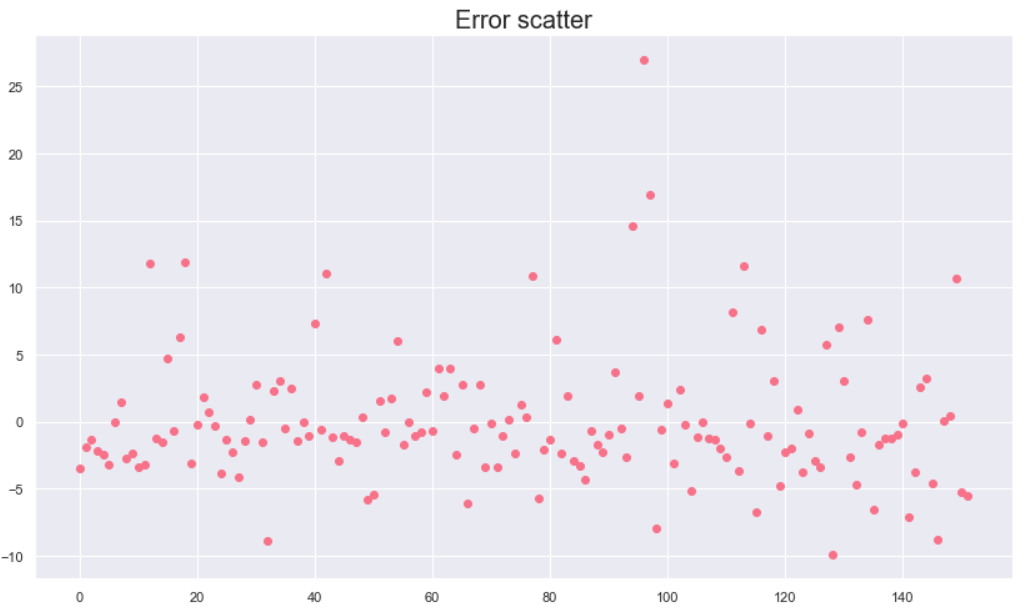
Данный алгоритм реализуется с помощью scipy.stats.t. В качестве параметров он принимает 1-a – 1 - степень уверенности, n-p – число степеней свободы, loc – значение предикта и scale – среднеквадратичная ошибка регрессии.

Написанная функция возвращает датафрейм с предсказанными значениями, а также нижними и верхними границами доверительного интервала.



## Рисунок 15 - Доверительный интервал результатов линейной регрессии

Далее реализуем ещё несколько различных регрессионных моделей и проведем сравнение их качества. Для начала оценим качество линейной регрессии. Построим график распределения ошибки регрессии, в зависимости от порядкового номера проверочного датасета.



## Рисунок 16 - График распределения ошибки линейной регрессии

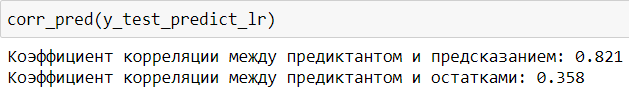
Используя визуальный анализ графика выше, не получается найти какие-то признаки непостоянства дисперсии и тем более какие-то зависимости. Теперь строим график сравнения значений предсказания и таргета регрессии.



## Рисунок 17 - График сравнения значений предсказания и таргета линейной регрессии

Из графика видно, что существуют большие ошибки линейной регрессии. Также одно из значений отрицательно, что априори невозможно.

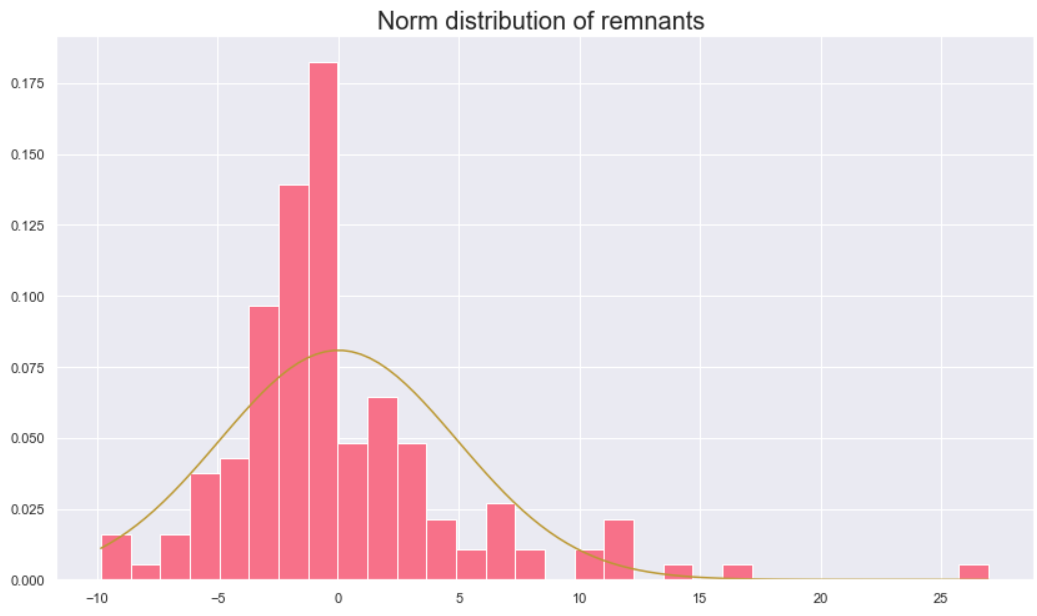
Одной из оценок качества регрессионной модели выступает коэффициенты корреляции между предиктантом и предсказанием или остатками.



## Рисунок 18 – Корреляция предиканта с предсказанием и остатками

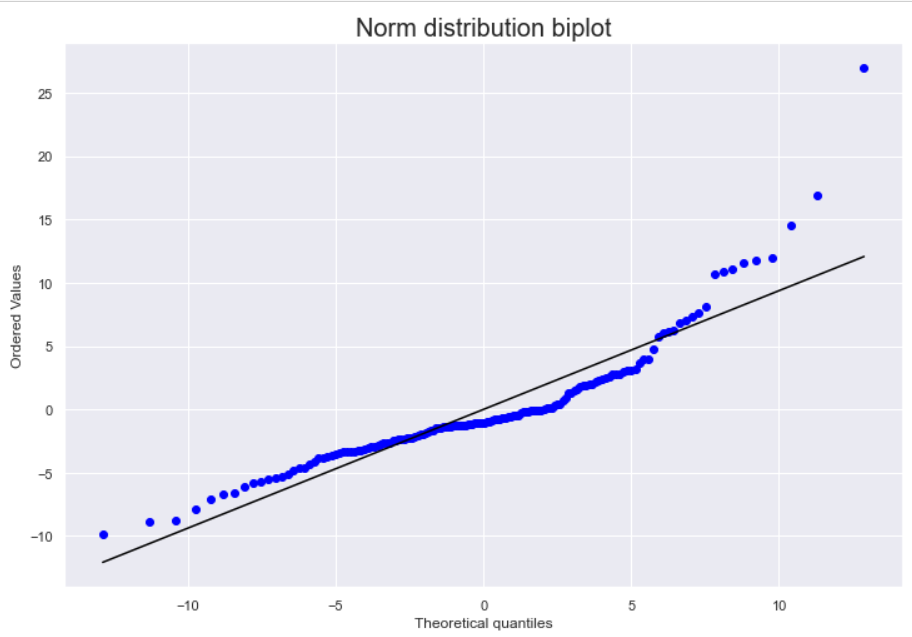
Чем выше коэффициент корреляции между предиктантом и предсказанием, тем лучше подобрана модель регрессии. Для корреляции между предиктантом и остатками наоборот.

Также критерием качества подобранной модели регрессии выступает проверка гипотезы нормальности распределения остатков с математическим ожиданием равным 0. Построим гистограмму остатков и график нормального распределения.



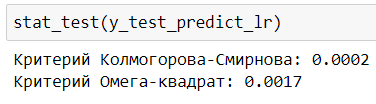
## Рисунок 19 - График нормального распределения остатков линейной регрессии

Построим квантильный биплот, основываясь на квантилях остатков линейной регрессии и теоретического нормального распределения.



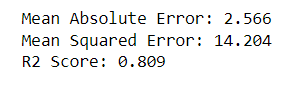
## Рисунок 20 - Квантильный биплот остатков линейной регрессии и нормального распределения

Визуальный анализ показывает, что данные не очень хорошо сходятся, необходимо проверить гипотезу нормальности распределения остатков регрессии с помощью статистических тестов. Воспользуемся критерием Колмогорова-Смирнова и Омега-квадрат.



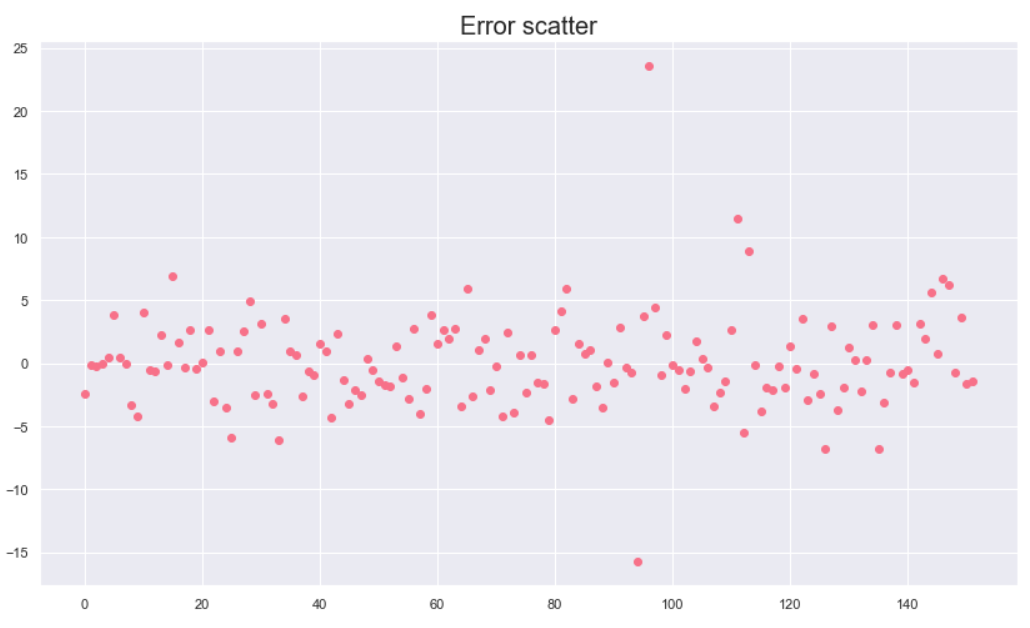
## Рисунок 21 - Статистические тесты для остатков линейной регрессии

Теперь попробуем реализовать полиномиальную модель регрессии, используя полином второй степени. Тогда оценки регрессии будут равны:



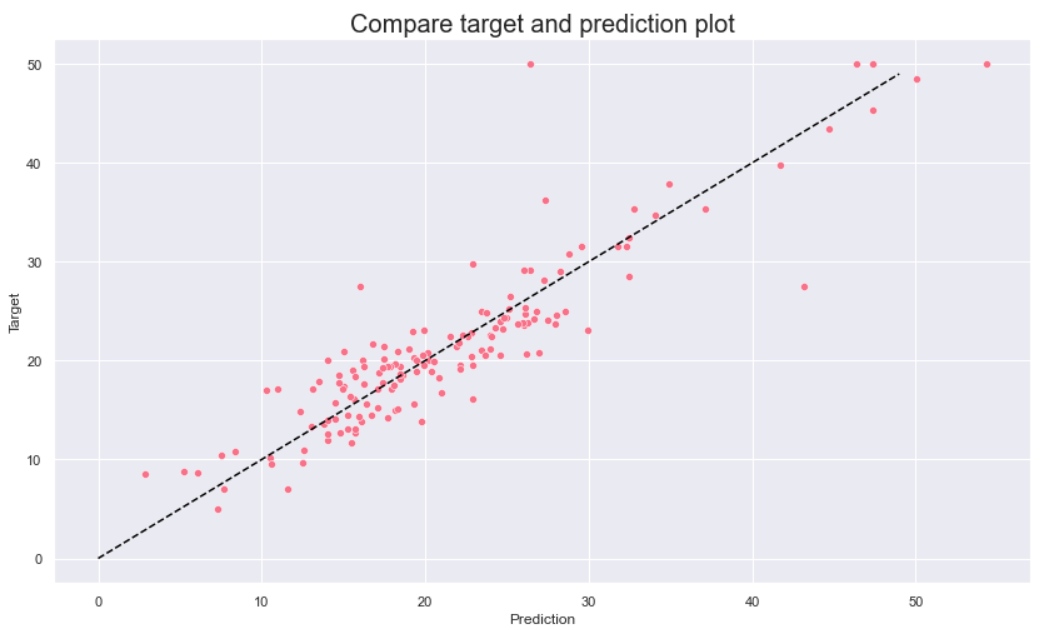
## Рисунок 22 - Оценки полиномиальной регрессии

Оценки регрессии стали намного лучше, проведем еще несколько сравнительных визуальных характеристик модели.



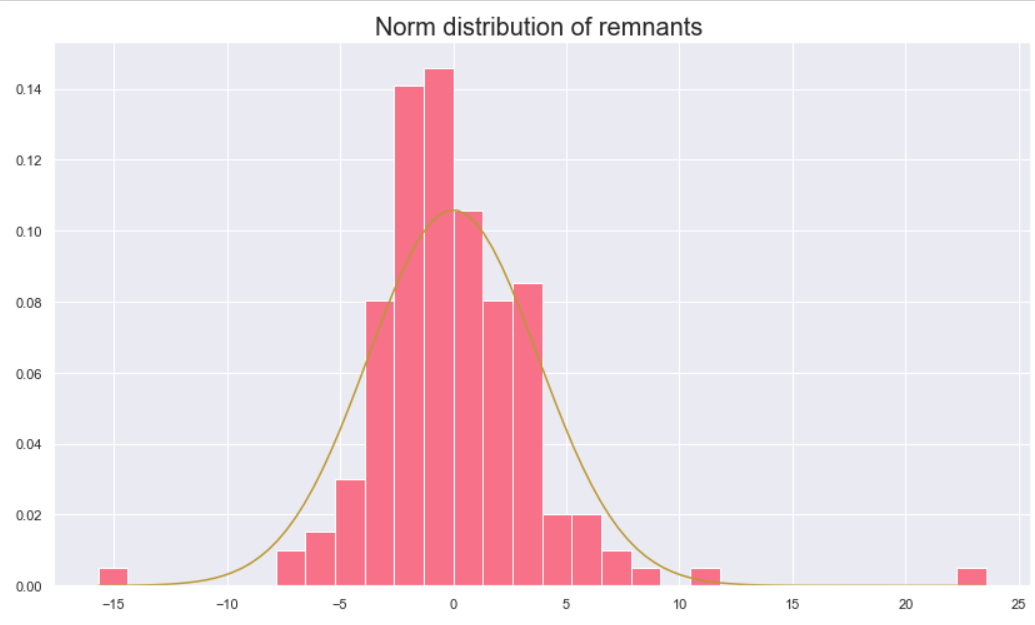
## Рисунок 23 - График распределения ошибки полиномиальной регрессии

Как и для линейной регрессии, из графика нельзя выявить явные закономерности распределния остатков. Построим график сравнения значений предсказания и таргета регрессии.



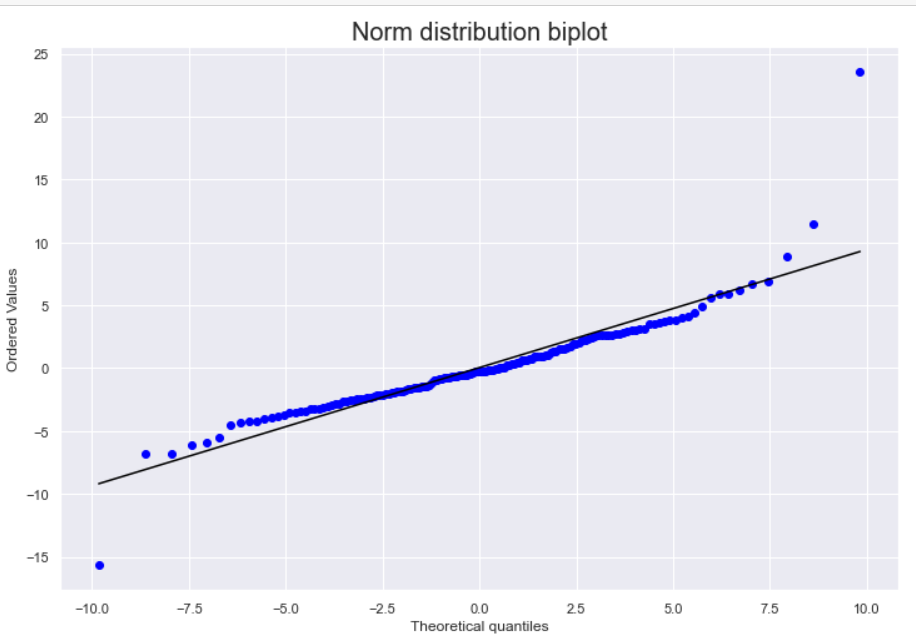
## Рисунок 24 - График сравнения значений предсказания и таргета полиномиальной регрессии

Далее проверяем остатки полиномиальной регрессии на нормальность распределения.



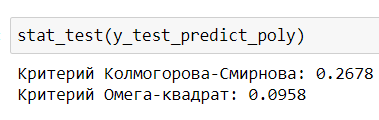
## Рисунок 25 - График нормального распределения остатков полиномиальной регрессии

По первичному анализу графика, можно сделать вывод, что данные могут распределяться по нормальному закону распределения. Построим квантильный биплот.



## Рисунок 26 - Квантильный биплот остатков полиномиальной регрессии

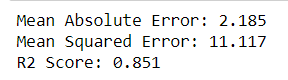
Все точки значений квантилей теоретического нормального распределения лежат практически на главной диагонали, кроме хвостов. Чтобы окончательно принять гипотезу о нормальности распределения остатков полиномиальной регрессии, проведем статистические тесты.



## Рисунок 27 - Статистические тесты для остатков полиномиальной регрессии

Значения критериев больше 0.05, таким образом, мы принимаем нулевую гипотезу, о том, что остатки полиномиальной регрессии распределены по нормальному закону.

Следующей регрессионной моделью, которую мы будем реализовывать – случайный лес. Это алгоритм машинного обучения, суть которого заключается в использовании ансамбля деревьев решений. В качестве гиперпараметра – количества решающий деревьев примем 50.



## Рисунок 28 - Оценки случайного леса

Случайный лес уже намного лучше предсказывает целевую переменную. Построим график распределения ошибки.



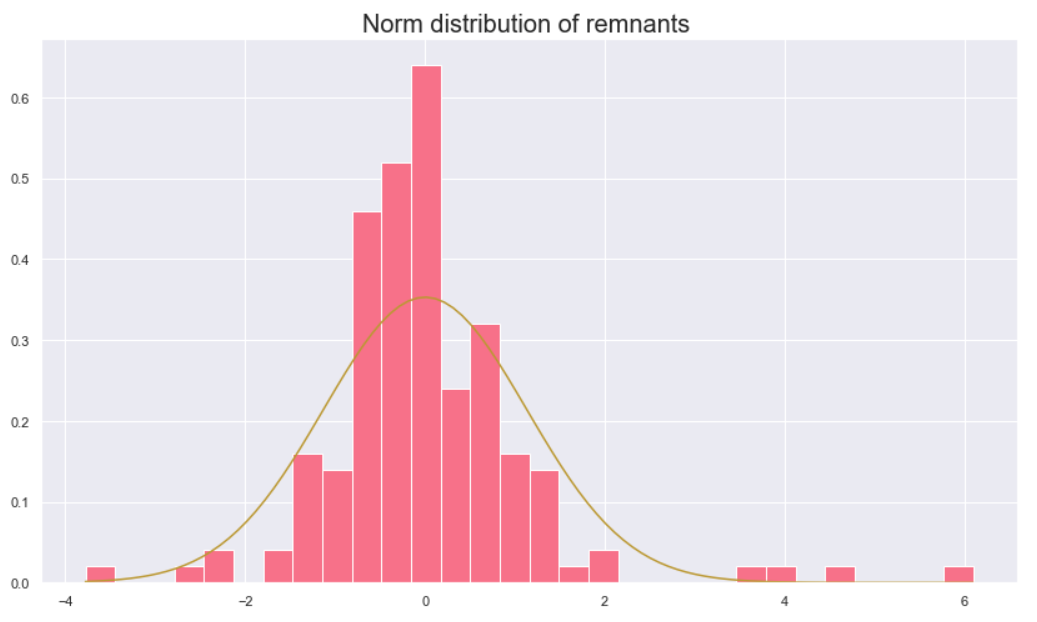
## Рисунок 29 - График распределения ошибки случайного леса

Точки ошибки регрессии уже лежат ближе к нулю. Визуализируем сравнение предсказания и таргета.

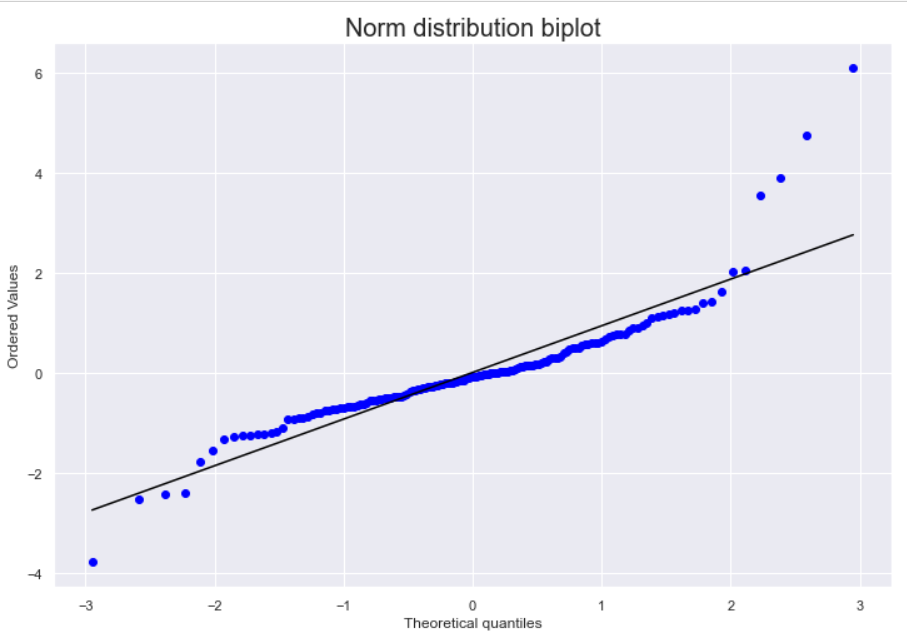


## Рисунок 30 - График сравнения значений предсказания и таргета случайного леса

Значения предсказанных значений целевой переменной практически совпадают с таргетом. Далее проверим остатки регрессии на нормальность.

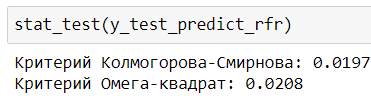


## Рисунок 31 - График нормального распределения остатков случайного леса



## Рисунок 32 - Квантильный биплот остатков случайного леса

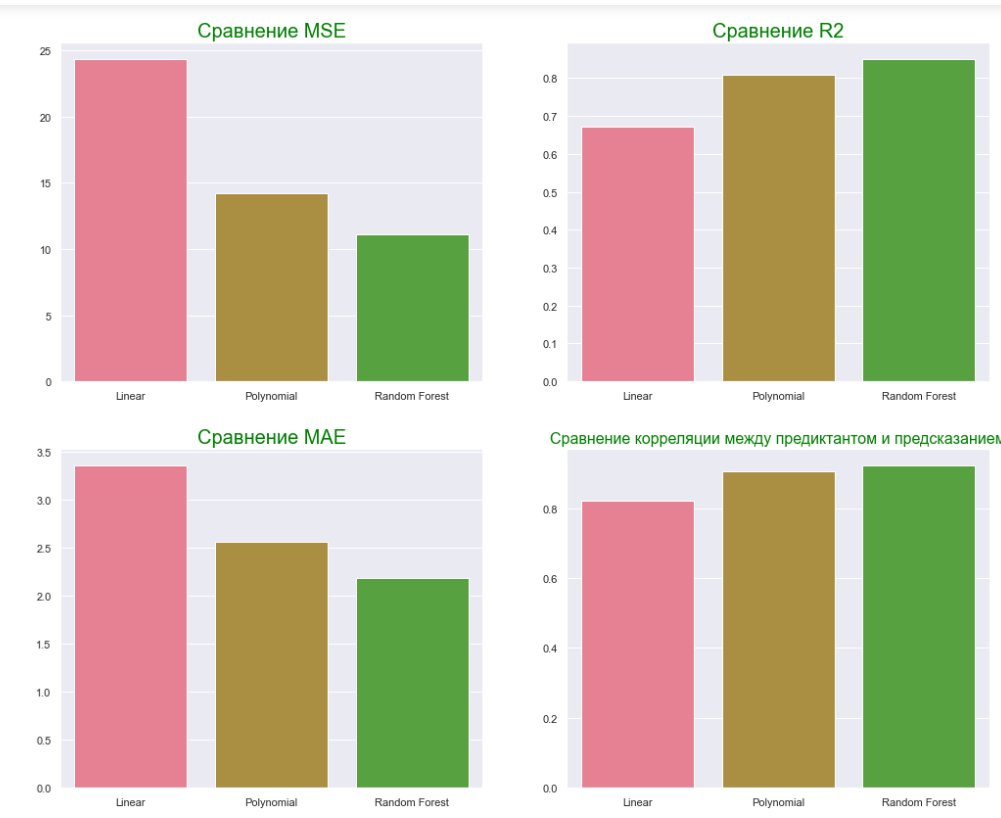
Теперь проведем статистические тесты.



## Рисунок 33 - Статистические тесты для остатков случайного леса

Хоть и показатели статистических тестов не сильно маленькие, они не превышают 0.05, что говорит о том, что мы отвергаем нулевую гипотезу о нормальности распределения остатков случайного леса.

Далее сравним основные оценки реализованных моделей регрессии.

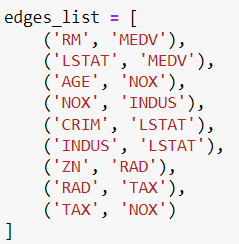


## Рисунок 34 - Сравнение оценок регрессионных моделей

Хоть и остатки случайного леса не подвергаются закону распределения, данная регрессионная модель наилучшая из реализованных.

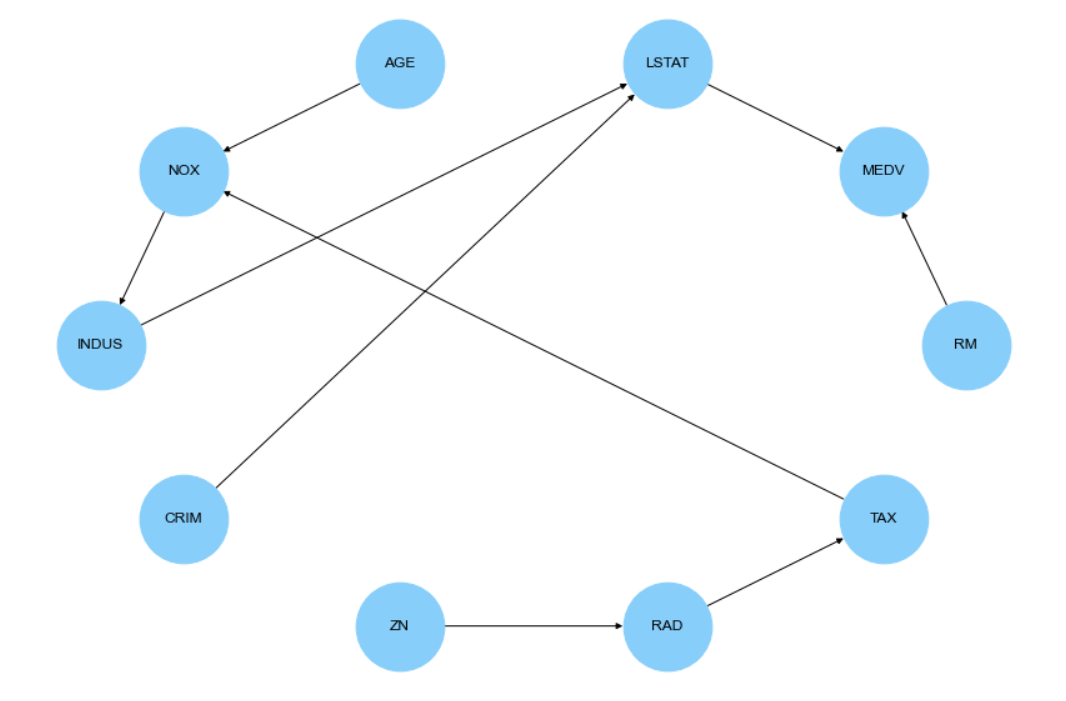
# Построение байесовской сети

Первым этапом выступает определение структуры байесовской сети. Сделать это можно несколькими способами, первым из которых является определение структуры на основе многомерного анализа. Построив таблицы многомерной корреляции непрерывных и дискретных величин, мы можем определить условные зависимости предикторов между собой.



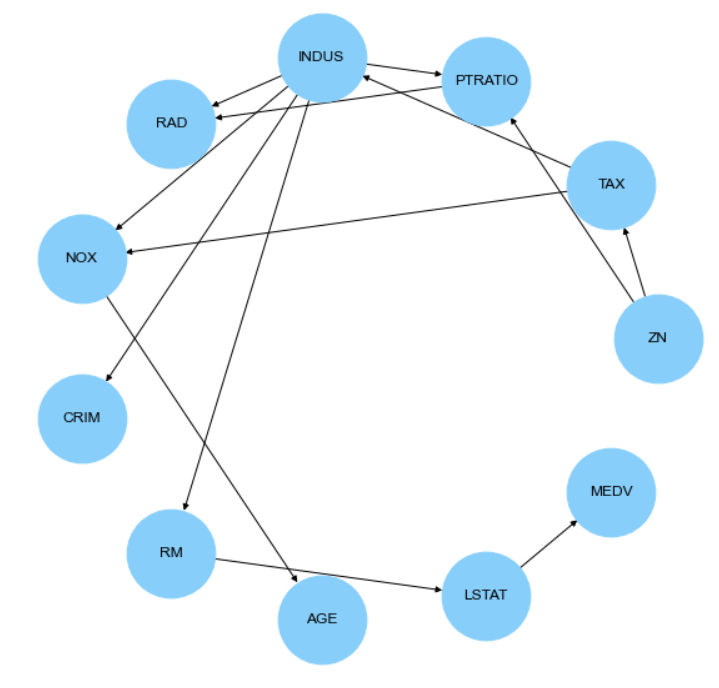
## Рисунок 35 - Определение условных зависимостей

Таким образом, граф байесовской сети будет выглядеть следующим образом:



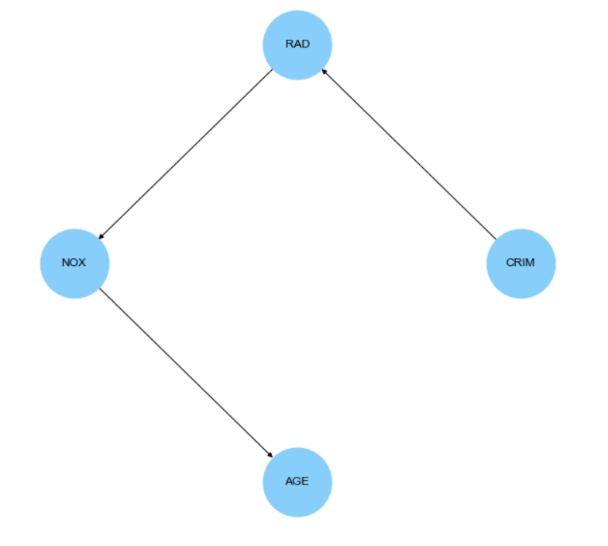
## Рисунок 36 - Вручную определённая структура байесовской сети

Также обучать структуру байесовской сети можно с помощью различных мер качества сети. Одним из таких является алгоритм Hill-Climbing, суть которого заключается в жадном поиске в пространстве ориентированных графов. В качестве score-функции будем использовать K2Score и BicScore. Тогда для K2Score структура байесовской сети будеть выглядеть следующим образом:



## Рисунок 37 - Структура сети K2Score

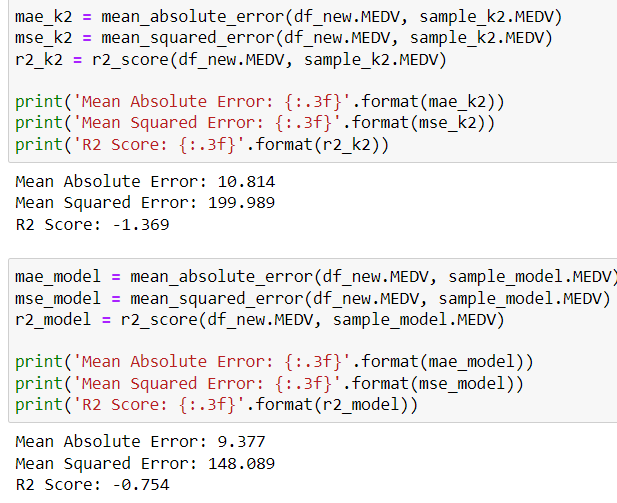
Для BicScore:



## Рисунок 38 - Структура сети BicScore

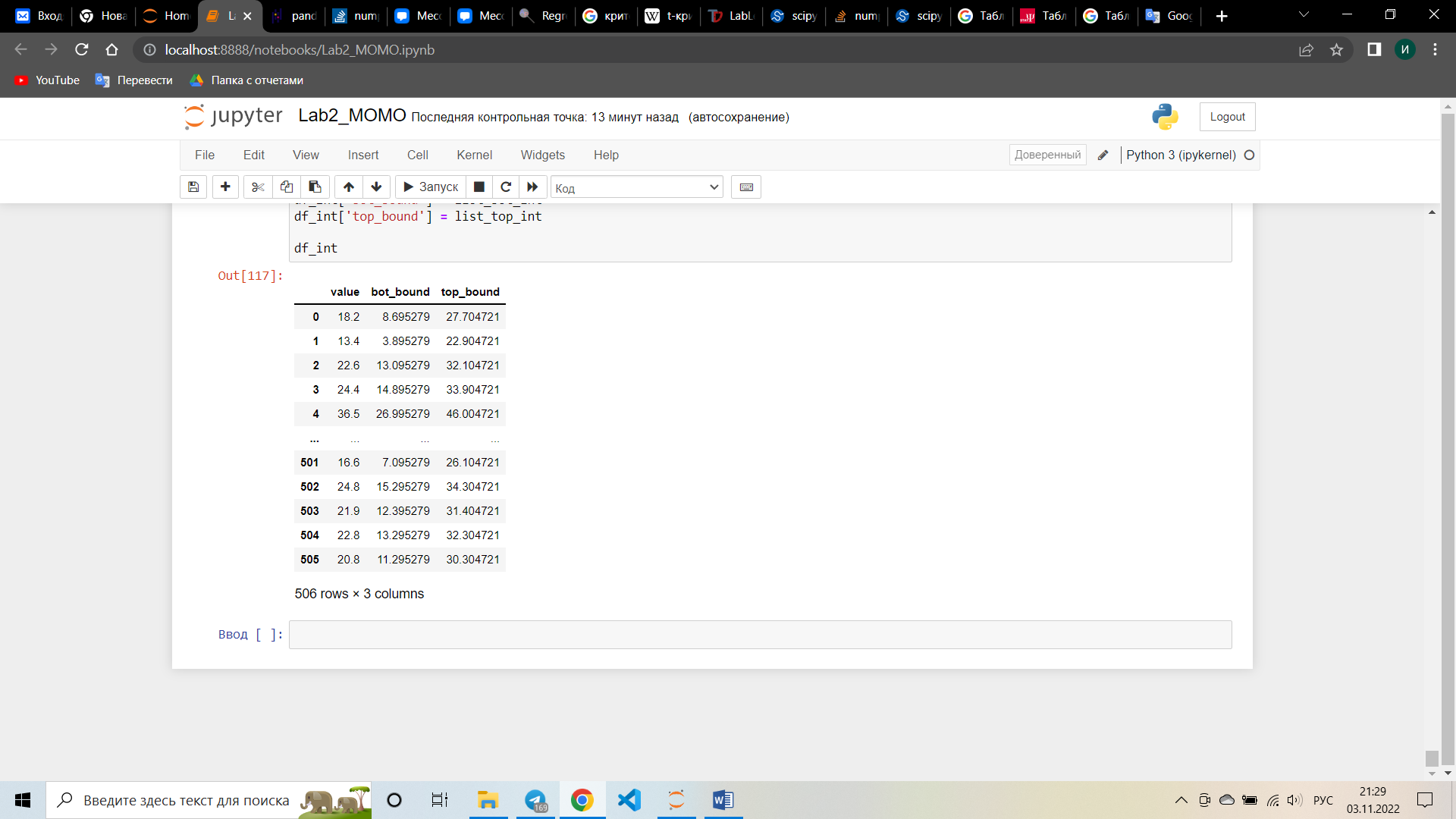
# Реализация задачи регрессии с помощью БС

Для того, чтобы решить задачу регрессии, используя байесовские сети, необходимо с уже смоделированной структурой сети, произвести сэмплирование. Из сэмплированной выборки и таргета сравниваем целевую переменную.



## Рисунок 39 - Оценки регрессии байесовской сети

Построим доверительные интервалы результатов регрессии, полученные с помощью байесовской сети также с помощью t критерия Стьюдента:



## Рисунок 40 – Доверительные интервалы для регрессии байесовской сети

Лучшие результаты показывает модель со структурой, которая была определена вручную. Оценки качества моделей плохи, поэтому для поставленной задачи лучше всего использовать модель случайного леса.

# Исходный код

Ссылка на репозиторий:

<https://github.com/barbak11/MOMO/tree/main/Lab2_multivariate%20random%20variable>