УДК 681.5

ЧИСЛЕННЫЙ МЕТОД ПОСТРОЕНИЯ МОДЕЛЕЙ ДЛЯ ПРОГНОЗИРОВАНИЯ ЭКОНОМИЧЕСКИХ ПОКАЗАТЕЛЕЙ СЕЛЬСКОГО ХОЗЯЙСТВА

Бийбосунов Б.И.
д. физ-мат. наук, профессор
Кыргызский государственный университет имени И. Арабаева
Бишкек, Кыргызстан

Сабитов Б.Р.
к. ф.-м. н., доцент
Кыргызский национальный университет имени Ж. Баласагына
Бишкек, Кыргызстан

Алмасбекова З.
Старший преподаватель
Кыргызский национальный университет имени Ж. Баласагына
Бишкек, Кыргызстан

Эсенаманова Г.К.
Старший преподаватель
Кыргызский национальный университет имени Ж. Баласагына
Бишкек, Кыргызстан

Аннотация: В настоящей статье изучается применение технологии машинного обучения для задач сельского хозяйства. Рассмотрены технологии построения моделей в растениеводстве с применением регрессионного анализа. Изучены два подхода к построению моделей с линейными и нелинейными данными с применением градиентного метода оптимизации. Создан пакет программ для различных моделей АПК с применением различных технологий машинного обучения. Используя линейные и нелинейные данные построены модели в растениеводстве и животноводстве. Рассмотрена оптимизационная задача применительно к задачам АПК. Изучены варианты самого общего алгоритма -метода градиентного спуска для построения моделей для задач сельского хозяйства, в частности построены модели и прогнозы по реализации сельхозпродукций с использованием регрессионного анализа. Создан пакет программ с использованием интеллектуальных систем машинного обучения по технологиям Python с целью прогнозирования продукции растениеводства и животноводства. Данная методика позволяет изучать и строить модели для широкого круга задач в сфере АПК, рассматриваются сложные нелинейные данные, которые описывают количество реализованной продукции в отраслях растениеводства и животноводства и строятся модели на основе регрессионного анализа. В частности, рассмотрены сложные нелинейные модели реализации поголовья скота в животноводстве, зерновых и зерно-бобовых в растениеводстве с использованием различных моделей.

Ключевые слова: нелинейные данные, АПК (агропромышленный комплекс), мультиколлинеарность, численный метод, стохастический градиентный спуск, регрессионный анализ**,** прогнозирование, машинное обучение

NUMERICAL METHOD OF MODELLING FOR FORECASTING OF ECONOMIC INDICATORS IN AGRICULTURAL SECTOR

Biibosunov B.I.
Doctor of Physics&Math, Professor
Kyrgyz State University named after I. Arabaev
Bishkek, Kyrgyzstan

Sabitov B.R.
Candidate of Physics&Math, Associated Professor
Kyrgyz National University named after J. Balasagyn
Biskek, Kyrgyzstan

Almasbekova Z.
Senior Lecturer
Kyrgyz National University named after J. Balasagyn
Biskek, Kyrgyzstan

Esenamanova G.K.
Senior Lecturer
Kyrgyz National University named after J. Balasagyn
Biskek, Kyrgyzstan

Abstract: Usage of machine learning for the goals of agricultural sector is researched in the current article. There have been reviewed technologies of modelling in planting by means of regression analysis. There are reviewed also approaches to modelling with linear and non-linear data by means of gradient optimization method. A software package has been created for various models of agribusiness using various machine learning technologies. Using linear and non-linear data, models in crop production and stockbreeding are constructed. The optimization problem is considered in relation to the problems of agriculture. There are studied variants of the most general algorithm, which are as the gradient descent method for building models for agricultural tasks; in particular, there are constructed models and forecasts for the sale of agricultural products using regression analysis. A software package was created by using intelligent machine learning systems based on Python technologies. This method allows us to study and build models for a wide range of tasks in the agricultural sector. We consider complex nonlinear data that describe the number of products sold in the crop production and livestock industries and can build their model based on regression analysis. In particular, we consider complex nonlinear models for the implementation of livestock in husbandry, cereals and legumes in crop production by using different models.

Key words: non-linear data, AIC (agro-industrial complex), multicollinearity, numerical method, stochastic gradient descent, regression analysis, forecasting.

Введение

В настоящее время бурно развивается применение новых информационных технологий, а именно технологии машинного обучения для задач прикладного характера [1]. В машинном обучении имеется множество алгоритмов построения различных моделей. Особенно ярко выражено применение данных технологий в различных задачах экономики, науки, образования и промышленности. Большинство статистических экономических моделей опирается на параметрические методы построения моделей. В таких моделях соотношение между результатом и входными данными выражается через простые фиксированные уравнения. Данные применяются для определения оптимальных значений неизвестных частей уравнения. В эту категорию моделей входят линейная, логистическая и нелинейная регрессии. [2]

Цель исследования

В данной статье мы исследуем задачи прогнозирования с применением регрессионного анализа для задач сельского хозяйства. Рассмотрим задачу прогнозирования количества реализованной продукции или объемов реализации продукции в отрасли растениеводства с применением технологий машинного обучения. Модели будем обучать с применением регрессионного анализа. Данная технология для многих задач АПК используется для анализа и прогнозирования моделей. Создадим пакет программ для различных моделей АПК с применением различных технологий машинного обучения. Используя линейные и нелинейные данные построены модели в растениеводстве и животноводстве. Обычно параметры моделей определяются с использованием функции издержек, которые определяется функционалом:

$$\mathrm{J}(\mathrm{X},h_\theta)=\frac{1}{m}\sum_{i=1}^{m}(\theta^T * x^{(i)} - y^{(i)})^2$$

Необходимо правильно выбрать алгоритм обучения модели и оптимальное значение параметров для изучаемой задачи.

Материал и методы исследования

Изучим две различные технологии обучения моделей. Первое, применение прямого уравнения в виде нормального уравнения, из которого можно определить параметры модели.

Второй подход к применению численного подхода к вычислению параметров модели с применением градиентного метода оптимизации. Метод градиентного спуска постепенно корректирует параметры модели, для нахождения минимума функционала. Рассмотрим несколько вариантов градиентного спуска. Для задач растениеводства и животноводства можно применять оба метода. Изучим конкретные примеры по построению моделей по реализации продукций в животноводстве и растениеводстве.

А именно рассматриваются сложные нелинейные данные, которые описывают количество реализованной продукции в отраслях растениеводства и животноводства и построим их модель на основе регрессионного анализа. В частности, рассмотрены сложные нелинейные модели реализации поголовья скота в животноводстве, зерновых и зерново-бобовых в растениеводстве с использованием различных моделей.

Рассмотрим сначала первый подход к построению моделей в АПК. Изучим следующую задачу оптимизации:

$$\min_{\theta} \mathrm{J}(\mathrm{X}, h_{\theta}) = \min_{\theta} \frac{1}{m}\sum_{i=1}^{m}(\theta^{T} * x^{(i)} - y^{(i)})^{2}$$

Параметры модели θ можно определить с помощью прямого уравнения в виде нормального уравнения. Найденное значение θ, которое сводит к минимуму функцию издержек, записывается в аналитическом виде, то есть, математическое уравнение, дающее результат напрямую в виде формулы. Оно называется нормальным уравнением и представляется в виде решения:

$$\hat{\theta}=(X^{T} * X)^{-1}X^{T}y$$

Для тестирования, на линейную зависимость или мультиколлинеарность можно использовать число обусловленности матрицы $X^T X$. Число обусловленности равно отношению большего собственного числа к меньшему. Большое число определенности или наличие близких к нулю собственных чисел является признаком мультиколлинеарности.

К сожалению инвертирование, т.е. нахождение обратной матрицы , матрицы $X^T X$ при нечеткой мультиколлинеарности численно нестабильно, но существует решение. Известно также, что полноранговую матрицу X размера $n \times m$ можно представить в виде:

$$X=QR,$$

где

$Q^T Q = E$

R треугольная матрица размера $m \times m$.

При таком подходе из необходимого условия минимума функционала:

$$\frac{\partial}{\partial \theta} \mathrm{J}(\mathrm{X}, h_\theta)=0,$$

следует, что

$$\theta = R^{-1} Q^T \mathrm{y}$$

Для реализации данного подхода к задачам, например, по реализации молочной продукции в животноводстве, где нам больше подходит линейная функция y(t)=A*t+B, применение нормального уравнения (3) дает следующие результаты прогноза:

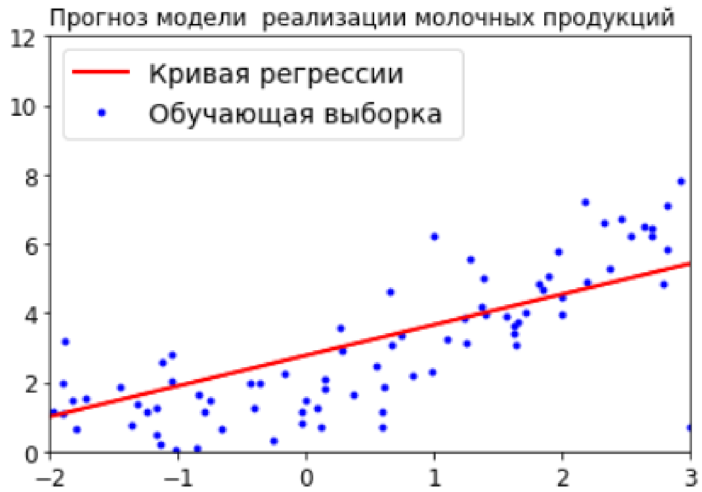

Рис. 1. Прогноз модели реализации молока

А для урожайности пшеницы более удобным для нас является полиномиальная функция, например, вида y = 0.5 * X**2 + X + 2

Например, по реализации льняного семени по функции вида $y = 0.25x^2 + 4x + 0.25$ прогнозная кривая имеет следующий вид:

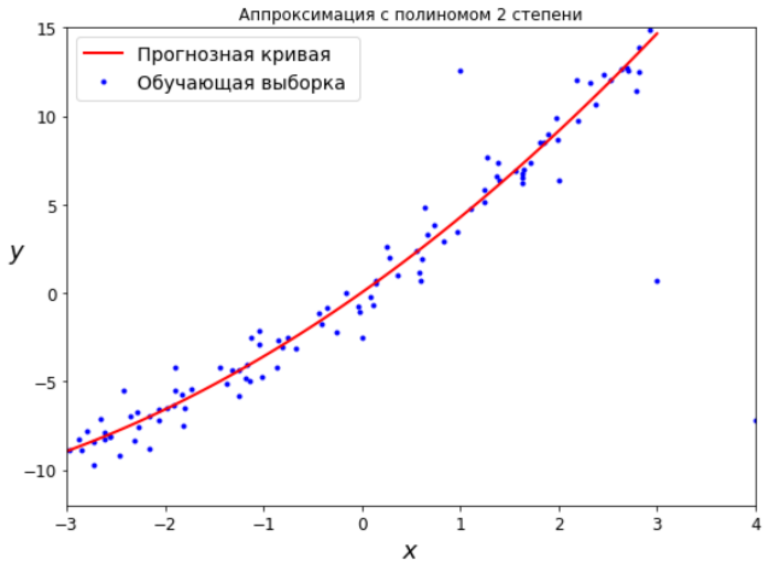

Рис. 2. Аппроксимация с полиномом 2 степени

Более сложные случаи встречаются например, при реализации продукций в животноводстве. Наиболее подходящей в данном случае является логистическая функция вида y(t)=A/(1+exp(B-C*t)), графическое изображение которой с шумом имеет вид:

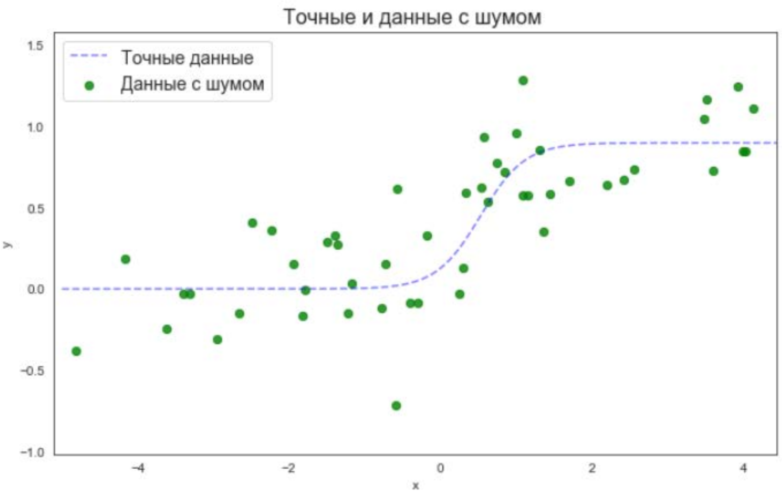

Рис. 3. Точные и данные с шумом

В этом случае применение технологией линейной регрессии не дает утешительных результатов. Вот результат применения прогноза, полученный регрессией:

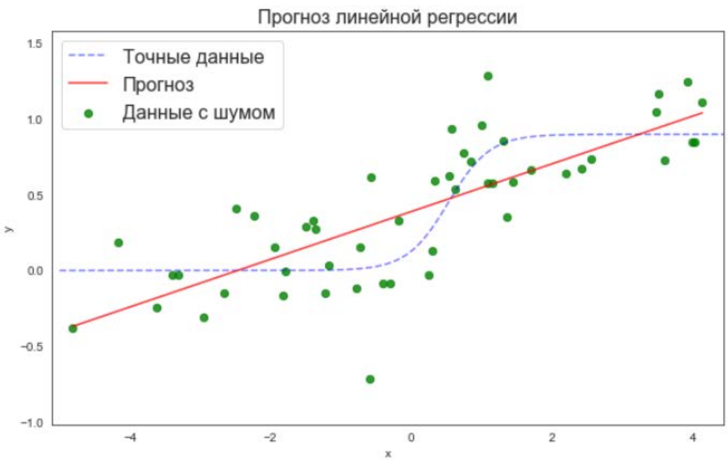

Рис. 4. Прогноз линейной регрессии

В таких случаях необходимо применять технологии нелинейного моделирования машинного обучения или применяется метод регуляризации [3] А.Н.Тихонова.

Второй общий подход к применению - это численный подход к вычислению параметров модели с применением градиентного метода оптимизации [4]. Изучается несколько вариантов численного подхода к вычислению минимума функционала.

Метод градиентного спуска для задач оптимизации представляет собой самый общий алгоритм оптимизации, способный находить оптимальные решения широкого диапазона задач [5,6]. Основная идея градиентного спуска заключается в том, чтобы итеративно подстраивать параметры для сведения к минимуму функции издержек, производная которой определяется по формуле:

$$\frac{\partial}{\partial\theta_j}J(\theta)=\frac{2}{m}\sum_{i=1}^{m}(\theta^T * x^{(i)} - y^{(i)})\, x_j^{(i)}$$

или в векторной форме:

$$\nabla_\theta J(\theta)=(\frac{\partial}{\partial\theta_0}J(\theta),\frac{\partial}{\partial\theta_1}J(\theta),\dots\frac{\partial}{\partial\theta_n}J(\theta))^T = \frac{2}{m}X^T(X\theta - y)$$

Процесс реализации градиентного метода в данном случае определяется по следующей формуле.

$$\theta^{k+1} = \theta^k - \eta\nabla_\theta J(\theta^k)$$

Для реализации зерновых и зерново-бобовых культур и построения модели наиболее подходит в данном случае параболическая функция вида: $y = 0.25x^2 + x + 2$.

Результаты исследования и их обсуждение

Составим программу реализации данного численного метода с применением библиотек Python для различных параметров η

eta = 0.1

n_iterations = 1000

```
m = 100

theta = np.random.randn(2,1)

for iteration in range(n_iterations):
gradients = 2/m * X_b.T.dot(X_b.dot(theta) - y)
theta = theta - eta * gradients
theta
                                                        Out [15]:
array ([[4.21509616],
      [2.77011339]])
X_new_b.dot(theta)
                                                        Out [16]:
array([[4.21509616],
      [9.75532293]])
theta_path_bgd = []

def plot_gradient_descent(theta, eta, theta_path=None):
    m = len(X_b)
    plt.plot(X, y, "b.")
    n_iterations = 1000
    for iteration in range(n_iterations):
        if iteration < 10:
        y_predict = X_new_b.dot(theta)
        style = "b-" if iteration > 0 else "r--"
        plt.plot(X_new, y_predict, style)
        gradients = 2/m * X_b.T.dot(X_b.dot(theta) - y)
        theta = theta - eta * gradients
        if theta_path is not None:
        theta_path.append(theta)
```

Вот результаты данной реализации:

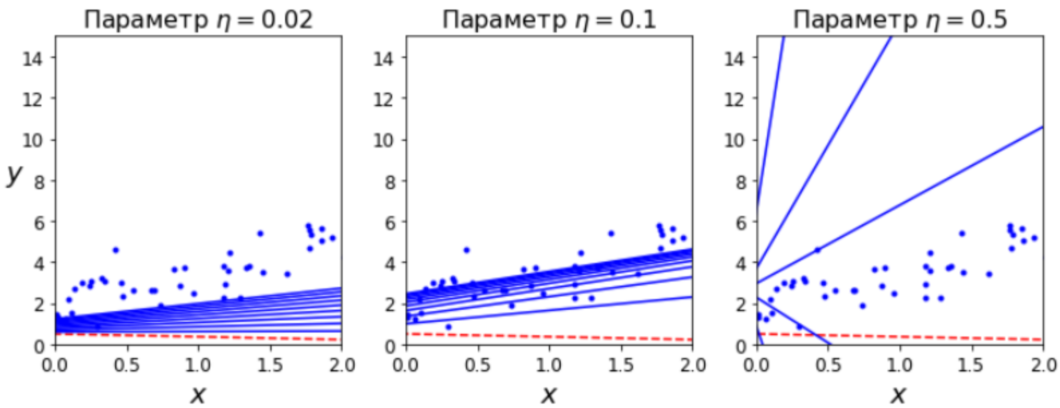

Рис. 5. Скорость обучения модели

Слева скорость обучения модели при $\eta = 0.02$ слишком низкая: в конце концов, алгоритм достигнет решения, но это займет долгое время. На рисунке, посредине при $\eta = 0.1$ скорость обучения выглядит довольно неплохо: алгоритм в этом случае сходится к решению всего за несколько итераций. А вот на рисунке справа при $\eta = 0.5$ скорость обучения очень высокая и наш алгоритм расходится, беспорядочно перескакивая с места на место и фактически с каждым шагом все больше удаляясь от исходного минимума функционала.

Стохастический градиентный спуск

Основная проблема использования пакетного градиентного спуска заключается в использовании полного обучающего набора для вычисления градиентов на каждом этапе, что в свою очередь замедляет его в случае крупного обучающего набора.

Выпишем теперь альтернативный алгоритм стохастического градиентного метода (**Stochastic Gradient Descent**). Ниже приведен алгоритм данного численного метода. Сначала введем входные данные:

1. Входные данные: X^i – это выборка, h- это скорость обучения, λ –так называемый темп забывания;
2. Выходные данные: вектор весов ω.

Выпишем сам алгоритм метода:

1. Необходимо задать значения ω_j, для j=0,…,n
2. Вычислить градиенты G_i=∇L_i (ω), i=1,…,m
3. Вычислить оценку функционала: $\tilde{Q}$=$\frac{1}{m}\sum_{i=1}^{m} L_i(\omega)$
4. Цикл по j
5. Выбираем объект x_i из X^i случайным образом
6. Вычислим потерю ε_i=$L_i(\omega)$
7. Далее необходимо вычислить градиент G_i=$\nabla L_i(\omega)$,
8. Вычислить шаг градиента по следующей формуле:

$$\omega = \omega - h\sum_{j=1}^{m} G_j$$

и наконец вычислить функционал:

$\tilde{Q}$=(1-λ) $\tilde{Q}$+$\lambda\varepsilon_i$,

до тех пор пока $\tilde{Q}$ или ω не сходятся. Как оценить функционал Q? После каждого шага вычисления ω по одному x_i, чтобы избежать оценки Q по всей выборке $x_1, \dots, x_n$, необходимо воспользоваться средней арифметической формулой по следующей рекуррентной формуле:

$$\tilde{Q}_m=\frac{1}{m}\sum_{i=1}^{m}\varepsilon_i,\ \tilde{Q}_m=(1-\frac{1}{m})\ \tilde{Q}_{m-1}+\frac{1}{m}\varepsilon_m$$

Как видно выше из алгоритма, стохастический градиентный спуск на каждом шаге просто выбирает из обучающего набора случайный образец и вычисляет градиенты на основе только этого единственного образца. Очевидно, алгоритм становится гораздо быстрее, т.к. на каждой операции ему приходится манипулировать совсем малым объемом данных. Также появляется возможность проводить обучение на гигантских обучающих наборах, потому что на каждой итерации в памяти должен находиться только один образ.

Мини-пакетный градиентный спуск

Теперь изучим применение алгоритма градиентного спуска, который называется мини-пакетным градиентным спуском (Mini-Batch Gradient Descent). Понять его довольно просто, т.к. вы уже знаете пакетный и стохастический градиентные спуски. На каждом этапе вместо вычисления градиентов на основе полного обучающего набора (как в пакетном градиентном спуске) или только одного образца (как в стохастическом градиентном спуске) мини-пакетный градиентный спуск вычисляет градиенты на небольших случайных наборах образцов, которые называются мини-пакетами (Mi1ii-Batch). Главное превосходство мини-пакетного градиентного спуска над стохастическим градиентным спуском в том, что вы можете получить подъем производительности от аппаратной оптимизации матричных операций, особенно когда используются графические процессоры. Продвижение этого алгоритма в пространстве параметров не так нерегулярно, как в случае SGD (Stochastic Gradient Descent), в особенности при довольно крупных мини пакетах.

В результате мини-пакетный градиентный спуск закончит блуждание чуть ближе к минимуму, чем SGD. Но с другой стороны ему может быть труднее уйти от локальных минимумов (в случае задач, которые страдают от локальных минимумов, в отличие от линейной регрессии, как мы видели ранее).

На рис. 6 ниже показаны пути, проходимые тремя алгоритмами градиентного спуска в пространстве параметров во время обучения. В итоге все они оказываются близко к минимуму, но путь пакетного градиентного спуска фактически останавливается на минимуме, тогда как стохастический и мини пакетный градиентные спуски продолжают двигаться около минимума.

Тем не менее, не забывайте о том, что пакетный градиентный спуск требует длительного времени при выполнении каждого шага, а стохастический и мини пакетный градиентные спуски также смогут достичь минимума, если мы будем использовать квадратичный график обучения.

```
theta_path_mgd = []
n_iterations = 50
minibatch_size = 20
np.random.seed(42)
theta = np.random.randn(2,1)
t0, t1 = 200, 1000
def learning_schedule(t):
    return t0 / (t + t1)

t = 0
for epoch in range(n_iterations):
    shuffled_indices = np.random.permutation(m)
    X_b_shuffled = X_b[shuffled_indices]
    y_shuffled = y[shuffled_indices]
    for i in range(0, m, minibatch_size):
        t += 1
        xi = X_b_shuffled[i:i+minibatch_size]
        yi = y_shuffled[i:i+minibatch_size]
        gradients = 2/minibatch_size * xi.T.dot(xi.dot(theta) - yi)
        eta = learning_schedule(t)
        theta = theta - eta * gradients
        theta_path_mgd.append(theta)
```

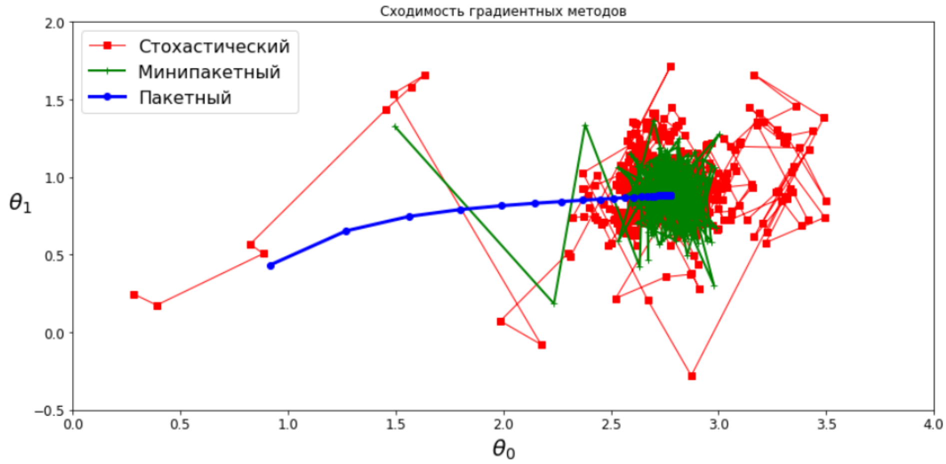

Рис. 6. Сходимость градиентных методов

Заключение

Рассмотрена оптимизационная задача применительно к задачам АПК. Изучены варианты самого общего алгоритма -метода градиентного спуска для построения моделей для задач сельского хозяйства, в частности построены модели и прогнозы по реализации сельхозпродукций с использованием регрессионного анализа. Создан пакет программ с использованием интеллектуальных систем машинного обучения по технологиям Python, на основе которого в дальнейшем планируется создание системы искусственного интеллекта. Данная методика позволяет изучать и строить модели для широкого круга задач в сфере АПК Кыргызстана по анализу и прогнозу продукции растениеводства и животноводства.

Библиографический список:

1. Бринк Х. Машинное обучение [Текст] / Х. Бринк, Дж. Ричардс, М. Феверолф. — пер. с англ. Рузмайкина И. — Санкт-Петербург: Питер, 2017. — 336 с.

2. Мюллер А. Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными [Текст] / А. Мюллер, С. Гвидо. — пер. Груздев А. — Москва: Альфа-книга, 2017. — 480 с.

3. Тихонов А.Н. Методы решения некорректно поставленных задач [Текст]/ Тихонов А.Н., Арсенин В.Я. – Москва: Наука, 1978. – 288 с.

4. Васильев Ф.П. Численные методы решения экстремальных задач: Учеб. Пособие для вузов [Текст]-Москва: Наука, 1988. -552 с.

5. Официальный сайт Python: официальный сайт Python; дистрибутивы, документация, примеры. – Режим доступа: http://scikit-rearn.org.

6. Энциклопедия языков программирования [Электронный ресурс]. – Режим доступа: http://progopedia.ru/language/python/.

Оригинальность 96,6%