

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ**

Государственное образовательное учреждение высшего  
профессионального образования

**«МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ  
(государственный университет)»**

**Факультет управления и прикладной математики**

**Кафедра Интеллектуальные системы**

**Яшков Даниил Дмитриевич**

**УЧЁТ ФИЗИЧЕСКИХ ОГРАНИЧЕНИЙ В  
ПРЕДСКАЗАТЕЛЬНЫХ МОДЕЛЯХ  
ХИМИЧЕСКОГО СОСТАВА СТАЛИ**

**Магистерская диссертация**

**направление подготовки:**

**01.04.02 «Прикладная математика и информатика»**

Магистерская программа 010400 Анализ данных

Научный руководитель:

к.ф.-м.н. Лобачев В.А. / \_\_\_\_\_

Студент:

Яшков Д.Д. / \_\_\_\_\_

Москва 2016

# Оглавление

<b>1</b>	<b>Аннотация</b>	<b>2</b>
<b>2</b>	<b>Введение</b>	<b>3</b>
<b>3</b>	<b>Постановка задачи</b>	<b>5</b>
<b>4</b>	<b>Существующие исследования</b>	<b>7</b>
<b>5</b>	<b>Описание предложенных методов</b>	<b>8</b>
5.1	Линейная регрессия + MatrixNet . . . . .	8
5.2	Кусочно-линейная регрессия + MatrixNet . . . . .	12
5.3	Нейронные сети . . . . .	17
<b>6</b>	<b>Построение моделей</b>	<b>19</b>
6.1	Построение признакового описания . . . . .	19
6.2	Фильтрация данных и отбор признаков . . . . .	22
6.3	Обучение моделей . . . . .	26
<b>7</b>	<b>Сравнение моделей</b>	<b>28</b>
<b>8</b>	<b>Заключение</b>	<b>37</b>

# 1. Аннотация

В данной работе рассматривается применение методов анализа данных и машинного обучения для моделирования конвертерного процесса выплавки стали. В частности, рассматривается влияние легирующих добавок в виде ферросплавов для получения необходимого химического состава стали. В работе предложены двухуровневые методы построения регрессионных моделей, предсказывающих химический состав стали, основанные на параметрах плавки и добавлениях ферросплавов. Помимо стандартных метрик точности алгоритма, в работе рассматриваются физические ограничения, накладываемые на регрессионную модель, связанные с процессом выплавки стали, а также анализируется интерпретируемость полученной модели.

## 2. Введение

Конвертерный процесс производства стали является наиболее популярным способом в силу высокой производительности и относительно невысокой стоимости. Для того, чтобы сталь получила необходимые физические свойства, она должна иметь определенный химический состав. Требования к химическому составу стали задаются диапазонами содержаний элементов, а соответствие химического состава стали заданному диапазону достигается с помощью легирования стали - введения ферросплавов. Metallургические предприятия тратят миллиарды долларов на ферросплавы, поэтому задача оптимизации стоимости добавленных ферросплавов играет огромную роль в производстве.

Один из возможных способов решения данной задачи заключается в реализации двух последовательных этапов:

1. На первом этапе строится предсказательная модель химического состава стали, способная по набору статических параметров плавки и набору ферросплавов предсказывать химический состав готовой стали.
2. На втором этапе строится оптимизационная задача, решение которой минимизирует стоимость добавленных ферросплавов при условии того, что полученная сталь попадет в заданные диапазоны содержаний химических элементов.

При решении задачи оптимизации необходимо уметь предсказывать заданный химический состав стали по известным параметрам. Для решения оптимизационной задачи можно использовать black-box алгоритмы оптимизации [1, 2], не накладывающие никаких условий на вид оптимизируемой функции и ограничений.

Целью данной работы является решение задачи первого этапа - построение предсказательной модели химического состава стали. Так как полученная модель будет использоваться при решении задачи оптимизации, а оптимум может достигаться вне области исторических данных, необходимо чтобы она была интерпретируемой и физически корректной, в том числе, в областях, не покрытых историческими данными.

В данной работе показывается, что применение классических методов машинного обучения не дает желаемого результата, и предлагается двухуровневый алгоритм построения интерпретируемой регрессионной модели.

### 3. Постановка задачи

Введем несколько обозначений:

- $\vec{x}$  - количество вносимых ферросплавов
- $x_f$  - элемент вектора  $\vec{x}$ , соответствующий ферросплаву  $f$
- $E$  - множество химических элементов
- $f_e$  - доля содержания элемента  $e \in E$  в ферросплаве  $f$
- $x_e$  - суммарное внесенное количество элемента  $e \in E$   
 $x_e = \sum_f x_f \cdot f_e$
- $\vec{a}$  - остальные параметры плавки, не вошедшие в  $\vec{x}$
- $\vec{y}$  - химический состав стали
- $y_e$  - элемент вектора  $\vec{y}$ , полученное содержание элемента  $e$  в стали

Необходимо построить регрессионную зависимость:

$$\vec{y} = f(\vec{x}, \vec{a}) \quad (3.1)$$

При этом мы хотим, чтобы полученная модель была интерпретируема и удовлетворяла некоторым физическим свойствам: мы хотим, чтобы при увеличении количества добавляемого элемента предсказанный химический состав стали по этому элементу не уменьшался. Если сформулировать это математически:

$$\frac{\partial y_e}{\partial \sum_{x \in \vec{x}} x_e} \geq 0, \quad (3.2)$$

То есть это условие просто говорит нам о том, что у полученной компоненты регрессионной зависимости частные производные по добавляемому элементу должны монотонно неубывать. При этом мы не хотим накладывать никаких дополнительных ограничений на предсказания модели, поэтому, будем смотреть, насколько хорошо данное условие выполняется для различных областей данных.

Для построенной обучающей выборки  $\mathbf{X}, \mathbf{y} = (\vec{x}^i, \vec{a}^i, \vec{y}^i)_{i=1}^N$ , запишем функционал качества регрессии для конкретного элемента  $e$ .

$$RMSE_e = \sqrt{\frac{1}{N} \sum_{i=1}^N (\vec{y}_e^i - f(\vec{x}^i, \vec{a}^i)_e)^2} \quad (3.3)$$

Предлагается рассмотреть более простой случай - вместо построения одной векторной регрессионной зависимости, построить несколько регрессионных зависимостей, по одной для каждой компоненты вектора  $\vec{y}$ , соответственно и качество для данных моделей считать отдельно.

В таком случае,  $f(\vec{x}^i, \vec{a}^i)_e = f_e(\vec{x}^i, \vec{a}^i)$ .

Таким образом искомая задача построения векторной регрессионной модели сводится к набору задач оптимизации:

$$\sum_{i=1}^N (\vec{y}_e^i - f_e(\vec{x}^i, \vec{a}^i))^2 \longrightarrow \min_{f_e}, e \in E \quad (3.4)$$

При этом, надо заметить, что введение функционала качества, который будет оценивать совокупность построенных моделей, довольно сложная задача, требующая большого количества экспертных суждений, так как все элементы имеют разный порядок содержаний (от нескольких процентов до сотых долей процента) и имеют разную важность.

## 4. Существующие исследования

В течение последних десятилетий активно растет число исследований, посвященных проблемам прогнозирования на основе моделирования различных сложных процессов, которые часто встречаются во многих отраслях промышленности и реальных жизненных ситуациях.

В металлургии широко распространена задача оптимизации процесса продувки кислородом. Например, в работе [3] с помощью модификации алгоритма дифференциальной эволюции поддерживается оптимальность параметров последнего этапа продувки: объем продуваемого кислорода, режимы работы фурмы, время и объем добавления различных вспомогательных материалов. В статье [4] рассматривается задача построения модели динамического оптимального управления. В первой части вычисляются оптимальные параметры продувки (объем кислорода и охлаждающей жидкости), во второй части предсказываются температура и содержание углерода в полученной стали.

Работа [5] посвящена построению математической модели продувки в кислородном конвертере. Модель включает в себя кинетику плавки лома, параметры растворения, образование и положение капель металла в эмульсии, кинетику реакции выгорания углерода в различных зонах конвертера, химию шлака.

Вообще, математическое моделирование процесса выплавки стали довольно широко распространенная задача [6, 7, 8, 9]. В каждой из этих работ рассматривается один определенный этап производства стали.

В настоящее время нет работ, посвященных оптимизации стоимости добавляемых ферросплавов, или работ, посвященных построению предсказательной модели химического состава стали.



# 5. Описание предложенных методов

## 5.1 Линейная регрессия + MatrixNet

### Линейная регрессия

Дана обучающая выборка  $X^l = (x_i, y_i)_{i=1}^l$ , введем несколько обозначений:

- матрица объекты-признаки  $\mathbf{X} = (x_1, \dots, x_l)^T$
- вектор ответов  $\vec{y} = (y_1, \dots, y_l)^T$

Мы хотим построить модель в виде:

$$\hat{y}(x) = \omega^T x + b \quad (5.1)$$

при этом, минимизируя среднеквадратическую ошибку (RMSE) на обучающей выборке  $X^l$ .

$$\sqrt{\frac{1}{l} \sum_{i=1}^l (\hat{y}(x_i) - y_i)^2} \longrightarrow \min \quad (5.2)$$

В искомой модели легко избавиться от  $b$ , добавив константный признак в матрицу  $\mathbf{X}$ . В дальнейшем считаем, что этот константный признак уже добавлен.

Решение этой задачи легко находится:

$$\omega = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \vec{y} \quad (5.3)$$

Предсказания линейной регрессии на объекте  $x$  будем обозначать как  $LR(x)$

## Градиентный бустинг над решающими деревьями

Бустинг - алгоритм машинного обучения, строящий предсказательную модель в виде композиции "слабых" алгоритмов.

Пусть есть

- обучающая выборка  $X^l = (x_i, y_i)_{i=1}^l$
- семейство базовых алгоритмов  $\mathcal{B}$
- функция потерь  $\mathcal{L}$

Градиентный бустинг строит модель итерационно, на каждом шаге увеличивая размер полученной композиции, добавляя в нее алгоритм из семейства базовых алгоритмов  $\mathcal{B}$ .

Общий вид полученной модели можно представить как:

$$a(x) = \sum_{t=1}^T \alpha_t b_t(x), b_t \in \mathcal{B} \quad (5.4)$$

На каждой итерации минимизируется функционал качества с заранее определенной функцией потерь  $\mathcal{L}(a, y)$ :

$$Q(\alpha, b, X^l) = \sum_{i=1}^l \mathcal{L}\left(\sum_{t=1}^T \alpha_t b_t(x_i), y_i\right) \longrightarrow \min_{\alpha, b} \quad (5.5)$$

Градиентный бустинг решает данную задачу следующим образом: на каждой итерации он ищет такой базовый алгоритм, который наилучшим образом аппроксимирует вектор антиградиента, достаточно полно это

представлено в статье [10].

$$b_t = \arg \min_b \sum_{i=1}^l (b(x_i) + \mathcal{L}'(f_i, y_i))^2$$

$$\alpha_t = \arg \min_{\alpha > 0} \sum_{i=1}^l \mathcal{L}(f_i + \alpha b_t(x_i), y_i)$$

$$f_i = \sum_{k=1}^{t-1} \alpha_k b_k(x_i)$$

Обоснование бустинга приведено в [11], в этой работе приводится верхняя оценка на количество ошибочных классификаций, которая не зависит явным образом от количества алгоритмов, используемых в композиции, а зависит от сложности базового семейства, то есть выгоднее брать более простое базовое семейство алгоритмов. Чаще всего градиентный бустинг применяется к решающим деревьям, то есть в качестве базового семейства алгоритмов берутся решающие деревья.

В данной работе мы использовали MatrixNet [12], что по сути является градиентным бустингом над ODT - oblivion decision trees, являющимися частным случаем решающих деревьев. ODT отличаются от обычных решающих деревьев тем, что в них для каждой вершины количество вершин в левом и правом поддереве совпадает. Кроме этого, при построении MatrixNet-а накладывается ещё несколько ограничений на ODT:

- 1 На каждом уровне дерева рассматривается только один признак  $x_j$
- 2 Для всех вершин дерева, находящихся на одном уровне, рассматривается одно и то же пороговое правило:  $[x_j > \delta]$

Таким образом, сложность базового семейства сильно уменьшается по сравнению с семейством всех решающих деревьев, что приводит к дополнительной регуляризации и улучшению обобщающей способности.

В дальнейшем, предсказания модели MatrixNet на объекте  $x$  будем обозначать как  $mnet(x)$

## Линейная регрессия + MatrixNet

Данный алгоритм можно представить как композицию двух предыдущих:

$$a(x) = mnet(x) + LR(\tilde{x}) \quad (5.6)$$

Обучение этих двух моделей производится следующим образом

1 Строим линейную регрессию  $y = \omega^T \tilde{x}$

2 В качестве обучающей выборки для MatrixNet используем  $(x_i, y_i - \omega^T \tilde{x}_i)_{i=1}^l$

Тут  $\tilde{x}$  - вектор признаков, который подается в линейную регрессию. Подразумевается, что этот вектор признаков может отличаться от вектора, который будет использовать MatrixNet.

Идейно, данный алгоритм может использоваться в случае наличия каких-либо линейных трендов в исходных данных. MatrixNet, да и вообще решающие деревья, строят модели, которые являются кусочно-постоянными. Поэтому, при наличии линейного тренда в данных, будет лучше избавиться сначала от тренда, а после этого аппроксимировать остатки с помощью MatrixNet.

Докажем одно небольшое утверждение:

**Утверждение 1.** Пусть отсутствуют объекты обучающей выборки, для которых  $j$ -й признак принимает значения из промежутка  $[a, b]$ , тогда MatrixNet, обученный на этой выборке, будет принимать не более 2-х различных значений на этом промежутке, при фиксированном остальном наборе признаков.

**Доказательство.** Допустим MatrixNet принимает более двух различных значений. Тогда в обученной модели есть как минимум два пороговых правила  $[x_j < \theta_1]$ ,  $[x_j < \theta_2]$  таких, что  $\theta_i \in [a, b]$ ,  $i = 1, 2$ ,  $\theta_1 < \theta_2$ . В силу того, что на каждой итерации добавляется ровно одно пороговое правило, получим, что какое-то из них было добавлено раньше. Не уменьшая общности, будем считать что это  $[x_j < \theta_1]$ . Но тогда в любой последующий момент добавление порогового правила  $[x_j < \theta_2]$  не изменит

разбиение обучающей выборки по листьям, то есть не изменит градиента ошибки. Это противоречит итерационному процессу, в соответствии с которым мы добавляем следующее пороговое правило. ■

## 5.2 Кусочно-линейная регрессия + MatrixNet

### Кусочно-линейная регрессия

Логичным обобщением линейной модели является кусочно-линейная модель [pwlr]. Кусочно-линейная регрессия строилась в пространстве двух признаков. Для начала опишем общую постановку задачи её построения

#### Общая постановка задачи.

Допустим, мы имеем конечное число точек  $(x^i, y_i) \in \mathbb{R}^n \times \mathbb{R}, i = 1, \dots, m \in \mathbb{N}$ . Тогда мы хотим решить следующую задачу:

$$\min \|w - y\|_q, \tag{5.7a}$$

$$w_i = f(x^i), \forall i = 1, \dots, m, \tag{5.7b}$$

$$f \in F \tag{5.7c}$$

где  $F$  определяется как множество кусочно-линейных функций с общей областью определения, содержащей все точки  $x^i$ , а  $\|\cdot\|_q$  -  $\ell_q$ -норма в  $\mathbb{R}^n$ . Другими словами, мы хотим найти функцию  $f^* \in F$ , наилучшим образом аппроксимирующую точки из набора данных в соответствии с мерой  $\|\cdot\|_q$ .

Определим кусочно-линейную функцию как непрерывную функцию  $f$  с областью определения  $\bigcup_{P \in \mathcal{P}} P$ , где множество  $\mathcal{P}$  конечно, каждое множество  $P \in \mathcal{P}$  - политоп полной размерности, для любых двух  $P, Q \in \mathcal{P}$  их внутренности не пересекаются,  $f$  линейна в области, ограниченной любым политопом  $P \in \mathcal{P}$ .

Для  $q < \infty$ , уравнение (5.7) эквивалентно

$$\min \sum_{i=1}^m |f(x^i) - y_i|^q, \quad (5.8a)$$

$$f \in F \quad (5.8b)$$

Далее, мы будем считать, что  $q = 2$ .

Теперь рассмотрим случай, когда  $F$  - множество кусочно-линейных функций, определенных на заранее известном объединении политопов  $\mathcal{P}$ . Обозначим  $V(P)$  - множество вершин  $P \in \mathcal{P}$ , и пусть  $\mathcal{V}(\mathcal{P}) = \bigcup_{P \in \mathcal{P}} V(P)$ . Также, для каждой вершины  $v \in \mathcal{V}$  введём  $\mathcal{P}_v$  - множество политопов, содержащих  $v$ , а для каждого  $i$ , обозначим политоп, содержащий  $x^i$ , как  $P_i \in \mathcal{P}$ . Пример для двухмерного случая изображен на рисунке 5.1.

Для всех  $P \in \mathcal{P}$  обозначим  $V^+(P)$  множество из  $n+1$  линейно независимых вершин  $P$ . Для любого  $x \in P$  пусть  $\lambda_x^{P,v}, v \in V^+(P)$  - множество барицентрических координат  $x$  относительно  $V^+(P)$ , то есть уникальный набор скалярных параметров, с помощью которых можно представить  $x$  в виде линейной комбинации элементов  $V^+(P)$ . Пример для двухмерного случая изображен на рисунке 5.2. Тогда любая функция  $f \in F$  может быть представлена в виде

$$f(x) = \sum_{v \in V^+(P)} \lambda_x^{P,v} f_v, \forall x \in P \quad (5.9)$$

Для каждого  $x^i$  обозначим  $\lambda_i^{P_i,v}, v \in V^+(P_i)$  аналогично  $\lambda_x^{P,v}$ . В этом случае модель принимает вид:

$$\min \sum_{i=1}^m \left| \sum_{v \in V^+(P_i)} \lambda_i^{P_i,v} f_v - y_i \right|^q, \quad (5.10a)$$

$$f_v = \sum_{u \in V^+(P)} \lambda_v^{P,u} f_u, \forall P \in \mathcal{P}_v, \forall v \in \mathcal{V}(\mathcal{P}) \quad (5.10b)$$

$$f_v \in \mathbb{R}, \forall v \in \mathcal{V}(\mathcal{P}) \quad (5.10c)$$

где  $f_v = f(v), \forall v \in \mathcal{V}(\mathcal{P})$ .

Условия (5.10b) необходимы, так как значения функции в каждом  $P$

выражаются через линейную комбинацию значений функции в точках из  $V^+(P)$ , поэтому значения функции в вершинах должны совпадать в смежных политопах. Однако здесь стоит заметить, что если  $\mathcal{P}$  является триангуляцией или ее обобщением для более высоких размерностей, то условия (5.10b) будут априори выполнены, поэтому они могут быть исключены, и (5.10) сводится к классической задаче безусловной оптимизации для переопределенной системы линейных уравнений.

Так как мы будем строить кусочно-линейную регрессию в пространстве двух признаков, нас будет интересовать именно триангулированная область определения для  $n = 2$  и  $q = 2$ . Тогда для каждой точки необходимо определить треугольник, которому она принадлежит, а затем вычислить барицентрические координаты этой точки относительно треугольника (три коэффициента). Обозначим  $\lambda_i^{v_{ij}}, j \in \{0, 1, 2\}$  коэффициенты для  $x^i$ ,  $v_{ij}$  - вершины  $P_i$ . Положив  $f_i = f(x^i)$ , имеем  $f_i = \lambda_i^{v_{i1}} f_{v_{i1}} + \lambda_i^{v_{i2}} f_{v_{i2}} + \lambda_i^{v_{i3}} f_{v_{i3}}$ , тогда модель запишется следующим образом:

$$\min \sum_{i=1}^m |f_i - y_i|^2 \quad (5.11a)$$

$$f_i = \lambda_i^{v_{i1}} f_{v_{i1}} + \lambda_i^{v_{i2}} f_{v_{i2}} + \lambda_i^{v_{i3}} f_{v_{i3}}, \forall i = 1, \dots, m \quad (5.11b)$$

Полученную систему можно решить методом наименьших квадратов (МНК), как описано в [13]. Неизвестными здесь являются значения  $f_v$  функции в вершинах треугольников, так как для любой точки  $x^i$  треугольник, которому она принадлежит, и соответствующие барицентрические координаты точки относительно его вершин определяются однозначно.

### **Регуляризация.**

Полученная модель имеет ряд недостатков.

Во-первых, так как нашей целью является построение модели, отражающей локальные тренды в данных, на остатках которой будет строиться новая модель, то желательно, чтобы поведение модели от треугольника к треугольнику менялось несильно. Основная задача - не допустить резких перегибов кусочно-линейной поверхности, которые могут возникнуть при решении задачи оптимизации за счет высокой степени свободы в областях с малым количеством данных.

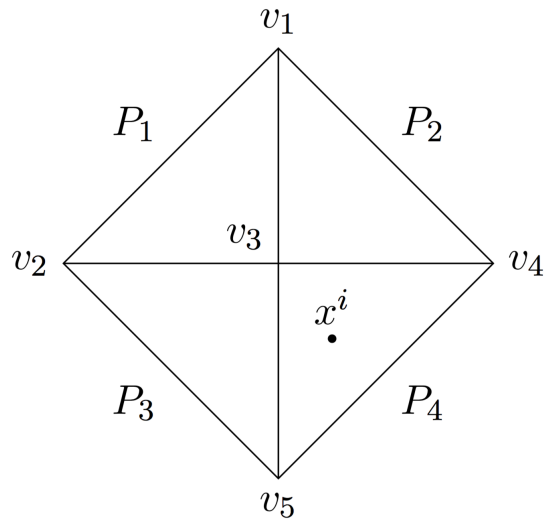


Рис. 5.1: Область определения при  $n = 2$ .  
 $V(P_2) = \{v_1, v_3, v_4\}$ ,  $\mathcal{P}_{v_2} = \{P_1, P_3\}$ ,  $P_i = P_4$ .

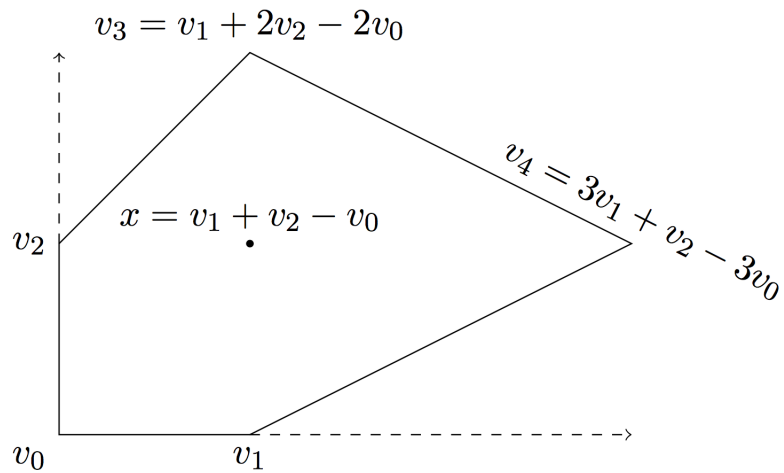


Рис. 5.2: Барцентрические координаты точки при  $n = 2$  и  $V^+(P) = \{v_0, v_1, v_2\}$ .



Во-вторых, недостатком модели являются нулевые столбцы в матрице  $\mathbf{X}$  в случае, если в триангуляции существует хотя бы один узел, смежные треугольники которого не содержат ни одной точки из данных. Пусть такой узел существует и ему соответствует номер  $t$ . Тогда понятно, что для того, чтобы хотя бы одна из строк матрицы  $\mathbf{X}$  имела в столбце  $t$  значение, отличное от нуля, необходимо, чтобы, по крайней мере, одна точка в данных принадлежала какому-то из треугольников, имеющему в качестве одной из вершин  $v_t$ . В случае, когда данные в области определения распределены неравномерно, при определенных триангуляциях высока вероятность существования "плохих" узлов. Если в матрице  $\mathbf{X}$  присутствуют нулевые столбцы, то  $\det(\mathbf{X}) = 0$  и обратить матрицу  $\mathbf{X}^T \mathbf{X}$  невозможно, так как  $\det(\mathbf{X}^T \mathbf{X}) = \det(\mathbf{X}^T) \det(\mathbf{X}) = 0$ .

Для решения обеих проблем предлагается наложить ограничение на разности градиентов смежных треугольников. Предположим, что всего в триангуляции  $L$  треугольников. Пронумеруем их от 1 до  $L$ . Обозначим  $P^l \in \mathcal{P}, \forall l = 1, \dots, L$  треугольник под номером  $l$ .

Пусть

$$\mathcal{N} = \left\{ (P^{l_1}, P^{l_2}) \left| \begin{array}{l} l_1, l_2 \in \{1, \dots, L\}, \quad l_1 < l_2, \\ \exists v_1, v_2 \in \mathcal{V}(\mathcal{P}) : (v_1 \in P^{l_1} \cap P^{l_2}) \wedge (v_2 \in P^{l_1} \cap P^{l_2}) \end{array} \right. \right\}$$

- множество всех пар смежных треугольников (с точностью до перестановки внутри пары). Понятно, что в силу линейности функции в области, ограниченной любым из треугольников, выполнено:

$$\nabla f(x_1) = \nabla f(x_2) \quad \forall x_1, x_2 \in P \quad \forall P \in \mathcal{P} \quad (5.12)$$

Тогда целевую функцию модели (5.11) дополним ограничением на сумму квадратов разностей градиентов функции для всех пар смежных треугольников. Обозначим  $\nabla f_l, l = 1, \dots, L$  значение вектора градиента в произвольной точке  $l$ -ого треугольника (выбор точки не имеет значения).

Получаем итоговую модель:

$$\min \sum_{i=1}^m |f_i - y_i|^2 + \lambda \sum_{(P^{l_1}, P^{l_2}) \in \mathcal{N}} \|\nabla f_{l_1} - \nabla f_{l_2}\|_2 \quad (5.13a)$$

$$f_i = \lambda_i^{v_{i1}} f_{v_{i1}} + \lambda_i^{v_{i2}} f_{v_{i2}} + \lambda_i^{v_{i3}} f_{v_{i3}}, \forall i = 1, \dots, m \quad (5.13b)$$

В дальнейшем предсказание кусочно-линейной модели на объекте  $x$  будем обозначать  $PWLR(x)$ .

## Кусочно-линейная регрессия + MatrixNet

По аналогии с 5.6 можно обучить совместную модель кусочно-линейной регрессии и MatrixNet:

$$a(x) = mnet(x) + PWLR(\tilde{x}) \quad (5.14)$$

Обучение модели 5.14 отличается от 5.6 использованием кусочно-линейной регрессии вместо линейной.

1. Строим кусочно-линейную регрессию  $y_{pwlr} = f(\tilde{x})$
2. В качестве обучающей выборки для MatrixNet используем  $(x_i, y_i - y_{pwlr})_{i=1}^l$

## 5.3 Нейронные сети

Рассмотрим классические нейронные сети, состоящие из выходного слоя и скрытых полносвязных слоев. По обучающей выборке  $X^l = (x_i, y_i)_{i=1}^l$  надо построить регрессионную зависимость  $\hat{y} = f(x)$ , где  $x$  - вектор признаков объекта. Опишем, как работает один слой нейросети.

Выход полносвязного слоя нейросети определяется матрицей весов и функцией активации:

$$f^i(x) = f_{act}^i(\mathbf{W}_i x), \quad (5.15)$$

здесь использованы следующие обозначения:

- $f_{act}$  - функция активации

- $W$  - матрица весов
- $x$  - вход полносвязного слоя
- $f(x)$  - выход полносвязного слоя
- $i$  - номер слоя

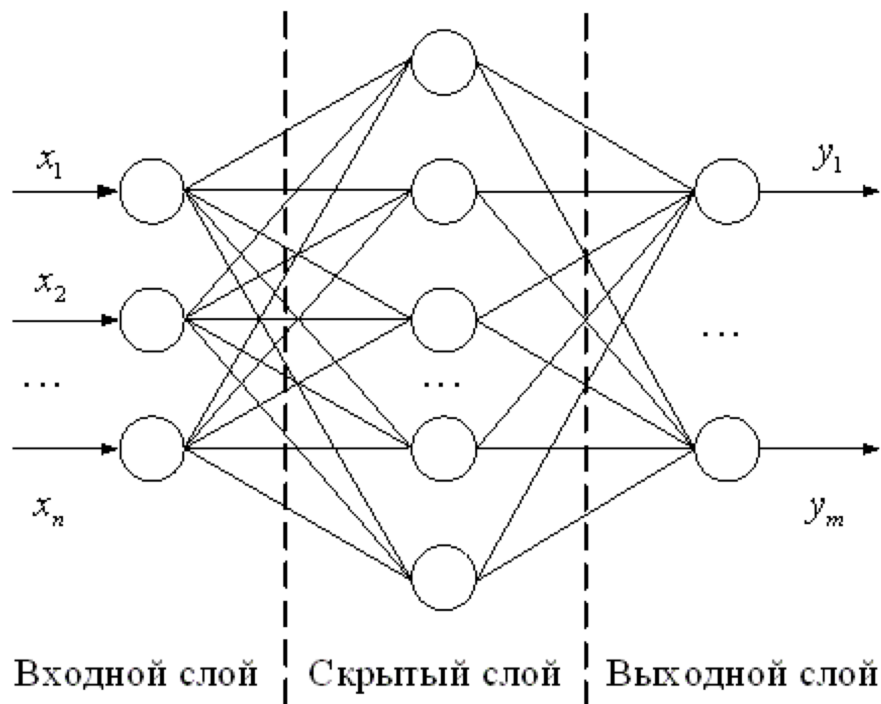


Рис. 5.3: Нейросеть с одним скрытым слоем

Нейронная сеть, состоящая из одного полносвязного слоя (вообще, подразумевается, что она состоит из двух слоев - входного и выходного, но т.к. входной слой весов не имеет, обычно его не учитывают), представляет собой обычную линейную регрессию, поэтому стоит рассматривать нейросети, состоящие из двух слоев и более. Схематическое изображение нейросети с одним скрытым слоем представлено на рис. 5.3

Процесс обучения нейросети заключается в настройке параметров слоев, а именно - весов нейросети. Веса слоев настраиваются с помощью метода обратного распространения ошибки, хорошо описанного в статье [14]. При этом архитектура нейросети ( количество слоев, количество нейронов на каждом слое, функции активации нейронов) должна быть зафиксирована заранее. Настройка вышеуказанных параметров требует больших вычислительных ресурсов.

## 6. Построение моделей

### 6.1 Построение признакового описания

Признаковое описание можно разделить на две части: статические признаки и добавления ферросплавов. Основное отличие текущих групп признаков проявляется в дальнейшем использовании данной модели - на этапе оптимизации ферросплавы будут настраиваемыми параметрами.

Многие признаки были введены, исходя из замечаний экспертов. Так как исходные данные были очень шумными и с большим количеством пропусков, в основном использовались признаки, которые, по мнению экспертов, сильно влияли на процесс плавки. Следующие признаки были посчитаны из входных данных:

- масса лома, чугуна
- отношение массы лома к чугуну
- шихтовочный вес плавки (сумма масс лома и чугуна)
- нормированный шихтовочный вес плавки - шихтовочный вес, умноженный на коэффициент  $(1 - s)$ , где  $s$  - приблизительная доля стали, уходящая в шлак, равная 0.111
- длительность слива стали
- стойкость футеровки
- признаки, отражающие температуру воздуха на улице - номер месяца, сезон, средняя температура в месяце
- группа марки стали (one-hot encoded)

- технологическая группа стали (one-hot encoded)
- время начала продувки (с момента начала плавки)
- объем продутого кислорода
- содержание углерода
- окисленность металла

Признаки, отражающие добавления ферросплавов:

- суммарная масса добавленного элемента (по каждому элементу)
- относительная масса добавленного элемента (по каждому элементу) - отношение суммарной массы добавленного элемента к шихтовочному весу плавки.

## Построение признаков поэлементных добавлений

Исходными данными являлись добавления ферросплавов в каждой плавке. При этом было известно среднее содержание чистого элемента в конкретном ферросплаве, которое сохранялось на протяжении всего исторического периода. Также были известны входные анализы исторических закупок ферросплавов. По информации, полученной от экспертов, истинное содержание чистого элемента в ферросплавах может отличаться на 5 – 10% от среднего значения. Поэтому для получения более точной информации о содержаниях чистых элементов в ферросплавах в конкретной плавке решалась следующая задача:

- **Дано:** Множество исторических плавков  $X$ . Массы добавленных ферросплавов в плавке  $x \in X$ ,  
 $(x_f)_{f \in F}$ ,  
 день плавки  $d$ ,  
 входные анализы ферросплавов  $(f_e^d)_{f \in F, e \in E}$ ,  
 $d$  - день, когда был получен входной анализ
- **Найти:**  $\tilde{f}_e^d$  - содержание элемента  $e$  в ферросплаве  $f$  в день  $d$ .

ТПо причине того, что ферросплав начинают использовать не сразу после его получения, а только через несколько дней, и что кроме него могут использоваться предыдущие поставки, будем искать искомое содержание элемента в ферросплаве в день  $d$ , как среднее входных анализов, пришедших в определенный промежуток времени до дня  $d$ .

$$\tilde{f}_e^d = \frac{1}{N} \sum_{k \in [d-\delta_1, d-\delta_2]} f_e^k, \quad (6.1)$$

где  $N$  - количество входных анализов ферросплава  $f$ , пришедших за промежуток времени  $[d - \delta_1, d - \delta_2]$ .

Параметры  $\delta_1, \delta_2$  можно найти из минимизации среднеквадратичной ошибки RMSE:

$$\sum_{x \in X} \left( \alpha \sum_{f \in F} \tilde{f}_e^d(\delta_1, \delta_2) - y_e \right)^2 \longrightarrow \min_{\alpha, \delta_1, \delta_2} \quad (6.2)$$

Данная задача решалась следующим образом:

- Фиксировались все возможные пары  $\delta_1 \in [10, 150], \delta_2 \in [0, 10]$   
Предполагалось, что пришедший ферросплав начинает использоваться не позднее, чем через 10 дней, а партия ферросплавов заканчивается по истечении менее 150 дней.
- Для каждой пары  $(\delta_1, \delta_2)$  находился минимум функционала 6.2.

По сути, этот алгоритм делает следующее: для каждой возможной пары  $(\delta_1, \delta_2)$  строится обучающая выборка  $(x_e, y_e)$ , где  $x_e$  - количество добавленного элемента в соответствии с содержаниями элементов, полученными из 6.1,  $y_e$  - финальный химический анализ стали. Затем строится линейная регрессия  $y_e(x_e)$  и считается среднеквадратичная ошибка, минимум которой мы стремимся достичь.

Полученные результаты приведены на рис 6.1, здесь  $\delta_2 = 3$ ,  $\delta_1$  - меняется. Делаем вывод, что наилучший RMSE достигается при  $\delta_1 = 90$  дней,  $\delta_2 = 3$  дня. Эти результаты соответствовали экспертным ожиданиям.

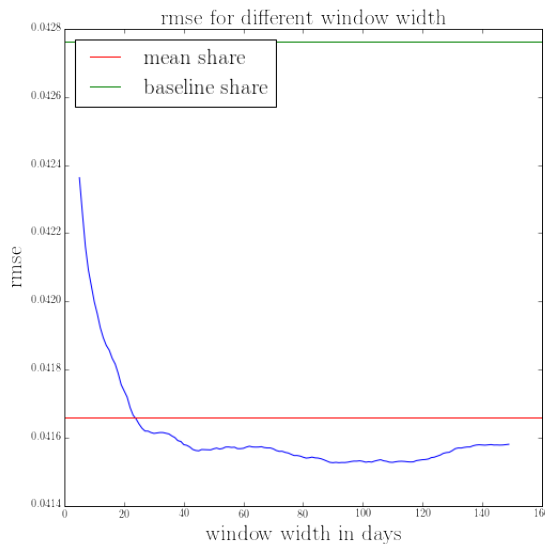


Рис. 6.1:  $RMSE$  для различных  $\delta_1$  при  $\delta_2 = 3$

## 6.2 Фильтрация данных и отбор признаков

В качестве исходных данных для задачи регрессии использовались исторические данные о плавках стали. Их общее количество составляло около 120000. При этом большая часть данных содержала какие-то шумные или неполные данные, а в некоторых случаях даже ручные правки. Требовалось отфильтровать эти данные перед обучением моделей. Первым этапом фильтрации было выбрасывание плавков, содержащих неадекватные значения важных параметров. Определение неадекватности, как и определение важного параметра, согласовывалось с экспертами.

Пример полученных после первого этапа данных можно увидеть на рис. 6.2. Далее данные были отфильтрованы, исходя из физических соображений:

- 1 усвоение любого элемента не может быть более 100%
- 2 усвоение любого элемента не может быть менее 50%

После всех проведенных фильтраций осталось около 70000 плавков. Оставшиеся данные приобрели вид, представленный на рис. 6.3

В силу того, что у нас есть априорные физические соображения о том, какие именно признаки должны влиять на модель в большей степени, надо проверять обученные модели на наиболее важные для них признаки,

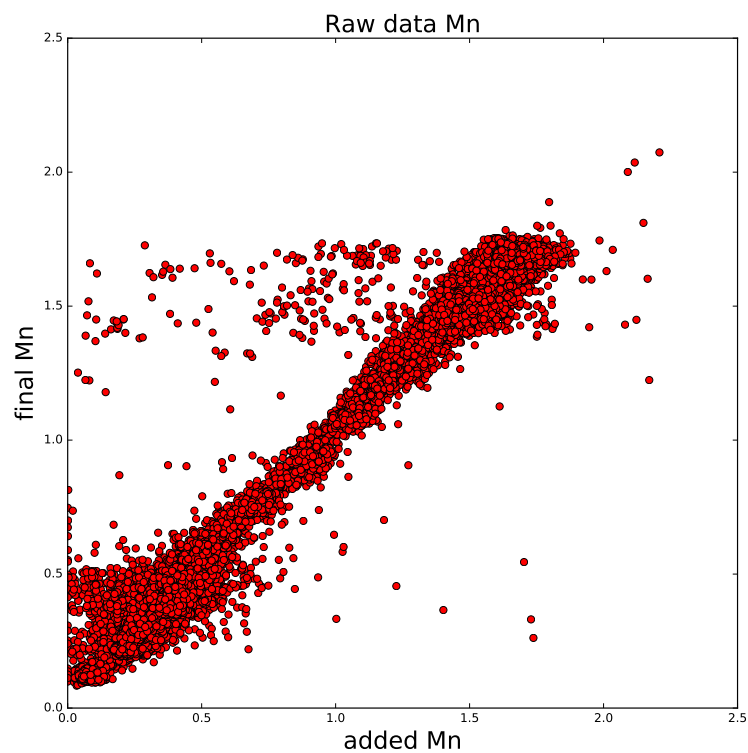


Рис. 6.2: Неотфильтрованные данные по марганцу

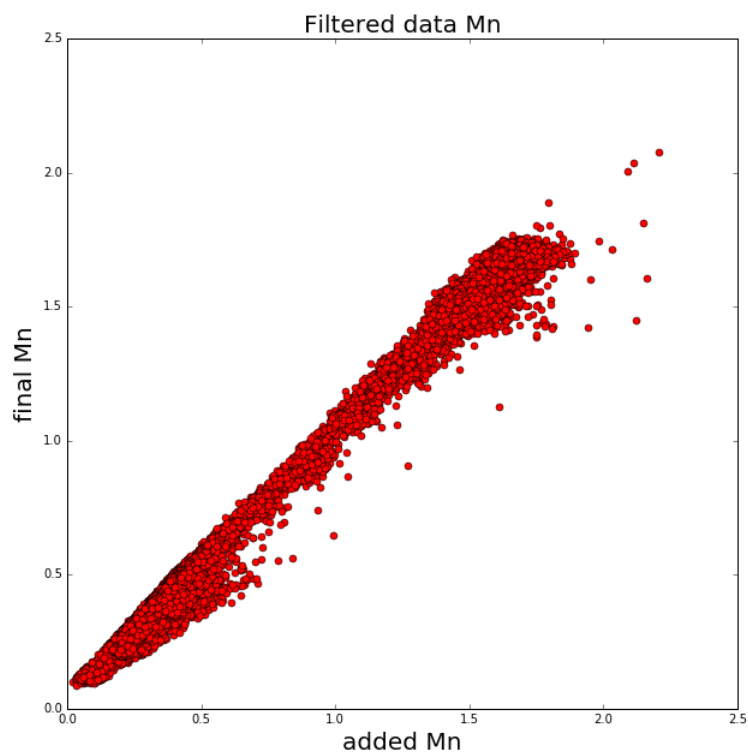


Рис. 6.3: Отфильтрованные данные по марганцу



а также контролировать наличие косвенной зависимости от неизвестных нам признаков.

В частности, заметим, что в перечисленном выше списке признаков отсутствуют признаки, отражающие добавления непосредственно самих ферросплавов. Это объясняется несколькими причинами:

- 1 влияние конкретного ферросплава на плавку обусловлено только его химическим составом
- 2 Различные ферросплавы могут быть практически эквивалентны по химическому составу (то есть взаимозаменяемы в плавке), но при этом имеют разные названия.
- 3 модель, обученная на добавлениях ферросплавов, будет скорее всего по-разному учитывать различные ферросплавы, пусть и с одинаковым составом, причина - исторические добавления.
- 4 обученная модель может применяться в новых плавках, в которых присутствуют ферросплавы, никогда не добавлявшиеся в истории (появилась новая модель ферросплава)

Кроме этого, для моделей, в которых применялся MatrixNet, использовался алгоритм жадного обтора признаков, при этом, для ускорения, итерации проводились по отсортированному по feature importance [15] массиву  $\vec{x} = (x_1, \dots, x_N)$ :

```

1  Инициализируем лучшее качество  $s^* := rmse(y(\vec{x}))$ .
2  Инициализируем набор выброшенных признаков  $D$ .
3  Инициализируем переменную окончания отбора  $repeat := True$ 
4  до тех пор, пока  $repeat$  выполнять
5  |    $repeat := False$ 
6  |   для каждого  $x \in (\vec{x}/D)$  выполнять
7  |   |   Считаем качество модели, обученной без признака  $x$ :
8  |   |   |    $s := rmse(y(\vec{x}/(D \cup x)))$ 
9  |   |   |   если  $s \leq s^*$  тогда
10 |   |   |   |    $s^* := s$ 
11 |   |   |   |    $D := D \cup x$ 
12 |   |   |   |    $repeat := True$ 
13 |   |   конец
14 конец

```

### Алгоритм 1: Жадный алгоритм отбора признаков

Полученные модели проверялись на косвенные зависимости. Это делалось для увеличения интерпретируемости модели.

Например, модель по углероду имела следующие значения важностей признаков:

1. Масса добавленного фосфора
2. Масса добавленной серы
- ...
11. Масса добавленного углерода

Здесь, как мы видим, признак, который должен быть самым важным, и по которому, собственно, будет идти оптимизация, находится на 11 месте, что говорит нам о том, что полученная модель работает неправильно. Признаки, расположенные выше 11-й позиции можно убрать из модели.

## 6.3 Обучение моделей

### Настройка линейных моделей и MatrixNet

Задача предсказания финального химического состава стали решалась отдельно по каждому элементу. То есть предполагалось, что зависимость содержания конкретного элемента в готовой стали сосредоточена в вычисленных признаках и не обусловлена содержанием других элементов.

- **Линейная регрессия.** Линейная регрессия строилась всего по одному признаку: суммарное количество добавленного элемента. Для каждой модели (каждого элемента) этот признак был свой.
- **Линейная регрессия + MatrixNet.** Результат линейной регрессии вычитался из целевого значения, после чего MatrixNet обучался на всех признаках.
- **Кусочно-линейная регрессия + MatrixNet.** Кусочно-линейная регрессия строилась в пространстве двух признаков: добавленный марганец и добавленный кремний. При этом применялась она также только для этих двух элементов. Триангуляция строилась жадным образом из прямоугольной сетки, которая, в свою очередь, задавалась вручную.

Для остальных элементов строилась линейная регрессия.

### Построение нейросетей

В работе было рассмотрено несколько способов построения нейросетей с различными архитектурами.

- Нейросеть с одним скрытым слоем
- Нейросеть с двумя скрытыми слоями
- Нейросеть с несколькими выходами

- Нейросеть с разделением признаков на две группы: соответствующих добавлениям ферросплавов и остальных признаков

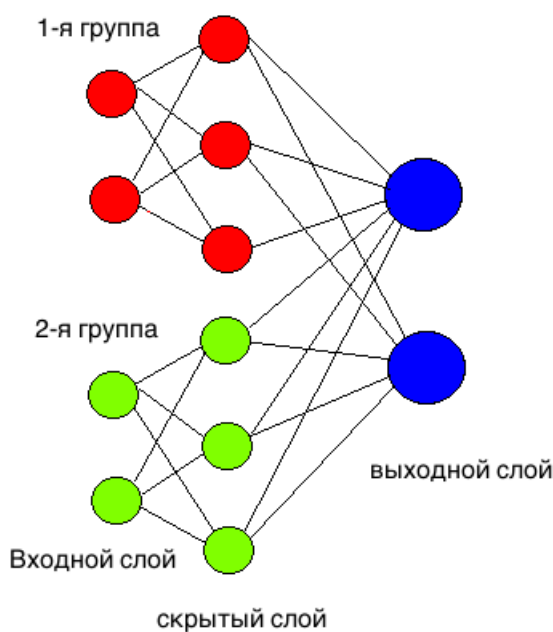


Рис. 6.4: Нейросеть с разделением признаков на 2 группы

Последняя нейросеть схематически изображена на рис. 6.4. Стоит заметить, что количество скрытых слоев необязательно должно быть равно одному. На самом деле, нейросеть с разделением признаков - это просто частный случай нейросети без разделения, в которой убраны связи между двумя группами нейронов (на рис. 6.4 - красная и зеленая группы). Перед обучением нейросети все признаки были стандартизованы, так как наилучший шаг обучения сильно зависит от порядка значений входных данных. Если исходные признаки имеют различный порядок, то их лучше стандартизовать, чтобы уменьшить вероятность попадания процесса оптимизации весов в локальный минимум.

## 7. Сравнение моделей

Мы будем сравнивать полученные модели в историческом порядке. Начнем с модели, которая тестировалась первой - MatrixNet. Причины для использования MatrixNet было несколько:

1. градиентный бустинг над решающими деревьями является одним из наиболее универсальных алгоритмов
2. решающие деревья умеют работать с закодированными категориальными признаками
3. прост в использовании и реализации

В итоге, получено качество, приведенное в таблице 7.1. Разные этапы в данной таблице соответствуют разным этапам производства стали, которые, применительно к нашей задаче, характеризуются вариативными наборами статических признаков.

Предложенная модель обладала одним большим недостатком. При ее использовании в некоторых плавках, в условии добавления большего количества ферросплавов, наблюдалось уменьшение предсказания химического состава, чего на самом деле быть не должно. Одной из причин этого могло быть отсутствие достаточного количества данных в области, в которой происходила оптимизация (а иногда и полное отсутствие данных).

На рис. 7.1 цветом показано предсказание модели. Из рисунка отчетливо видно, что наша модель кусочно-постоянная (разбита на квадраты), что противоречит физичности модели. Кроме того, на графике показано, что при добавлении кремния мы получаем значительно больше марганца, чем если бы мы добавляли марганец. Это снова делает нашу модель неинтерпретируемой.

Таблица 7.1: Качество MatrixNet

Этап	Элемент	SD	RMSE	$\frac{RMSE}{SD}$
FINAL	Cr	0.27315%	0.03678%	0.13466%
FINAL	Mn	0.47819%	0.02402%	0.05024%
FINAL	Mo	0.02048%	0.00960%	0.46845%
FINAL	Nb	0.01557%	0.00268%	0.17232%
FINAL	Ni	0.20563%	0.00973%	0.04733%
FINAL	Si	0.20438%	0.02314%	0.11320%
FINAL	V	0.01579%	0.00254%	0.16074%
INTERMEDIATE	Cr	0.24862%	0.01770%	0.07121%
INTERMEDIATE	Mn	0.47840%	0.02456%	0.05133%
INTERMEDIATE	Mo	0.02063%	0.01169%	0.56674%
INTERMEDIATE	Nb	0.01555%	0.00246%	0.15835%
INTERMEDIATE	Ni	0.25851%	0.04271%	0.16523%
INTERMEDIATE	Si	0.20985%	0.02197%	0.10471%
INTERMEDIATE	V	0.01605%	0.00286%	0.17788%
REFINING	Cr	0.26131%	0.03324%	0.12722%
REFINING	Mn	0.04197%	0.02045%	0.48723%
REFINING	Mo	0.02452%	0.00837%	0.34132%
REFINING	Nb	0.01570%	0.00273%	0.17413%
REFINING	Ni	0.19509%	0.00887%	0.04546%
REFINING	Si	0.03249%	0.01745%	0.53695%
REFINING	V	0.01543%	0.00261%	0.16929%
ROUGH	Cr	0.24870%	0.01897%	0.07628%
ROUGH	Mn	0.46141%	0.02566%	0.05562%
ROUGH	Mo	0.02326%	0.01172%	0.50379%
ROUGH	Nb	0.01523%	0.00251%	0.16498%
ROUGH	Ni	0.22992%	0.01776%	0.07723%
ROUGH	Si	0.20922%	0.02177%	0.10406%
ROUGH	V	0.01567%	0.00231%	0.14750%

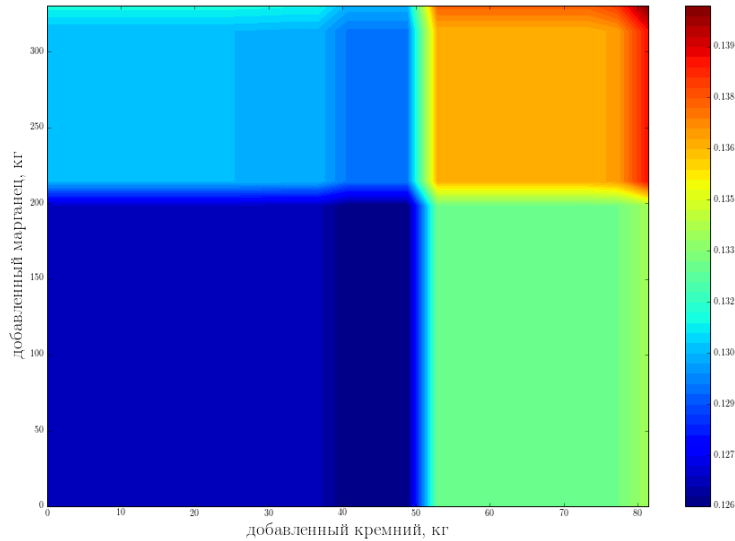


Рис. 7.1: Предсказания MatrixNet для стали с низким содержанием марганца

Как можно заметить из утверждения 5.1, градиентный бустинг в таком случае дает плохие результаты. Для их устранения было решено опробовать алгоритм "Линейная регрессия + MatrixNet".

Основными предпосылками к использованию данной модели являлись суждения:

- Сделать простую и интерпретируемую модель, удовлетворяющую физическим свойствам
- Аппроксимировать остатки предыдущей модели с помощью MatrixNet

Самой простой и интерпретируемой моделью, которую было решено использовать, оказалась линейная регрессия. Качество, полученное алгоритмом "линейная регрессия + MatrixNet" приведено в таблице 7.2

Обратим внимание, качество немного улучшилось, но мы также ожидаем, что модель станет более интерпретируемой - в областях, не покрытых историческими данными линейная регрессия будет регуляризовать предсказания матрикснета. На рис. 7.2 показано, что, в отличии от чистой модели MatrixNet, данная модель является более интерпретируемой. Это можно объяснить недостатком данных в области с низким содержанием марганца.

Таблица 7.2: Качество LR + MatrixNet

Этап	Элемент	SD	RMSE	$\frac{RMSE}{SD}$
FINAL	Cr	0.01912%	0.01810%	0.94671%
FINAL	Mn	0.03966%	0.02516%	0.63454%
FINAL	Mo	0.01211%	0.00932%	0.76925%
FINAL	Nb	0.00446%	0.00262%	0.58737%
FINAL	Ni	0.01672%	0.01085%	0.64888%
FINAL	Si	0.03894%	0.02252%	0.57825%
FINAL	V	0.00402%	0.00260%	0.64830%
INTERMEDIATE	Cr	0.01708%	0.01662%	0.97262%
INTERMEDIATE	Mn	0.03825%	0.02451%	0.64065%
INTERMEDIATE	Mo	0.01231%	0.01129%	0.91741%
INTERMEDIATE	Nb	0.00429%	0.00257%	0.59879%
INTERMEDIATE	Ni	0.01720%	0.01586%	0.92213%
INTERMEDIATE	Si	0.03921%	0.02160%	0.55073%
INTERMEDIATE	V	0.00407%	0.00265%	0.65286%
REFINING	Cr	0.01946%	0.01729%	0.88831%
REFINING	Mn	0.02730%	0.02033%	0.74463%
REFINING	Mo	0.01215%	0.00838%	0.68932%
REFINING	Nb	0.00440%	0.00272%	0.61747%
REFINING	Ni	0.01647%	0.00893%	0.54185%
REFINING	Si	0.02147%	0.01726%	0.80402%
REFINING	V	0.00419%	0.00273%	0.65059%
ROUGH	Cr	0.01752%	0.01747%	0.99730%
ROUGH	Mn	0.03929%	0.02538%	0.64590%
ROUGH	Mo	0.01272%	0.01135%	0.89173%
ROUGH	Nb	0.00451%	0.00254%	0.56314%
ROUGH	Ni	0.01698%	0.01540%	0.90693%
ROUGH	Si	0.03889%	0.02128%	0.54721%
ROUGH	V	0.00391%	0.00258%	0.66080%



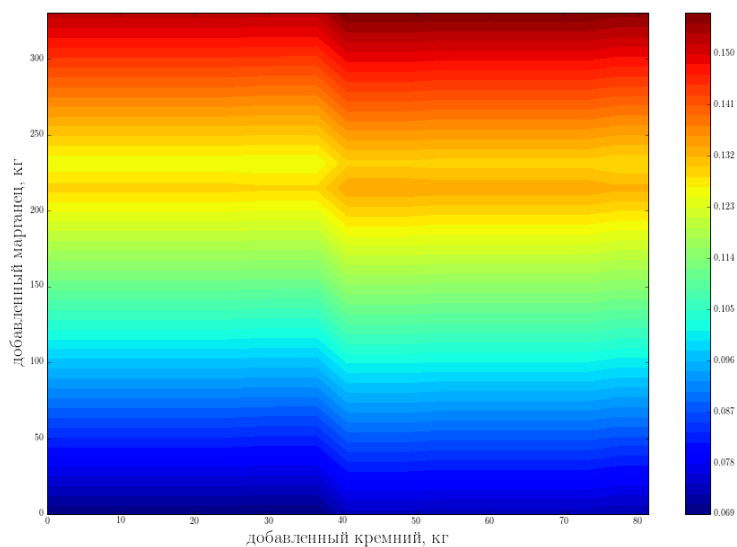
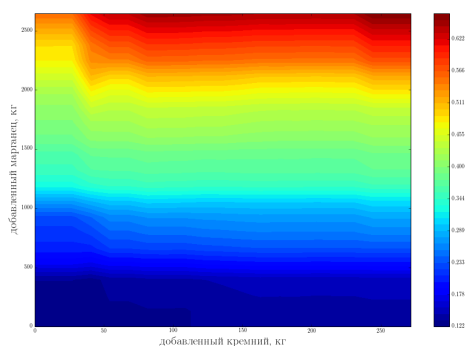
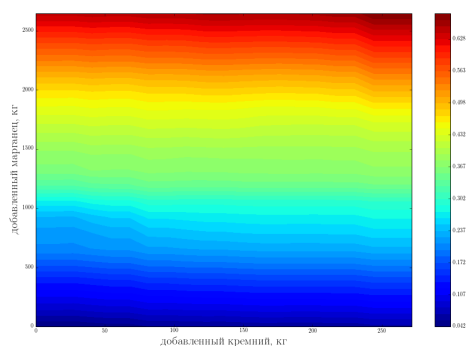


Рис. 7.2: Предсказания LR + MatrixNet для стали с низким содержанием марганца



(a) MatrixNet



(b) MatrixNet + LR

Рис. 7.3: Иллюстрация предсказаний моделей MatrixNet и MatrixNet + LR на конкретной плавке, цветом показано предсказание модели

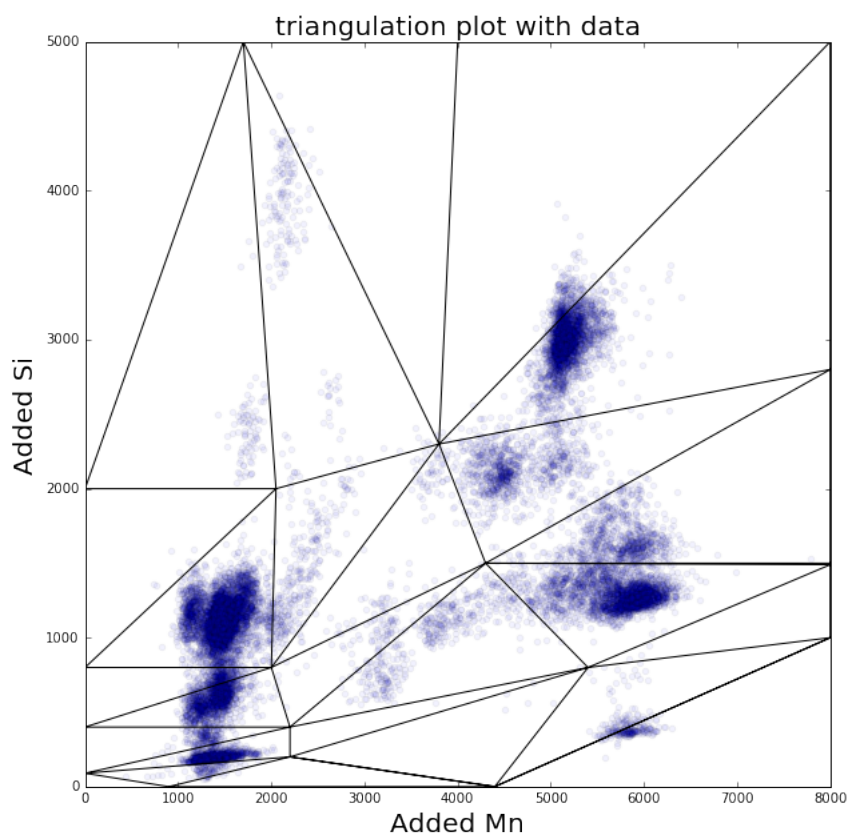


Рис. 7.4: Триангуляция обучающей выборки в пространстве добавленных марганца и кремния.

Сравнивая модели на более высоколегированной стали (рис. 7.3), заметим, что обе модели ведут себя достаточно хорошо, однако в MatrixNet по-прежнему возникают резкие скачки, которые свидетельствуют о нефизичности модели. У модели "Линейная регрессия + MatrixNet" эти скачки отсутствуют.

Можно сказать, что добавление линейной регрессии улучшило качество и интерпретируемость модели.

Полученную линейную регрессию мы усложнили, превратив ее в кусочно-линейную. Учитывая зависимость усвоения марганца от количества добавленного кремния, эта модель строилась в пространстве двух признаков - добавленный марганец и кремний.

Обучив алгоритм "кусочно-линейная регрессия + матрикснет" получили качество, представленное в таблице 7.3. Оказалось, что качество текущей модели превосходит качество модели с линейной регрессией. При этом стоит заметить, что одним из главных недостатков полученной модели является высокая зависимость от построенной триангуляции.

Таблица 7.3: Качество PWLR + MatrixNet

Этап	Элемент	SD	RMSE	$\frac{RMSE}{SD}$
FINAL	Cr	0.01912%	0.01810%	0.94671%
FINAL	Mn	0.03966%	0.02405%	0.60640%
FINAL	Mo	0.01211%	0.00932%	0.76925%
FINAL	Nb	0.00446%	0.00262%	0.58737%
FINAL	Ni	0.01672%	0.01085%	0.64888%
FINAL	Si	0.03894%	0.02144%	0.555059%
FINAL	V	0.00402%	0.00260%	0.64830%
INTERMEDIATE	Cr	0.01708%	0.01662%	0.97262%
INTERMEDIATE	Mn	0.03825%	0.02420%	0.63268%
INTERMEDIATE	Mo	0.01231%	0.01129%	0.91741%
INTERMEDIATE	Nb	0.00429%	0.00257%	0.59879%
INTERMEDIATE	Ni	0.01720%	0.01586%	0.92213%
INTERMEDIATE	Si	0.03921%	0.02161%	0.55073%
INTERMEDIATE	V	0.00407%	0.00265%	0.65286%
REFINING	Cr	0.01946%	0.01729%	0.88831%
REFINING	Mn	0.02730%	0.02033%	0.74463%
REFINING	Mo	0.01215%	0.00838%	0.68932%
REFINING	Nb	0.00440%	0.00272%	0.61747%
REFINING	Ni	0.01647%	0.00893%	0.54185%
REFINING	Si	0.02147%	0.01726%	0.80402%
REFINING	V	0.00419%	0.00273%	0.65059%
ROUGH	Cr	0.01752%	0.01747%	0.99730%
ROUGH	Mn	0.03929%	0.02478%	0.63070%
ROUGH	Mo	0.01272%	0.01135%	0.89173%
ROUGH	Nb	0.00451%	0.00254%	0.56314%
ROUGH	Ni	0.01698%	0.01540%	0.90693%
ROUGH	Si	0.03889%	0.02133%	0.54847%
ROUGH	V	0.00391%	0.00258%	0.66080%

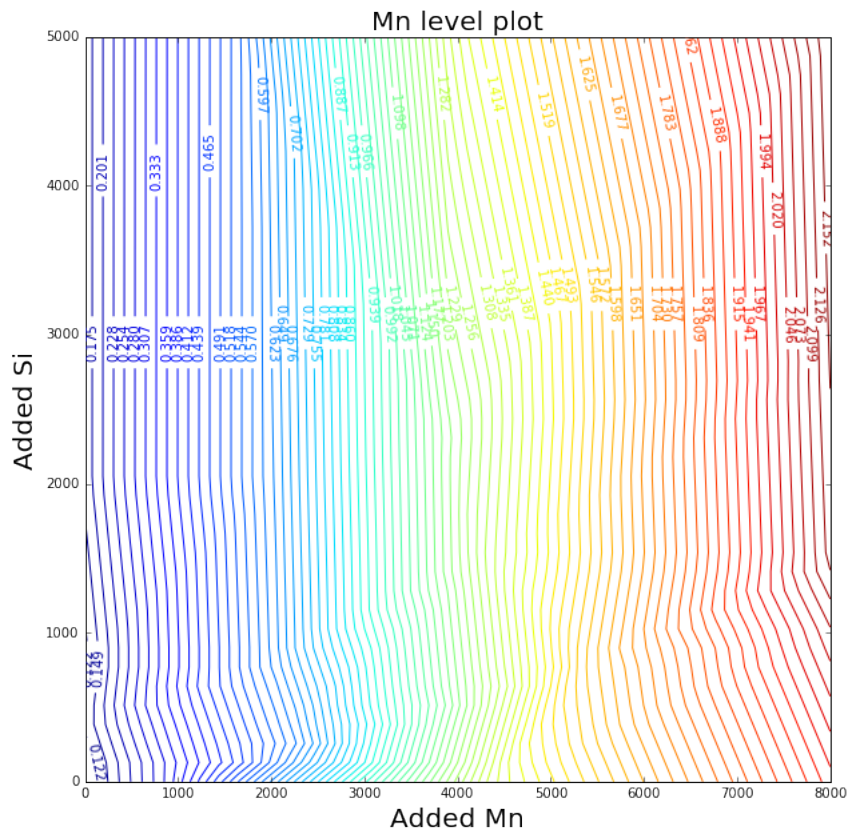


Рис. 7.5: Линии уровня кусочно-линейной модели, предсказывающей содержание марганца в готовой стали

Таблица 7.4: Сравнение всех алгоритмов

Элемент	MNet	MNet + LR	MNet + PWLR	Нейросеть
Mn	0.02566%	0.02538%	0.02478%	0.03073%
Si	0.02177%	0.02128%	0.02133%	0.0248%

Кусочно-линейная регрессия имела вид, показанный на рис. 7.5, а построенная жадным алгоритмом триангуляция изображена на рис. 7.4. Последний рисунок хорошо отражает высокую интерпретируемость модели.

Были предприняты попытки построить различные нейросети, при этом лучшим вариантом оказалась следующая нейросеть:

- Два скрытых слоя.
- Одновременно предсказывает марганец и кремний.

Результаты оказались значительно хуже, чем у аналогичных моделей с MatrixNet. Итоги сравнения на одном этапе указаны в таблице 7.4.

Лучшие результаты как по  $RMSE$ , так и по интерпретируемости, показала модель "кусочно-линейная регрессия + MatrixNet".

## 8. Заключение

В данной работе рассмотрена задача построения предсказательной модели химического состава стали. Предложен двухэтапный подход к построению регрессионной модели. В качестве модели первого уровня рассматривалась кусочно-линейная регрессия, а в качестве модели второго уровня рассматривался MatrixNet - реализация градиентного бустинга над решающими деревьями.

Можно выделить следующие преимущества подхода:

- Высокая интерпретируемость модели.
- Превосходство данного подхода над классическими алгоритмами.
- Интерпретируемое поведение модели в областях, не покрытых историческими данными.

Стоит отметить зависимость кусочно-линейной регрессии от триангуляции признакового пространства и использование выбранной экспертами сетки для её построения. В дальнейшем предлагается автоматизировать этот процесс - добавить возможность автоматического выбора сетки.

### **Результаты, выносимые на защиту:**

- Предложен двухэтапный подход к построению интерпретируемой регрессионной модели.
- Предложен метод построения признакового описания.
- Построена эффективная и интерпретируемая модель, предсказывающая химический состав стали.

# Список литературы

- [1] J.A. Nelder и R. Mead. “A simplex method for function minimization”. В: *Computer Journal* 7 (1965), с. 308—313.
- [2] Donald R. Jones, Matthias Schonlau и William J. Welch. “Efficient Global Optimization of Expensive Black-Box Functions”. В: *Journal of Global Optimization* 13 (1998), с. 455—492.
- [3] Jun Zhang. “Optimal Control Problem of Converter Steelmaking Production Process Based on Operation Optimization Method”. В: *Discrete Dynamics in Nature and Society* 2015 (2015).
- [4] Yao Zhao Min Han. “Dynamic control model of BOF steelmaking process based on ANFIS and robust relevance vector machine”. В: *Expert Systems with Applications: An International Journal archive* 38 (2011), с. 14786—14798.
- [5] N. Dogan, G.A. Brooks и M.A. Rhamdhani. “Comprehensive Model of Oxygen Steel-Making”. В: *ISIJ International* 51 (2011), с. 1086—1109.
- [6] С.В. Коминов, С.С. Бакума и Туркин С.О. “Математическая модель расчета основных технологических параметров управления мартеповской плавки”. В: *Известия вузов* 5 (1999), с. 19—21.
- [7] Д.В. Гулыга и А.В. Сущенко. “Модель расчета параметров кислородно-конвертерной плавки”. В: *Сталь* 12 (2003), с. 19—24.
- [8] А.А. Гусев и Б.Н. Огороков. “Математическая модель заключительного этапа продувки конвертерного процесса”. В: *Известия вузов. Черная металлургия* 2 (2000), с. 49—57.
- [9] K. Randerson, H.C. Carey и K. Schomberg. “A model for calculating solubility hydrogen in steel”. В: *Ironmaking and Steelmaking* 30 (2003), с. 369—378.

- [10] JH Friedman. “Stochastic gradient boosting”. В: *Computational Statistics & Data Analysis* 38 (2002), с. 367—378.
- [11] J. Friedman. “Greedy function approximation: A gradient boosting machine.” В: *Annals of Statistics* 29 (2000), с. 1189—1232.
- [12] A. Gulin. *Matrixnet*. Tex. отч. <http://www.slideshare.net/yandex/matrixnet>, 2012.
- [13] Alejandro Toriello и Juan Pablo Vielma. “Fitting piecewise linear continuous functions”. В: *European Journal of Operational Research* 219 (2012), с. 86—95.
- [14] Martin Riedmiller. “Advanced supervised learning in multi-layer perceptrons — From backpropagation to adaptive learning algorithms”. В: *Computer Standards & Interfaces* 16 (1994), с. 265—278.
- [15] L Breiman и др. “Classification and Regression Trees”. В: (1984).