Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

**Высшая школа интеллектуальных систем и суперкомпьютерных технологий**

**КУРСОВАЯ РАБОТА**

**Регрессия в машинном обучении**

по дисциплине «Вычислительная математика»

|  |  |  |
| --- | --- | --- |
| Выполнил  студент гр. 3530903/00001 |  | С.А.Логачева |
| Руководитель  доцент, д.т.н. |  | Е.Г. Хитров |

[26.05.2022]

Санкт-Петербург

[2022]

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

**Высшая школа интеллектуальных систем и суперкомпьютерных технологий**

**ЗАДАНИЕ**

**НА ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ**

студенту [3530903/00001] [Логачева София Александровна]

**1. Тема проекта (работы):**   
Регрессия в машинном обучении

**2. Срок сдачи студентом законченного проекта (работы)** 26.05.2022

**3. Исходные данные к проекту (работе**):   
Консультации с руководителем проекта, материалы сети «Интернет», учебно-методическая литература.

**4. Содержание пояснительной записки** (перечень подлежащих разработке вопросов): [введение, анализ предметной области(обучение, задачи машинного обучения), основная часть (определение целей, сбор и подготовка данных, разработка модели, тестирование, создание модели на Python, создание модели в Maple), Вывод]

**5. Перечень графического материала** (с указанием обязательных чертежей и плакатов):

**6. Консультанты**

**7. Дата получения задания:** 02.02.2022

|  |  |  |  |
| --- | --- | --- | --- |
| Руководитель |  |  | Е.Г. Хитров |
|  | (подпись) |  | (инициалы, фамилия) |
| Задание принял к исполнению |  |  | С.А.Логачева |
|  | (подпись) |  | (инициалы, фамилия) |
| 02.02.2022 |  |  |  |
| (дата) |  |  |  |

Оглавление

[Введение 4](#_Toc104319840)

[1. Анализ предметной области 5](#_Toc104319841)

[1.1. Обучение 5](#_Toc104319842)

[1.2. Задачи машинного обучения 6](#_Toc104319843)

[2. Основная часть 10](#_Toc104319844)

[2.1. Определение целей 10](#_Toc104319845)

[2.2. Сбор и подготовка данных 10](#_Toc104319846)

[2.3. Разработка модели 10](#_Toc104319847)

[2.4. Тестирование 11](#_Toc104319848)

[2.5. Создание модели на языке программирования Python 11](#_Toc104319849)

[2.6. Создание модели в Maple 13](#_Toc104319850)

[3. Вывод 16](#_Toc104319851)

# Введение

На данный момент известно такое понятие, как искусственный интеллект. Искусственный интеллект — это свойства интеллектуальных систем выполнять творчески поставленные задания. Машинное обучение является подразделением искусственного интеллекта. Машинное обучение концентрируется на разработке таких компьютерных программ и алгоритмов, которые сами учатся расти и адаптироваться при подаче новых данных. Этот процесс не похож на процесс интеллектуального анализа данных. Методы машинного обучения впервые упоминаются в 1950 году американскими инженерами А. Сэмюэлем и Дж. Вейценбаумом. Данные упоминания продолжались вплоть до 2010 года. В начале XXI века количество пользователей сети «Интернет» стало в разы больше, что привело к увеличению объема доступной информации. Такие данные содержат информацию не только о конкретных пользователях, но и о, например, предприятиях. Таким образом, появляется, так называемая выборка данных, на основе которых можно анализировать, например, ситуацию на рынке. Приложения, использующие машинное обучение, могу применяться, например, для определения, какой алгоритм является лучшим для решения конкретной задачи. Эти данные затем используются для интеллектуального анализа различными способами [1], например, такими системами рекомендаций, как похожие продукты на eBay, персонализированным контентом на страницах google plus, видеорекламами на сайтах, например, YouTube, и последними, но не наименьшими предложениями друзей на Facebook. Также используется для интеллектуального поиска в поисковых системах Google и Bing.

Цель работы: разработать программу, которая будет осуществлять регрессию

Объектом линейная регрессия путем аппроксимации, а предметом – зависимость оценки от количества часов, которые учится студент.

# Анализ предметной области

## Обучение

#### Обучение с учителем

Задачи машинного обучения можно разделить на два класса: обучение с учителем и обучение без учителя.

В обучении с учителем во время подготовки модели машинного обучения или, как обычно говорят, во время тренировки модели, мы заранее знаем правильные ответы, по которым можно понять, как решить задачу, и проверить, хорошо ли наша модель обучается. То есть у модели есть «учитель», который говорит ей, правильно она решает задачу или ошибается. Обучение с учителем по-другому называют контролируемым обучением (supervised learning)[2].

Представим, что необходимо научить маленького ребенка отличать кошек от собак. Можно показать ребенку много изображений кошек и собак и указывать, кто есть кто. Теперь, когда ребенок увидит кошку на улице, он скорее всего не перепутает ее с собакой и наоборот. Сложности могут возникнуть, если на улице попадется ну очень странная собакоподобная кошка или очень кошкообразная собака. Если картинки с животными были не очень наглядными, это тоже повлияет на результат. Но мы, как учитель, быстро сможем его поправить.

То же и с компьютером. Картинки кошек и собак — это данные и наш учитель, потому что заранее указано, где и кто. На них программа опирается, когда нужно определить на незнакомой картинке вид животного. Сложности с компьютером такие же: могут быть необычные данные или плохо размеченные обучающие данные. В общем, или кошки на улице странные, или кошки на картинках не очень-то на кошек похожи, или подписи перепутаны. Тогда можно откорректировать данные, добавить в обучающий набор необычные случаи и снова тренировать нашу модель.

#### Обучение без учителя

В обучении без учителя (unsupervised learning) нет правильных ответов, на которых машина могла бы сориентироваться. Компьютерная программа сама ищет зависимости между данными, но с условием, что она не может проверить правильность своего ответа[3].

Когда маленький ребенок видит большую компанию собак и кошек, он не знает, кто такие эти ваши «собаки» и «кошки». Он долго на них смотрит и понимает, что одни существа гавкают, а другие — мяукают. Одни смешно скачут, а другие плавно и мягко ходят. И ребенок начинает понимать, что вот эти гавкающие — это одно, а мяукающие — другое. Ему самому приходится искать закономерности. Так что если какая-то кошка вдруг начнет смешно скакать, то ребенок может решить, что она из вида «гавкающих».

Компьютер тоже ничего не знает о кошках и собаках. Он смотрит и по внешним признакам — форме морды и тела, окрасу, позам и т.д. — учится отличать одних животных от других.

## Задачи машинного обучения

Задачи обучения с учителем делятся на два типа: регрессия — предсказание значения на числовой прямой, и классификация — предсказание категории объекта.

#### Задача регрессии

Практическая суть задачи регрессии в том, что необходимо предсказать какое-то конкретное числовое значение. Причем предсказывать можно все, что угодно: температуру воздуха, цену или спрос на продукты и т.д.

Представим, есть набор отдельных объектов Х, у каждого объекта есть вектор «действительно-значных признаков» х.

Действительно-значные признаки — это признаки, значения которых могут быть только действительными числами (к ним относятся все положительные, отрицательные числа и ноль). В качестве ответа мы также получим действительно-значную переменную t.

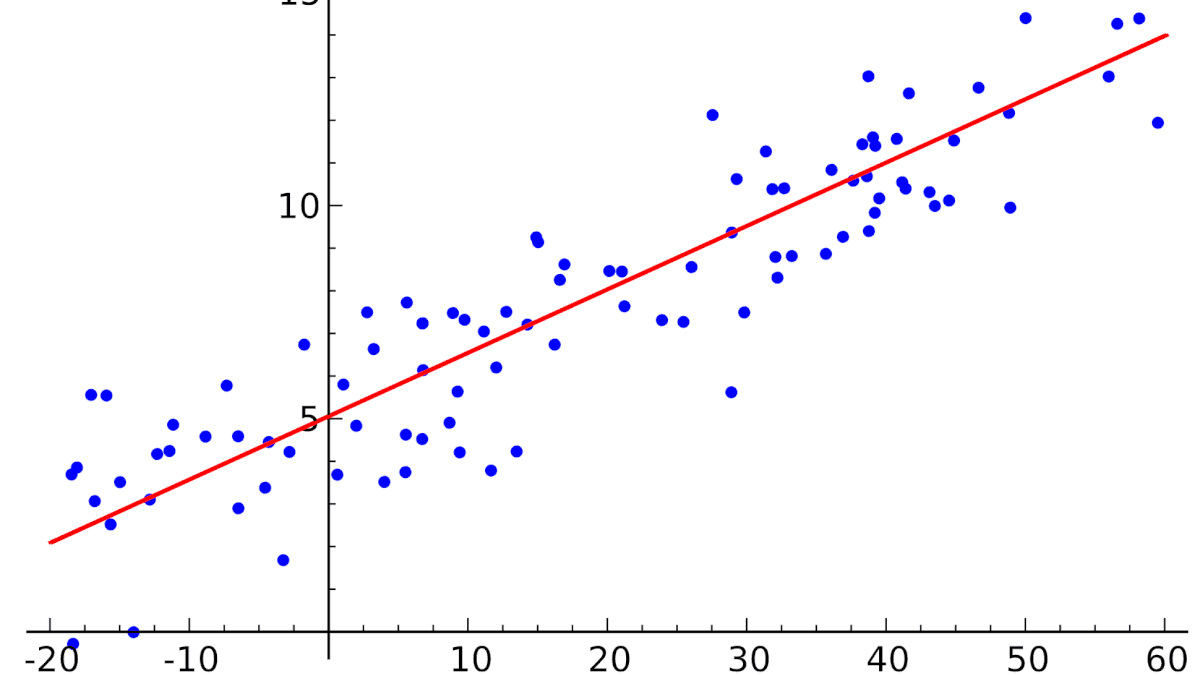
Задача регрессии часто помогает банковским работникам. Например, если человек хочет взять кредит, банк должен определить, стоит ли предоставлять запрашиваемую сумму этому человеку или нет.

Предполагается что на возврат кредита могут влиять разные социальные факторы. Банк смотрит, есть ли работа, как долго человек работает на этом месте и какая у него зарплата, какие траты по карте ежемесячно и т.д. После того как банк оценит эти показатели, он может принять решение: одобрить кредит или нет. Этот процесс теперь автоматизирован машинным обучением, а для тренировки используют большие банковские данные о клиентах.

Чтобы решить задачу регрессии, требуется построить алгоритм, так называемый регрессор. Этот алгоритм сможет спрогнозировать значение интересующей переменной. Это и будет результат работы машинного обучения — предсказание или, как обычно говорят, прогноз. Обработав набор данных, алгоритм вернет число, максимально близкое к настоящему ответу.

Однако стоит заметить, что данные в задачах не всегда могут быть в виде действительных чисел. Иногда данные выражены буквенными символами, текстом, например, название страны “Russia”. Важно уметь такие признаки приводить к числовому виду.

Задача модели — построить функцию, которая наиболее точно отразит зависимость между данными и ответом. То есть на графике строит линию, которая проходит очень близко со всеми значениями.



Когда регрессия рисует прямую линию, её называют линейной, когда кривую — полиномиальной. Это два основных вида регрессии. Существуют и другие способы предсказания ответа, например, дерево решений.

#### Задача классификации

Другая задача машинного обучения с учителем — это задача классификации. Исторически она возникла из задачи машинного зрения, когда нужно классифицировать объекты по визуальным признакам, например, определить, кто изображен на фотографии и разбить эти фотографии на группы. Примерами классификации можно считать определение типа письма (спам/не спам), языка для неизвестного текста. Благодаря классификации мы можем даже определять эмоциональную окраску поста в социальной сети.

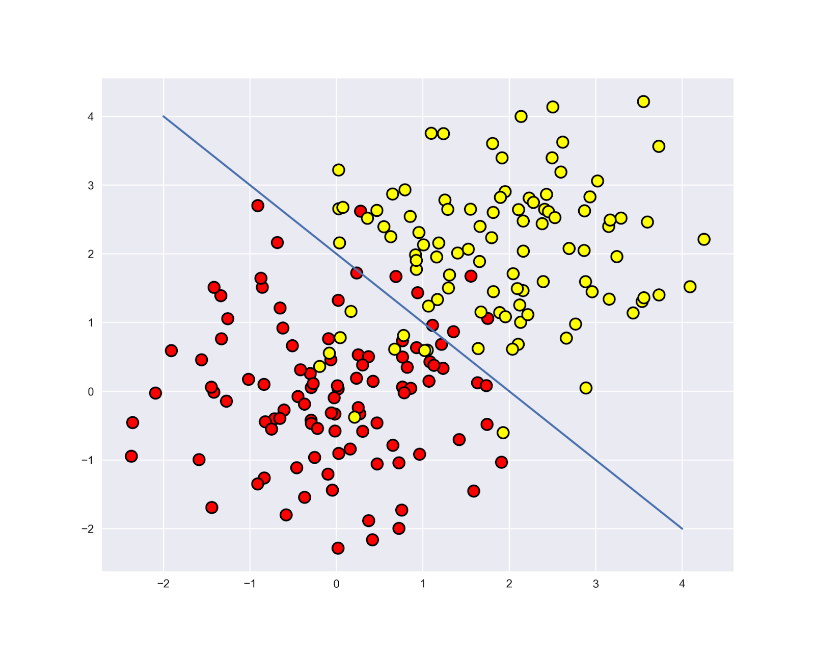
Классификация может быть бинарной, то есть в такой классификации есть всего два класса. Например, как в задаче с кошками и собаками.

В многоклассовой классификации может быть 3 класса или больше. Например, в медицине: по входным данным (результатам анализов, температуре и т.д.) требуется определить, чем пациент болеет и поставить ему диагноз. В данном случае может быть много разных болезней, по которым модель классифицирует каждого пациента.

В задаче классификации обучающая выборка — это набор отдельных объектов X, каждый из которых, как и в задаче регрессии, характеризуется вектором вещественнозначных признаков x. То есть в обучающей выборке представлены объекты с меткой класса. Необходимо найти алгоритм, который для каждого нового объекта определит принадлежность к классу.

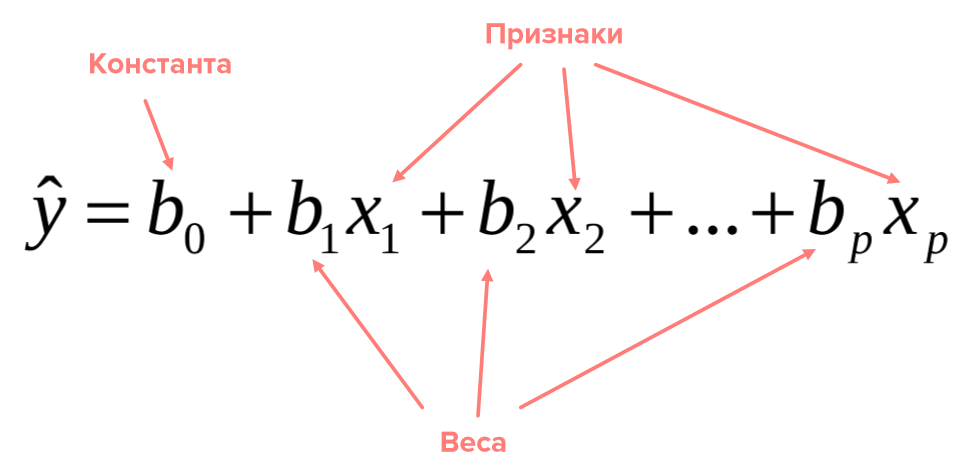
Ответом для объекта x является метка класса — переменная t, принимающая конечное число значений. Вернемся к медицине. Там X — это набор всех входных данных для всех пациентов, x — это данные конкретного пациента (температура, анализы и т.д.), а t — это предсказанный нашей моделью диагноз для конкретного пациента с данными x.

В задаче классификации строится классификатор, который по вектору признаков x возвращает ответ, к какому из классов принадлежит объект, или вероятность принадлежности к классам. На изображении график с результатом работы бинарного линейного классификатора: линия выступает как граница между двумя классами объектов.



Как и в задаче регрессии, существуют линейные и нелинейные модели для решения этой задачи.

Линейная регрессия — модель, которая строит прямую линию, опираясь на входные параметры. Формула совпадает с уравнением прямой, которую ты мог проходить в школе: y = kx + b. Единственное отличие в том, что линейную регрессию часто строят на многомерных данных (то есть количество признаков больше одного, но все они остаются линейными).



Опишем обозначения, которые мы используем:

Зависимая переменная (*ŷ*) — значение, предсказанное моделью. Это переменная, описывающая процесс, который мы пытаемся предсказать или понять.

Независимые переменные (*xi*) — i-й признак (например, *x2* — 2-й признак). Это переменные, которые мы используем для прогнозирования целевого значения, то есть зависимой переменной. В уравнении регрессии они располагаются справа от знака равенства.

Коэффициенты регрессии (*bi*) — это значения, которые мы получаем в ходе анализа. Для каждой независимой переменной получаем отдельную величину. Эти коэффициенты показывают силу и тип взаимосвязи независимой переменной по отношению к зависимой. Сложность модели будет зависеть именно от числа коэффициентов.

*bi* — коэффициент i-го признака (например, *b2* — коэффициент для второго признака)  
*b0* — свободный коэффициент.

Все, что нужно сделать, — это умножить каждый признак на коэффициент и сложить их. Коэффициент — это вес, который показывает важность того или иного признака. Чем больше коэффициент у конкретного признака, тем сильнее он влияет на прогнозируемое значение.

Задача машинного обучения — определить коэффициенты перед признаками или, как их чаще называют, веса.

# Основная часть

Разобьем разработку моделей на несколько этапов:

* Определение целей
* Сбор и подготовка данных
* Разработка модели
* Тестирование модели

## Определение целей

Необходимо понять, что нужно получить в результате работы и действительно ли для этого нужно использовать машинное обучение.

Введем понятие метрики. Метрика — это специальные числовые показатели, по которым мы можем оценить результат машинного обучения. Определи заранее, как будешь измерять результат, по каким параметрам. Разные задачи — разные метрики.

## Сбор и подготовка данных

Плохое качество данных может сделать всю твою работу бесполезной. Или ещё хуже: модель построить получится, но предсказания будут неправильными.

Предварительно обрабатываем, анализируем и готовим наши данные к работе. Данные — это топливо для алгоритмов машинного обучения: если данных мало или они плохого качества, то алгоритм не выдаст хороший результат. Этот принцип называется “Garbage in — garbage out”, или “Мусор на входе — мусор на выходе”.

Как много нужно данных для хорошей модели? Чем больше, тем лучше. Необходимо использовать все данные, которые можно достать. Далее можно сделать отбор и понять, какие данные пригодятся для решения задач, а какие — нет.

Если данных мало, то может понадобиться обогащение данных. Это процесс добавления данных в уже готовую выборку. Например, при прогнозировании спроса на жилье в каком-то районе можно добавить среднегодовую температуру в этом районе как новый признак.

## Разработка модели

Здесь мы переходим к выбору модели и её обучению. Модель МО — программа, которая по заданным входным данным выдаёт прогноз. Модель — это не реальность, а её приближение. Моделей много, они очень разные. У каждой есть свои преимущества, недостатки и уникальные особенности.

Спроектируем модель: «Зависимость оценки от количества часов обучения студентов». В данной модели метрикой будет являться вектор: {Оценка, Кол-во потраченных часов}.

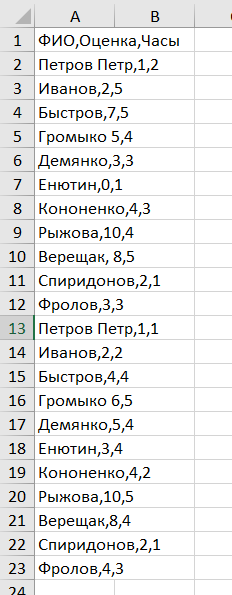
## Тестирование

Существует много разных видов тестирования, но в данной работе интересует вопрос валидации модели.

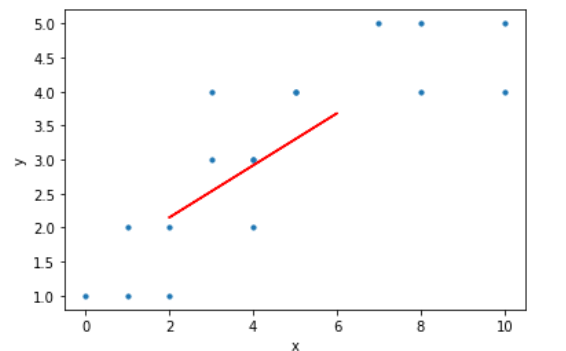
Валидация модели — это проверка соответствия предсказанных моделью данных по уже известным правильным ответам. Валидация проходит во время и после обучения модели. Для этого подготовленные данные обычно разделяют на две части: на обучающую и тестовую выборки. Обучающая выборка — это обычно большая часть датасета (около 70%), именно на этих данных обучается модель. Тестовая выборка — это оставшаяся часть подготовленного набора данных, которую мы используем, чтобы тестировать обученную модель. Для оценки результатов валидации используется метрика, которую мы выбрали в первом шаге. Этот вид тестирования помогает понять, решает ли модель поставленные задачи.

## Создание модели на языке программирования Python

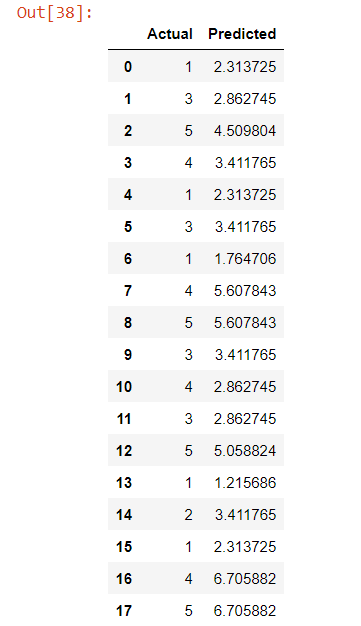
Создадим Excel файл, внутри которого поместим данные по студентам. Всего три колонки ФИО, оценка, Часы. Однако для анализа данных колонка ФИО вовсе не понадобится (пример отбора данных).



Возьмем 20% от данных и получим результат:



Для более «приятного восприятия» создадим таблицу, где будут отображаться реальные данные и данные, которые спрогнозировала модель:



Результат, безусловно, не идеальный. Однако стоит учитывать, что данных было мало для более точного анализа. Но заметим, что даже их хватило для приблизительного прогнозирования.

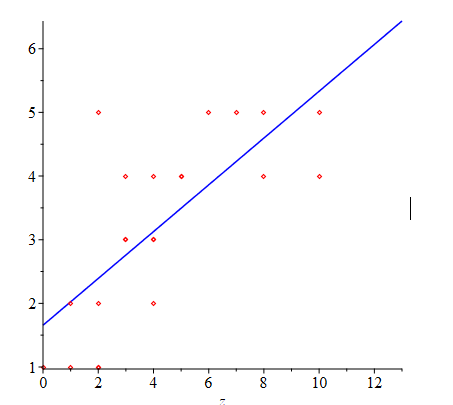
## Создание модели в Maple

Проведем теперь аналогичные действия, однако с помощью математического пакета Maple. Используем метод наименьших квадратов.

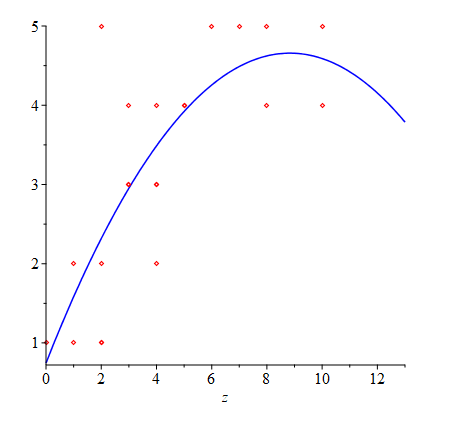
Суть метода наименьших квадратов.

Пусть имеется x – набор n неизвестных переменных, fi(x) –совокупность функций от этого набора переменных. Задача заключается в подборе таких x, чтобы значения функций были максимально близки к некоторых значениям y.

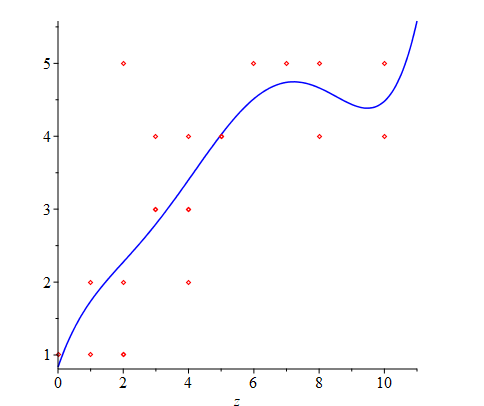
Используя базис с максимальной степенью 1, получили график линейной зависимости:



Изменив базис на [1,x,x2] получим полиномиальную зависимость:



Увеличим степень базиса до 5: [1,x2,x3,x4,x5]



Таким образом, увеличивая степень базиса – мы все лучше описываем нашу зависимость.

По получившейся функции можно делать прогнозы (предсказания).

Например:



То есть при 10 часах работы, студент напишет контрольную на 4 балла, при 3 часах подготовки – на 3 балла, при 7 часах – 4 или 5 соответственно.

# Вывод

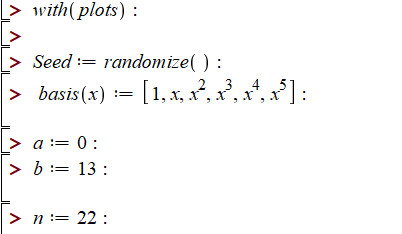
В работе была поставлена цель: разработать программу, которая будет осуществлять регрессию. Данная цель была выполнена. Разработана программа на языке программирования Python, а также программа для прогнозирования, используя метод наименьших квадратов.

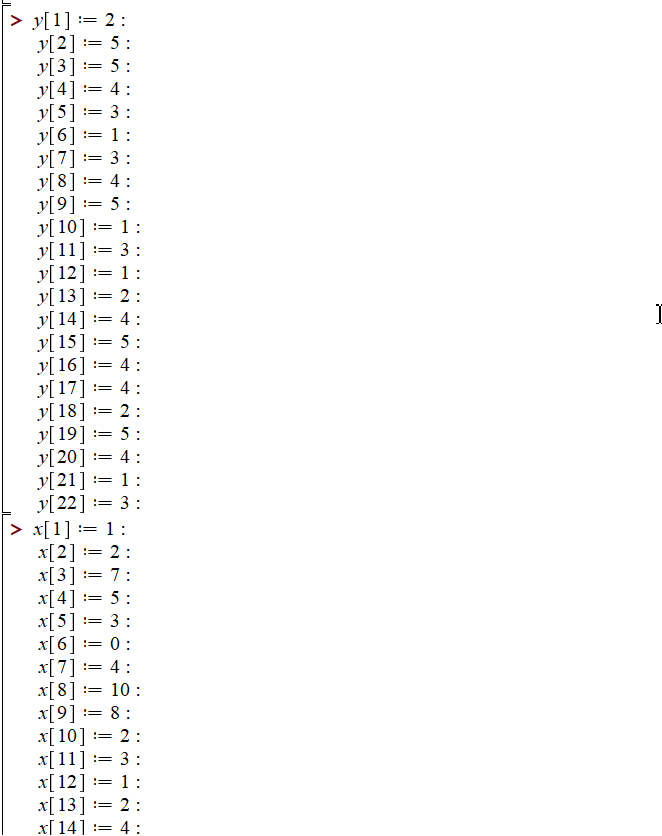
Список литературы

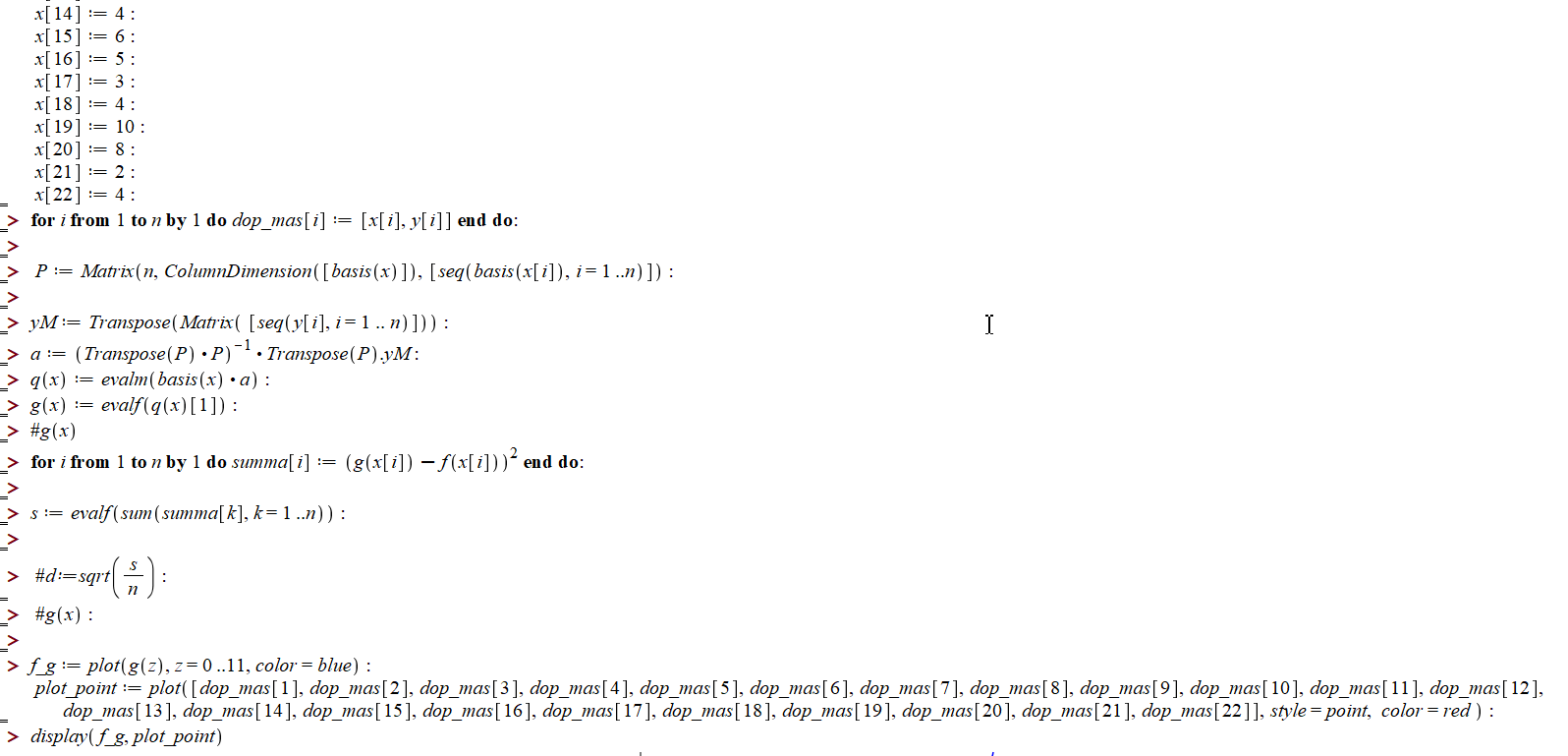
1. Донской В. И. Машинное обучение и обучаемость (2012)
2. Машинное обучение // MachineLearning.ru URL: <http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение>
3. Как обучаются машины? Научно-популярная статья URL: http://www.machinelearning.ru/wiki/index.php?title=%D0%9A%D0%B0%D0%BA\_%D0%BE%D0%B1%D1%83%D1%87%D0%B0%D1%8E%D1%82%D1%81%D1%8F\_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D1%8B%3F\_%D0%9D%D0%B0%D1%83%D1%87%D0%BD%D0%BE-%D0%BF%D0%BE%D0%BF%D1%83%D0%BB%D1%8F%D1%80%D0%BD%D0%B0%D1%8F\_%D1%81%D1%82%D0%B0%D1%82%D1%8C%D1%8F
4. Прикладной регрессионный анализ (курс лекций, B.В. Стрижов) URL: http://www.machinelearning.ru/wiki/index.php?title=%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%B0%D1%8F\_%D1%80%D0%B5%D0%B3%D1%80%D0%B5%D1%81%D1%81%D0%B8%D1%8F\_%D0%B8\_%D0%BE%D0%BF%D1%82%D0%B8%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F\_%28%D0%BA%D1%83%D1%80%D1%81\_%D0%BB%D0%B5%D0%BA%D1%86%D0%B8%D0%B9%2C\_B.%D0%92.%D0%A1%D1%82%D1%80%D0%B8%D0%B6%D0%BE%D0%B2%29
5. Линейная регрессия (пример) URL: http://www.machinelearning.ru/wiki/index.php?title=%D0%9B%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D0%B0%D1%8F\_%D1%80%D0%B5%D0%B3%D1%80%D0%B5%D1%81%D1%81%D0%B8%D1%8F

Приложение:

Код программы Maple







Код программы в Python:

