ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

НА ТЕМУ

«Разработка компьютерной системы прогнозирования налоговых и акцизных поступлений»

2024

Содержание

[Введение 4](#_Toc163759049)

[1. Анализ предметной области 7](#_Toc163759050)

[1.1. Анализ налогов и акцизов, поступающих в федеральный бюджет 8](#_Toc163759051)

[1.2. Анализ существующих решений 11](#_Toc163759052)

[1.3. Анализ существующих алгоритмов прогнозирования 13](#_Toc163759053)

[1.3.1. Экспоненциальное сглаживание 13](#_Toc163759054)

[1.3.2. Двойное экспоненциальное сглаживание 14](#_Toc163759055)

[1.3.3. Множественная линейная регрессия 14](#_Toc163759056)

[1.3.4. ARIMA 18](#_Toc163759057)

[1.3.5. SSA 20](#_Toc163759058)

[1.3.6. Выбор алгоритмов прогнозирования 22](#_Toc163759059)

[1.4. Анализ метрик для временных рядов 23](#_Toc163759060)

[2. Проектирование и разработка системы прогнозирования 25](#_Toc163759061)

[2.1. Проектирование базы данных 26](#_Toc163759062)

[2.2. Проектирование и реализация бэкенда 28](#_Toc163759063)

[2.2.1. Описание реализации дата-слоя 29](#_Toc163759064)

[2.2.2 Описание реализации слоя вычислений 30](#_Toc163759065)

[2.2.3. Описание реализации других слоев 33](#_Toc163759066)

[2.3. Проектирование и реализация фронтенда 37](#_Toc163759067)

[3. Анализ применения алгоритмов 45](#_Toc163759068)

[3.1. Прогноз для НДПИ 46](#_Toc163759069)

[3.2. Прогноз для налога на прибыль 47](#_Toc163759070)

[3.3. Прогноз для НДФЛ 48](#_Toc163759071)

[3.4. Прогноз для НДС 49](#_Toc163759072)

[3.5. Прогнозы акцизов 50](#_Toc163759073)

[3.6. Прогноз объема федерального бюджета 51](#_Toc163759074)

[4. Заключение 52](#_Toc163759075)

[Список терминов и сокращений 53](#_Toc163759076)

[Список литературы 55](#_Toc163759077)

[Приложение 57](#_Toc163759078)

# Введение

Важной государственной задачей является эффективное функционирование его институтов, что обеспечивается наполнением бюджета на разных уровнях. Налоги и акцизы являются основным инструментом для решения данной задачи. Прогнозируя налоги, на основе макроэкономических и налоговых индикаторов, можно предсказывать тренд развития экономики и, как следствие, страны.

При планировании любого бюджета, важна объективная точность, поэтому при прогнозировании важно предусмотреть возможность различных вариаций факторов, по которым рассчитываются целевые показатели, а также возможность прогнозировать различными методами.

Автоматизированная мульти-алгоритмическая компьютерная система позволит упростить процесс прогноза и улучшить его качество и точность благодаря одновременному использованию нескольких алгоритмов прогнозирования, в то время как в существующих системах используется единственный выбранный заранее алгоритм.

**Цель дипломной работы** – разработать автоматизированную систему прогнозирования налоговых поступлений и акцизов, совмещенную с базой данных для хранения истории изменения налоговых поступлений, с использованием существующих алгоритмов прогнозирования.

Для достижения данной цели были поставлены следующие **задачи:**

* Изучить существующие алгоритмы прогнозирования и выбрать из них несколько;
* Собрать данные по налогам и акцизам;
* Разработать базу данных для хранения данных по налогам и акцизам;
* Разработать систему с задокументированным API для прогнозирования с использованием выбранных алгоритмов, которая будет использовать данные из базы данных;
* Разработать Web-интерфейс для взаимодействия с системой;
* Проанализировать прогнозы, полученные выбранными алгоритмами прогнозирования, и с помощью этих прогнозных значений получить прогноз объема Федерального бюджета на следующий календарный период.

**Практическая ценность** заключается в реализации моделей прогнозирования и хранения данных для прогнозирования в единой системе, а также возможность добавления в будущем новых алгоритмов прогнозирования помимо тех, которые будут реализованы в этой работе.

**Научная новизна** работы состоит в новом алгоритме, который использует одновременно несколько алгоритмов прогнозирования для их сравнительного анализа и выбора наилучшего варианта с помощью критериев из математической статистики и математических ошибок.

**Структура работы.** Работа состоит из введения, 3 глав и заключения.

В первой главе будут перечислены налоги и акцизы, прогнозом которых мы будем заниматься в данной работе, и также будут рассмотрены выбранные алгоритмы прогнозирования.

Вторая глава будет посвящена разработке автоматизированной системы прогнозирования, а именно: будет описана структура БД; будут описаны принципы работы системы прогнозирования и точки доступа для web-приложения; будет описана структура web-приложения, внешний вид экранов, функционал и т.д.

Третья глава будет посвящена вычислению прогнозных значений налогов и акцизов на следующий календарный период с помощью разработанной системы, а также получению прогноза объема федерального бюджета с использованием полученных значений налогов и акцизов.

В заключении сформулированы основные выводы работы.

# 1. Анализ предметной области

В современном российском государстве основными источниками поступлений федерального бюджета[1-3] являются: налоговые поступления, неналоговые поступления, безвозмездные поступления.

Налоговые поступления (далее налоги) расходуются на социально важные государственные потребности, такие как: обеспечение национальной обороны, социальной политики, здравоохранения, развития культуры и спорта, образование и обслуживание государственного долга.

Налоги в рамках государства выполняют следующие основные функции:

* распределение ВВП между отраслями и гражданами;
* обеспечение доходной базы системы государства;
* влияние на спрос и предложение в экономике.

Именно поэтому планирование федерального бюджета, а следовательно и планирование поступлений налогов, важно для государства при планировании бюджета на следующий календарный период.

Самым основным регулятором внутреннего рынка государства, в условиях рыночной экономики, являются налоги. Налогом является обязательный, индивидуально безвозмездный платеж, взимаемый с организаций и физических лиц в форме отчуждения принадлежащих им на праве собственности, хозяйственного ведения или оперативного управления денежных средств в целях финансового обеспечения деятельности государства и (или) муниципальных образований.

Большую часть поступлений в бюджет занимают налоговые, поэтому в данной работе рассматривается их прогнозирование.

## 1.1. Анализ налогов и акцизов, поступающих в федеральный бюджет

Согласно Бюджетному кодексу[5] и налоговому законодательству[6], федеральный бюджет наполняют следующие налоги: налог на добавленную стоимость (НДС), налог на доходы физических лиц (НДФЛ), налог на добычу полезных ископаемых (НДПИ), налог на прибыль (НПр), акцизы и другие налоги и сборы.



Рисунок 1. График налоговых поступлений в федеральный бюджет за 2023 год

**Налог на добычу полезных ископаемых (НДПИ)** является основным по объему поступлений в Федеральный бюджет РФ. Налогоплательщиками являются предприятия, организации, индивидуальные предприниматели, которые пользуются природными ресурсами. Основной налоговой базой являются полезные ископаемые, производимые при добыче нефти, природного газа, угля и полезных руд.

**Налог на добавленную стоимость (НДС)** является косвенным и вторым налогом по объему поступлений, а также наиболее стабильным источником наполнения бюджета. Объектом налогообложения является реализация товаров и услуг. Поступления осуществляются за счет реализации продавцом товаров, услуг, имущественных прав покупателю.

**Налог на прибыль организаций**: основным объектом налогообложения является прибыль организаций и предприятий, получаемая в процессе их деятельности. Прибылью является разность суммы доходов и суммы расходов. Объем налога зависит от организации и ее финансовых результатов. Данный налог уплачивают все юридические лица (ОАО, ООО и тд), а также иностранные юридические лица и организации, которые получают доход от источника в РФ.

**Налог на доход физических лиц (НДФЛ)**: объектом налогообложения являются доходы, получаемые резидентами или нерезидентами Российской Федерации. Доходы, которые облагаются данным налогом – это доходы, получаемые при заработке, а также доходы, полученные при продаже и сдаче имущества в аренду, в том числе, доходы от различных выигрышей.

**Акцизы** – это косвенный налог, объектом налогообложения которого является реализация подакцизных товаров. Согласно Бюджетному кодексу и налоговому законодательству, подакцизными товарами являются:

1. этиловый спирт;
2. спиртосодержащие лекарственные и косметические средства;
3. алкогольная продукция;
4. табачная продукция;
5. автомобили;
6. автомобильный бензин;
7. дизельное топливо;
8. моторные масла;
9. средние дистилляты (углеводороды, продукты первичной и вторичной переработки нефти);
10. бензол;
11. авиационный керосин;
12. природный газ;
13. жидкости для электронных систем доставки никотина;
14. табак;
15. виноград, использованный для производства вина;
16. этан;
17. сталь жидкая.

Акцизы работают почти также как налог на добавленную стоимость – объем акциза закладывается продавцом в итоговую стоимость подакцизного товара. Поэтому факторами, влияющими на объем акцизных поступлений, являются суммарная стоимость реализованного подакцизного товара и акцизная ставку на этот товар.

Стоит отметить, что больше всего на суммарный объем акцизных поступлений влияют акцизы на:

1. табачную продукцию;
2. табак;
3. автомобильный бензин;
4. дизельное топливо;
5. алкогольная продукция;
6. авиационный керосин;
7. природный газ;
8. нефтяное сырье.

Сбор и контроль исполнения данных налоговых доходов является одной из стратегических функций Федеральной налоговой службы, так как данные поступления формируют государственный бюджет.

## 1.2. Анализ существующих решений

Существующие методы прогнозирования в экономике основываются на:

* отдельном сборе данных для прогнозирования и хранении их в отдельной системе;
* последующем построении прогноза с помощью различных алгоритмов и методик вручную или с помощью компьютерной системы.

Поэтому скорость получения прогноза такими способами довольно мала из-за большого количества ручной работы. К тому же, если возникнет потребность поменять алгоритм прогнозирования, потребуется повторно выполнять прогнозирование экономических факторов, а при большом объеме данных это может занять дни или даже недели. Помимо этого, повышается риск ошибки из-за человеческого фактора при переносе данных из одной системы в другую.

В связи с вышеизложенным возникает необходимость в разработке новой, более совершенной системы. Это должна быть автоматизированная мульти-алгоритмическая система прогнозирования экономических факторов, совмещающая в себе систему для прогнозирования и систему для хранения данных, которая позволит значительно ускорить процесс прогнозирования за счет совмещения двух ранее раздельных систем в одной. Также эта система должна позволять:

* С минимальными усилиями добавлять новые алгоритмы прогнозирования в случае, если потребуется перейти со старых алгоритмов на новые;
* Анализировать применение алгоритмов прогнозирования к различным экономическим факторам для выбора наилучшего.

Использование такой системы позволит с высокой скоростью строить прогнозы и анализировать качество этих прогнозов с помощью методов математической статистики, математических ошибок и графического представления. Также использование этой системы позволит с легкостью добавлять и обновлять данные по экономическим факторам и использовать новые данные для прогнозов. Более подробно про реализацию этой системы будет рассказано в следующей главе.

## 1.3. Анализ существующих алгоритмов прогнозирования

Временной ряд — собранный в разные моменты времени статистический материал о значении каких-либо параметров (в простейшем случае одного) исследуемого процесса. Изменение объема налога или акциза по годам можно представить в виде временного ряда, и с помощью алгоритмов прогнозирования для временных рядов получать прогноз объема налога или акциза на следующий календарный период.

Рассмотрим наиболее используемые алгоритмы прогнозирования временных рядов и выберем из них наиболее оптимальные и подходящие.

### 1.3.1. Экспоненциальное сглаживание

Один из самых простых в реализации и использовании алгоритм прогнозирования, который хорошо подходит для временных рядов, значения которых с течением времени находятся вокруг какой-либо константы. Например, для рядов, значения которых в рассматриваемые периоды изменяются в диапазоне A+-e, где А – константы, а е – ширина коридора.

Формула алгоритма:



где:

* – период расчета,
* – сглаженное значение за период ,
* – реальное значение ряда за период ,
* – коэффициент сглаживания.

### 1.3.2. Двойное экспоненциальное сглаживание

Это улучшение алгоритма экспоненциального сглаживания, которое хорошо подходит для прогнозирования рядов с выраженным трендом изменения значений. Формула алгоритма:

где:

* t – период расчета,
* – сглаженное значение за период t,
* – реальное значение ряда за период t,
* – коэффициент сглаживания,
* – величина влияния тренда
* – коэффициент сглаживания тренда

### 1.3.3. Множественная линейная регрессия

Этот алгоритм позволяет найти зависимость целевого значения (в нашем случае объем налога или акциза) от нескольких факторов в виде коэффициентов. Формула выглядит следующим образом:

где – спрогнозированное значение, – значения факторов.

Для использования этого алгоритма необходимо предварительно выбрать несколько факторов, от которых может зависеть целевое значение и вычислить значения коэффициентов, а для получения прогноза целевого значения необходимо сделать прогноз каждого из факторов любым подходящим способом и затем подставить спрогнозированные значения факторов в формулу линейной регрессии вместе с полученными значениями коэффициентов.

Коэффициенты регрессии можно искать разными способами. Один из них – с помощью нейронной сети. При подборе коэффициентов с помощью нейронной сети используется нейронная сеть определенного вида:

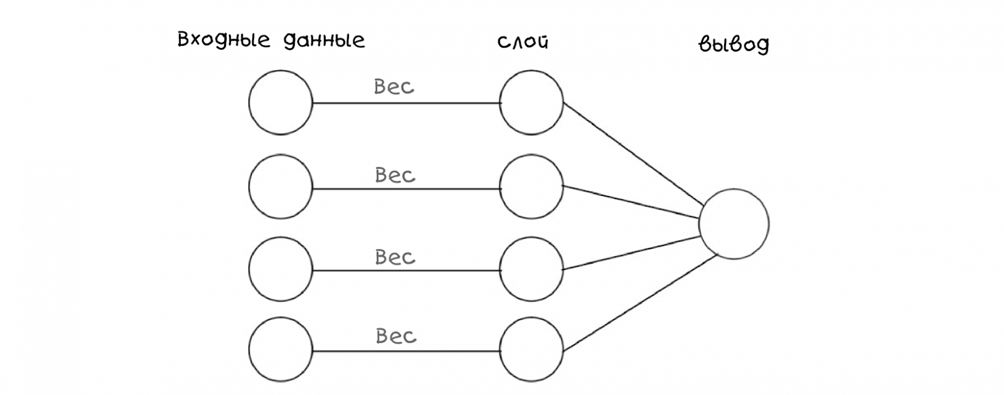


Рисунок 2 – Схема нейронной сети

где *входные данные* – значения факторов, *веса* – коэффициенты , *вывод* – наше спрогнозированное значение.

Алгоритм обучения нейронной сети:

1. *Умножает входные данные на вес*, связанный с определенным фактором (нейрон в контексте НС),
2. *Суммирует* перемноженные значения,
3. *Применяет* нормирующую функцию (функция активации):
4. *Применяет* нормирующую функцию к ожидаемому значению и получает ошибку:

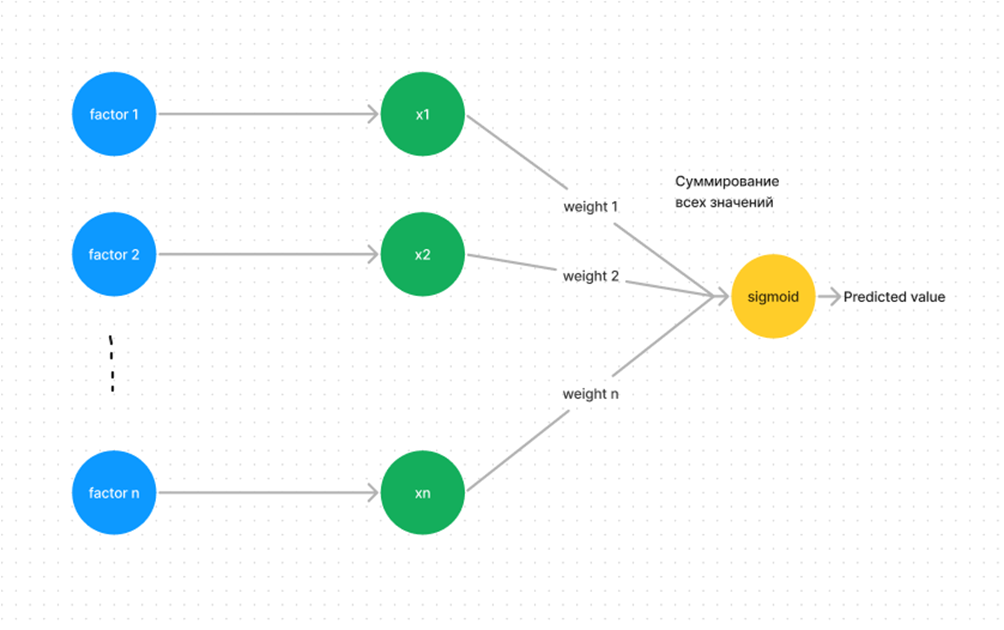
Степень у выражения будет меняться в зависимости от данных.

1. *Изменяет* веса в соответствии с формулой обратного распространения ошибки:

где

* – ошибка из предыдущего пункта,
* – производная функции активации,
* – значение фактора, связанного с этим весом,
* – произвольный множитель, влияющий на скорость обучения НС. Выбирается экспериментально.

На картинках показано графическое представление алгоритма:



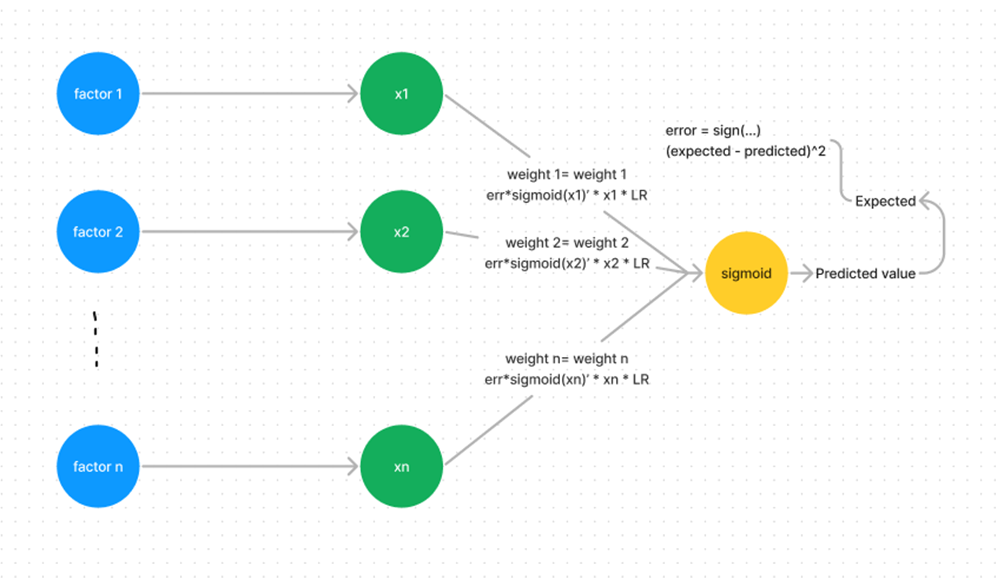


Рисунок 3 - Графическое изображение обучения нейросети и изменения весов нейронов

Данный алгоритм в среднем прогоняется 100000 раз для всего набора данных. Количество прогонов тоже выбирается экспериментально и важно выбрать его не слишком большим, иначе нейросеть найдет подходящие веса для конкретного набора данных и, когда полученные коэффициенты применятся к другим данным, может получиться слишком большое отклонение. Данная картинка хорошо описывает эту ситуацию:

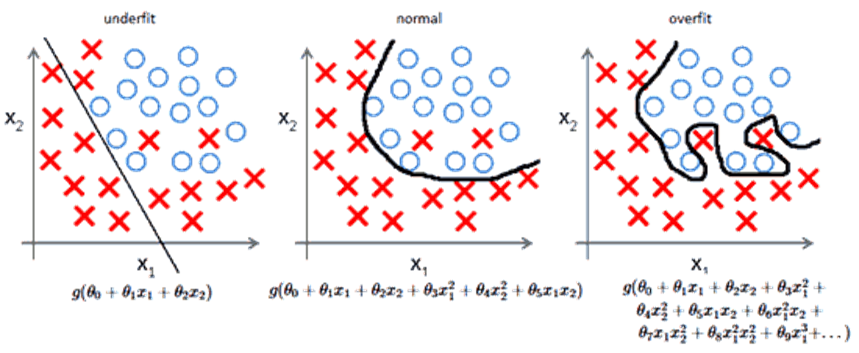


Рисунок 4 - Зависимость прогнозируемых значений от количества обучений на одном наборе данных

*Underfit* – НС недостаточно обучена, плохое прогнозирование.

*Normal* – НС обучена, приемлемое прогнозирование.

*Overfit* – НС переобучена, слишком точное прогнозирование, велика вероятность ошибок.

Чтобы понять стадию обучения НС будем использовать MRSE как индикатор. Сначала пять раз с нуля обучим НС на имеющихся данных и посмотрим какая будет получаться MRSE. Если отклонения в MRSE будут в районе 10-100, то можем считать, что НС обучена. Если же MRSE будут совпадать с точностью, то считаем что НС переобучена и уменьшаем количество итераций обучения.

После обучения НС мы сможем взять получившиеся веса как коэффициенты, a в выражении множественной линейной регрессии и использовать их для получения прогноза поступления на следующий календарный период.

### 1.3.4. ARIMA

Этот алгоритм состоит из 3 компонент: интегрирующая модель (I), авторегрессия (AR) и скользящее среднее (MA).

Сначала к временному ряду применяется интегрирующая модель. Она позволяет из нестационарных рядов получать стационарные. Нестационарные ряды – такие ряды, которые содержат в себе тренд и/или сезонную составляющую. Стационарные ряды таких особенностей не имеют.

Модель I имеет параметр d, которые отвечает за порядок дифференцирования значений временного ряда. Дифференцирование происходит по следующему принципу:

Где:

* – разность порядка 1 для периода t
* – значение ряда за период t

Такое дифференцирование можно делать несколько раз пока не получим стационарный ряд. Разности бОльших порядков определяются рекуррентно:

Модели AR и MA используют такие разности порядка d и работают следующим образом. Модель AR использует предыдущие значения ряда для получения нового значения:

Где p – число учитываемых предыдущих значений, называется порядком модели AR. Коэффициенты будут находиться по следующему принципу. Пусть в ряду 100 значений. Тогда коэффициенты можно найти следующим образом:

Модель MA использует значения предыдущих ошибок ряда для прогнозирования следующего значения:

Где:

* – среднее значение ряда до периода t
* – разница между средним значением ряда и значением за период i (ошибка)
* q – число учитываемых предыдущих значений ошибок

Итоговая модель ARIMA примет вид:

После получения прогнозного значения необходимо сделать преобразование, обратное тому, которое делает интегрирующая модель, и таким образом мы найдем прогноз целевого значения.

### 1.3.5. SSA

Базовый вариант метода состоит в:

* преобразовании одномерного ряда в многомерный с помощью однопараметрической сдвиговой процедуры (называют “Гусеница”);
* исследовании полученной многомерной траектории с помощью анализа главных компонент (сингулярного разложения);
* восстановлении (аппроксимации) ряда по выбранным главным компонентам.

Таким образом, результатом применения метода является разложение временного ряда на простые компоненты: медленные тренды, сезонные и другие периодические или колебательные составляющие, а также шумовые компоненты. Полученное разложение может служить основой прогнозирования как самого ряда, так и его отдельных составляющих.

Алгоритм допускает естественное обобщение на многомерные временные ряды, а также на случай анализа изображений. Также алгоритм содержит два входных параметра: длину “гусеницы” и число ее компонент, выбор которых существенно влияет на результат работы алгоритма.

Алгоритм работает следующим образом. За основу берется одномерный временной ряд (в нашем случае данные по объему налога или акциза) . Выбирается n – время жизни гусеницы, в диапазоне и вычисляется – длина гусеницы. Затем строится последовательность из n векторов следующего вида:

Обозначим . Будем называть Z нецентрированной матрицей наблюдений, порождённой гусеницей с временем жизни n.

Далее займемся анализом главных компонент ряда. Рассмотрим ковариационную матрицу полученной выборки: . Выполним ее svd-разложение: , где – диагональная матрица собственных чисел, и – ортогональная матрица собственных векторов.

Главные компоненты выглядят следующим образом: . После проведения анализа главных компонент обычно предполагается проведение операции восстановления исходной матрицы наблюдений по некоторому поднабору главных компонент, т. е. для и вычисляется . Далее восстанавливаются исходные последовательности. В одномерном случае i-ая компонента восстановленного ряда есть среднее значение по i-ой диагонали восстановленной матрицы наблюдений Z'.

Как строится прогноз? Определим:

и

Также положим:

Тогда прогнозное значение ряда в точке N+1 вычисляется по формуле:

Преимущество метода состоит в том, что он может выявлять сразу несколько относительно простых составляющих, которые влияют на итоговое значение временного ряда, что позволяет получать более долгосрочные и точные прогнозы на основе имеющихся данных.

### 1.3.6. Выбор алгоритмов прогнозирования

Для прогнозирования налоговых поступлений и акцизов будем использовать все алгоритмы, кроме экспоненциального сглаживания, так как экспоненциальное сглаживание хорошо работает только на стационарных рядах, а все ряды, которые мы будем исследовать в данной работе, стационарными не являются. Также стоит отметить, что для некоторых поступлений не будет использоваться множественная линейная регрессия ввиду недостаточного количества данных по факторам, влияющих на объем поступления.

# 1.4. Анализ метрик для временных рядов

Для оценки точности прогнозов будут использоваться метрики соответствия временных рядов. К таким метрикам относятся:

* Корень из среднеквадратичной ошибки (MRSE). Классическая метрика сравнения двух временных рядов. Быстро возрастает при минимальных несоответствиях в прогнозе, что позволяет быстро заметить отклонения. Вычисляется по формуле:

где – реальное значение ряда, – спрогнозированное значение ряда

* Алгоритм динамической трансформации (ADT). Позволяет определять соответствие рядов с учетом их возможного сдвига по оси времени или оси значений. Значение соответствие вычисляется по алгоритму:
  1. Берутся два временных ряда Q длины n и C длины m
  2. Строится матрица расстояний d порядка , где
  3. Используя матрицу расстояний, строится матрица трансформации D, элементы которой вычисляются по формуле
  4. Строится некоторый оптимальный путь W, берущий начало в и заканчивающийся в
  5. Ищется такой путь, чтобы расстояние трансформации было минимальным – .
* Средняя абсолютная масштабированная ошибка (MASE). Создана для того, чтобы убрать зависимость размера ошибки от значений сравниваемых рядов. Вычисляется по формуле:

где – реальное значение ряда, – спрогнозированное значение ряда.

Эти метрики будут использоваться для вычисления точности прогнозирования, а также для выбора наилучшего алгоритма прогнозирования для каждого из налогов.

# 2. Проектирование и разработка системы прогнозирования

В этой главе займемся разработкой системы. Система будет состоять из трёх компонент:

* **База данных (БД)** – будет хранить в себе все необходимые данные по налогам, акцизам и факторам, а также результаты прогнозов, вычисленные метрики и параметры;
* **Бэкенд[7]** – будет взаимодействовать с БД и выполнять прогнозы с помощью реализованных алгоритмов;
* **Фронтенд[8]** – графическая система для более удобного взаимодействия с бэкендом.

Вся система будет запускаться в окружении Docker в виде трёх контейнеров. Настройка контейнеров будет осуществляться в автоматическом режиме во время их сборки и запуска. За счет этого достигается простота развертывания системы на компьютере конечного пользователя. Далее будет описаны реализации каждой из компонент.

## 2.1. Проектирование базы данных

База данных будет работать на PostgreSQL и должна отвечать следующим требованиям:

1. Хранить данные по налогам, акцизам и факторам в удобном формате
2. Хранить результаты прогнозов в удобном формате
3. Хранить связанные с прогнозами данные: метрики, параметры

Для достижения этих целей была спроектирована следующая база данных:

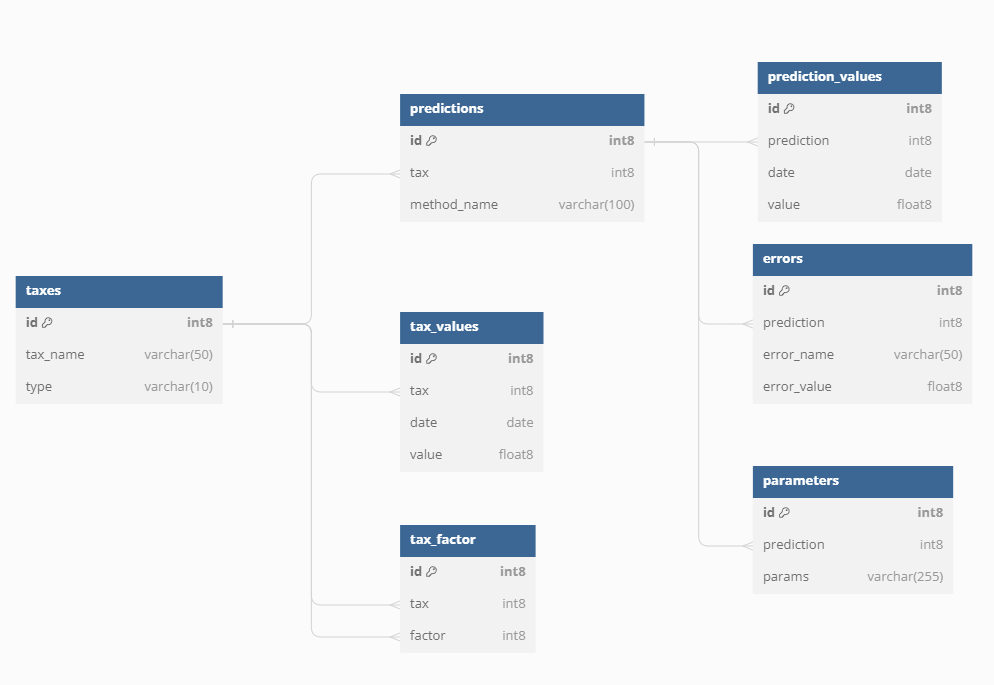


Рисунок 4 – Структура базы данных

В этой структуре главной таблицей является таблица, описывающая имеющиеся в системе налоги и их тип – налог или фактор для налога. Тип влияет на то как система будет взаимодействовать с этим налогом – либо позволять пользователю выполнять для этого налога прогнозы, либо использовать этот налог как фактор, влияющий на какой-то другой налог. От этой таблицы зависят таблицы со значениями этих налогов, таблица связей налогов с факторами и таблица прогнозов. В то же время, от таблицы прогнозов зависят таблицы со значениями прогнозов, таблица с метриками прогнозов и таблица с параметрами прогнозов. Такая структура базы данных позволит экономно хранить прогнозы с большим количеством значений за счет того, что строки с названиями хранятся в отдельных таблицах и не дублируются для каждой записи.

Создание описанной структуры будет происходит с помощью бэкенда, поэтому более подробно об этом будет рассказано в подглаве про бэкенд.

## 2.2. Проектирование и реализация бэкенда

Бэкенд будет разрабатываться на языке Java с использованием фреймворка Spring Boot, так как этот фреймворк значительно упрощает создание web-приложений. Web-приложение позволяет осуществлять взаимодействие посредством HTTP запросов с помощью заранее описанного API. Эта возможность добавляет большую гибкость системе, так как конечный пользователь, при желании, сможет реализовать свой фронтенд или внедрить бэкенд составляющую в уже существующую систему.

Опишем требования к возможностям бэкенда:

1. Выполнять запрос на прогнозирование
2. Выполнять запрос на получение всех прогнозов по налогу
3. Добавлять, изменять, просматривать и удалять данные по налогам и факторам
4. Получение параметров, с помощью которых был получен тот или иной прогноз
5. Получение метрик, описывающих качество прогноза

Для достижения этих целей реализуем следующую внутреннюю архитектуру приложения:

* **Бизнес-слой** – этот слой отвечает за прием запросов от пользователей. Состоит из нескольких контроллеров, в которых описаны точки доступа до приложения, а также из классов, которые контроллеры принимают от пользователя или отправляют в ответ на запросы. Получив запрос, перенаправляет его в сервисы в дата-слое или слое вычислений, а результат запросов во внутренние сервисы возвращает пользователю, предварительно преобразовав в удобные для конечного пользователя сущности;
* **Дата-слой** – слой, отвечающий за взаимодействие с базой данных и за агрегацию данных в удобные для бэкенда сущности. Состоит из сервисов, к которым приходят запросы от других сервисов и которые возвращают удобные для бэкенда сущности, и из репозиториев, через которые осуществляются запросы к базе данных;
* **Слой вычислений** – слой, отвечающий за применение реализованных алгоритмов и вычисление метрик. Состоит из сервиса, к которому приходят запросы на прогнозирование, реализованных алгоритмов и реализованных метрик. Реализации алгоритмов и метрик должны реализовывать определенные интерфейсы, и за счет этого достигается простота расширения функционала: достаточно написать новый класс, который реализует нужный интерфейс. Под реализацией интерфейса подразумевается реализация методов, описанных интерфейсом;
* **Слой конфигурации** – слой, отвечающий за различные глобальные настройки бэкенда: адрес базы данных, имя пользователя и пароль для базы данных, формат описания API и так далее;
* **Слой конфигурации базы данных** – слой, отвечающий за поддержание необходимой структуры и минимального наполнения базы данных в момент запуска бэкенда. Необходим для того, чтобы дать бэкенду гарантию того, что структура базы данных будет именно такой, как мы ожидаем.

### 2.2.1. Описание реализации дата-слоя

Главными слоями в бэкенде являются дата-слой и слой вычислений, так как в них хранятся и обрабатываются самые важные части бэкенда.

Минимально рабочая структура дата-слоя, отвечающая требованиями, такова:

* Сущности
  + Сущности для налогов
  + Сущности для прогнозов
* Мапперы
* Репозитории
  + Репозитории для налогов
  + Репозитории для прогнозов
* Представления
  + Представления налогов
  + Представления прогнозов
* Сервисы
  + Общий data сервис
  + Сервис для налогов
  + Сервис для прогнозов

Сущности описывают каждую из реализованных таблиц для их использования в системе. Изменения этих сущностей отражаются соответствуют изменениям соответствующих записей в связанной таблице. Репозитории позволяют получать из таблиц данные и записывать значения нужных записей в сущности.

Сервисы для налогов и прогнозов используют эти репозитории, чтобы получать доступ к таблицам базы данных, а затем с помощью мапперов преобразовывают полученные данные в представления для дальнейшего использования другими слоями бэкенда. Преобразовывание в представления необходимо для того, чтобы уйти от записей таблиц и работать уже с классами Java.

Общий data сервис нужен для того, чтобы агрегировать данные из одного или двух сервисов в какой-то один набор данных для удобного использования этих данных в других слоях. Через него проходят все запросы к дата-слою от других слоев.

### 2.2.2 Описание реализации слоя вычислений

Слой вычислений должен принимать запросы на вычисление прогнозов, во время которых будут вычисляться прогнозы налогов, метрики прогнозов, а также параметры, если таковые реализованы в алгоритме.

Структура слоя вычислений:

* Алгоритмы
  + Интерфейс для алгоритма прогнозирования
  + Интерфейс для параметров алгоритма прогнозирования
  + Класс для хранения значений для прогнозирования
  + Папка с реализованными алгоритмами
* Метрики
  + Интерфейс для метрики прогнозирования
  + Папка с реализованными метриками
* Классы для агрегирования результата прогноза
* Сервис для управления реализованными алгоритмами и метриками (или сервис управления реализованными сущностями)
* Сервис прогнозирования

Взаимодействие с этим слоем осуществляется только через сервис прогнозирования. При получении запроса на прогноз, этот сервис использует data сервис из дата-слоя, чтобы получить данные по налогу и данные по связанным факторам, затем обращается к сервису управления реализованными сущностями, чтобы найти необходимый алгоритм и выполнить прогноз налога.

После выполнения прогноза, сервис прогнозирования вновь обращается к сервису управления реализованными сущностями, чтобы получить все реализованные метрики и вычислить метрики для нового прогноза. Затем из data сервиса он получает уже выполненные прогнозы для данного налога, чтобы определить, насколько новый прогноз лучше или хуже уже имеющихся. Затем все полученные данные упаковываются в классы для агрегирования результатов прогноза и возвращаются тому, кто запрашивал прогноз.

Когда пользователю возвращается прогноз, он не записывается сразу в базу данных. Это сделано для того, чтобы в случае, когда пользователю по каким-либо причинам не понравится получившийся прогноз, предыдущий прогноз, который устраивал пользователя, не затерялся.

Теперь расскажем про внедрение в систему новых алгоритмов и метрик. Чтобы реализовать новый алгоритм прогнозирования, необходимо создать класс, который реализует интерфейс для алгоритма прогнозирования, а также содержит внутри себя класс, который реализует интерфейс для параметров алгоритма прогнозирования. После создания класса его необходимо поместить в папку с реализованными алгоритмами, и система уже при следующем запуске будет использовать этот алгоритм.

Интерфейс для алгоритма прогнозирования состоит из следующих функций:

* **makePrediction** – выполняет прогноз с использованием данных по налогу и связанными с этим налогом факторами;
* **getName** – позволяет получить названия алгоритма, которое будет использовать пользователь;
* **getParameters** – позволяет получить список параметров, необходимых алгоритму для получения прогноза. Возвращает класс, который реализует интерфейс для параметров алгоритма прогнозирования;
* **getDescription** – позволяет получить описание алгоритма, чтобы помочь пользователю понять что именно делает алгоритм или чтобы обьяснить смысл параметров;
* **getPredictionParameters** – возвращает список параметров, которые алгоритм получил при выполнении прогноза и которые несут смысловую нагрузку для пользователя.

Интерфейс для параметров алгоритма прогнозирования состоит из следующих функций:

* **getParametersNames** – возвращает список параметров, которые необходимо передать алгоритму для выполнения прогноза. Строка с названием параметра также содержит в себе описание возможных значений параметра;
* **parseParameters** – позволяет разобрать пришедшие строки на нужные значение параметров. Параметры передаются в виде строки, так как при выполнении запроса на прогноз можно указать сразу несколько значений параметра, чтобы алгоритм нашел наилучшее значение из перечисленных. Обычно несколько значений перечисляются через точку с запятой;
* **getParameterValues** – используется алгоритмом прогнозирования, чтобы получить переданные пользователем параметры.

Чтобы реализовать новую метрику, необходимо создать класс, который реализует интерфейс для метрики прогнозирования и поместить его в папку с реализованными метриками. Так же, как и с новым алгоритмом, при следующем запуске система начнет использовать новую метрику при выполнении прогнозов.

Интерфейс для метрики прогнозирования состоит из следующих функций:

* **calcError** – вычисляет значение метрики по реальным и спрогнозированным значениям налога;
* **getName** – позволяет получить название метрики, которое будет видеть пользователь в результате прогноза;
* **compareTo** – позволяет сравнить две метрики и узнать какая из них лучше.

### 2.2.3. Описание реализации других слоев

Другие слои бэкенда являются вспомогательными, поэтому их опишем кратко.

**Бизнес-слой** отвечает за прием запросов от клиента или фронтенда. Из него происходят обращения к дата-слою и слою прогнозирования. Состоит из двух контроллеров, первый принимает запросы на операции с данными по налогам, второй принимает запросы на операции с прогнозами. Перечислим все доступные точки доступа, а также кратко опишем их:

1. /prediction/predict – выполняет прогноз для выбранного налога с использованием выбранного алгоритма и необходимых параметров;
2. /prediction/predicts/get – возвращает все прогнозы для указанного налога;
3. /prediction/algorithms – возвращает список реализованных алгоритмов с описанием и списком необходимых параметров;
4. /prediction/confirm – используется для подтверждения или отклонения прогноза;
5. /taxes/values – позволяет получить список значений для налога или фактора;
6. /taxes/tax/names – позволяет получить список налогов;
7. /taxes/factor/names – позволяет получить список факторов;
8. /taxes/tax/factors – позволяет получить список факторов, связанных с указанным налогом;
9. /taxes/save – позволяет сохранить изменение данных по налогу или фактору;
10. /taxes/tax/factor/add – позволяет добавить связь налога с фактором;
11. /taxes/tax/add – позволяет добавить новые данные по налогу или фактору;
12. /taxes/add/file – позволяет добавить новые данные по налогу или фактору с помощью переданного файла в формате .csv;
13. /taxes/delete – позволяет удалить данные по налогу или фактору;
14. /taxes/tax/factor/delete – позволяет удалить связь фактора с налогом.

Полное описание каждой из точек доступа (API) можно найти по адресу http://\*адрес компьютера, на котором запущен бэкенд\*:8080/swagger-ui/index.html#/. По этому адресу можно найти подробное описание каждой точки доступа, ее входные параметры, а также возвращаемые значения с примерами. Пример описания точки доступа для получения выполненных прогнозов для налога:

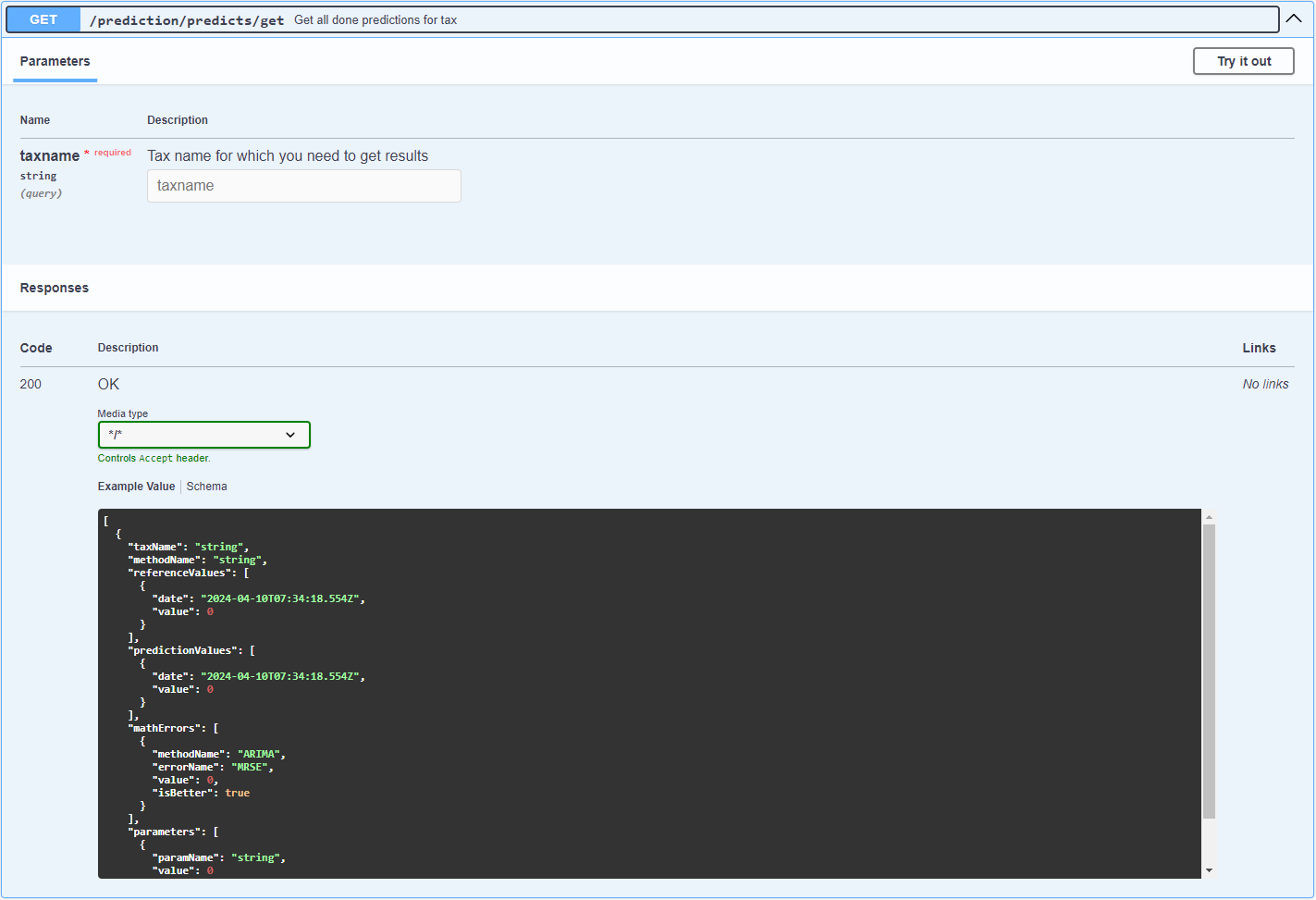


Рисунок 5 – Пример описания точки доступа

**Слой конфигурации** отвечает за подключение к базе данных, за настройку API. Состоит из нескольких классов-конфигураций и глобального класса настроек.

**Слой конфигурации базы данных** использует библиотеку Liquibase, которая записывает в базу данных изменения, описанные в специальном xml-документе. Пример описания двух таблиц в этом файле с указанием зависимости одной таблицы от другой:



Рисунок 6 – XML-документ для создания двух таблиц в базе данных при запуске бэкенда

Код бэкенда можно будет найти по ссылке в приложении.

## 2.3. Проектирование и реализация фронтенда

Фронтенд будет разрабатываться на языке TypeScript с использованием фреймворка React. Такое сочетание позволяет создавать сайты с динамическим наполнением страниц, что облегчает разработку.

Фронтенд (или по-другому – сайт) должен позволять конечному пользователю взаимодействовать с системой через удобный интерфейс, а не через API, что ускорит получение прогноза и операции с данными, а также уменьшит риск отправки некорректных данных, что тоже ускорит взаимодействие с бэкендом.

Фронтенд должен содержать следующие страницы:

* **Главная страница –** страница, которая будет встречать пользователя при первом заходе на сайт;
* **Страница выполнения прогноза** – на этой странице можно будет выбрать налог и алгоритм, с помощью которого пользователь будет выполнять прогноз. Когда будет выбран алгоритм прогнозирования, на странице появится описание алгоритма, а также список необходимых параметров с полями для их ввода;
* **Страница просмотра выполненных прогнозов** – на этой странице можно будет налог, а также алгоритм из списка тех, с помощью которых уже выполнялся прогноз для этого алгоритма. После этого на странице появится информация, связанная с этим прогнозом, а также графики и сводки, которые помогут лучше понять качество прогноза;
* **Страница для просмотра и изменения данных по налогам** – на этой странице можно будет: посмотреть и изменить данные по налогам; посмотреть и изменить данные по факторам; посмотреть с какими факторами связан тот или иной налог, добавить или удалить такие связи; добавить данные по налогу или фактору.

Для взаимодействия с бэкендом необходимы функции, которые будут использоваться на страницах для получения или отправки данных, а также сущности, которые будут использовать при получении и отправке данных.

Структура фронтенда:

* Сущности для взаимодействия с бэкендом
* Функции для взаимодействия с бэкендом
* Различные иконки и картинки, необходимые для страниц
* Код страниц

Теперь расскажем про внешний вид каждой из страниц. Внешний вид главной страницы:

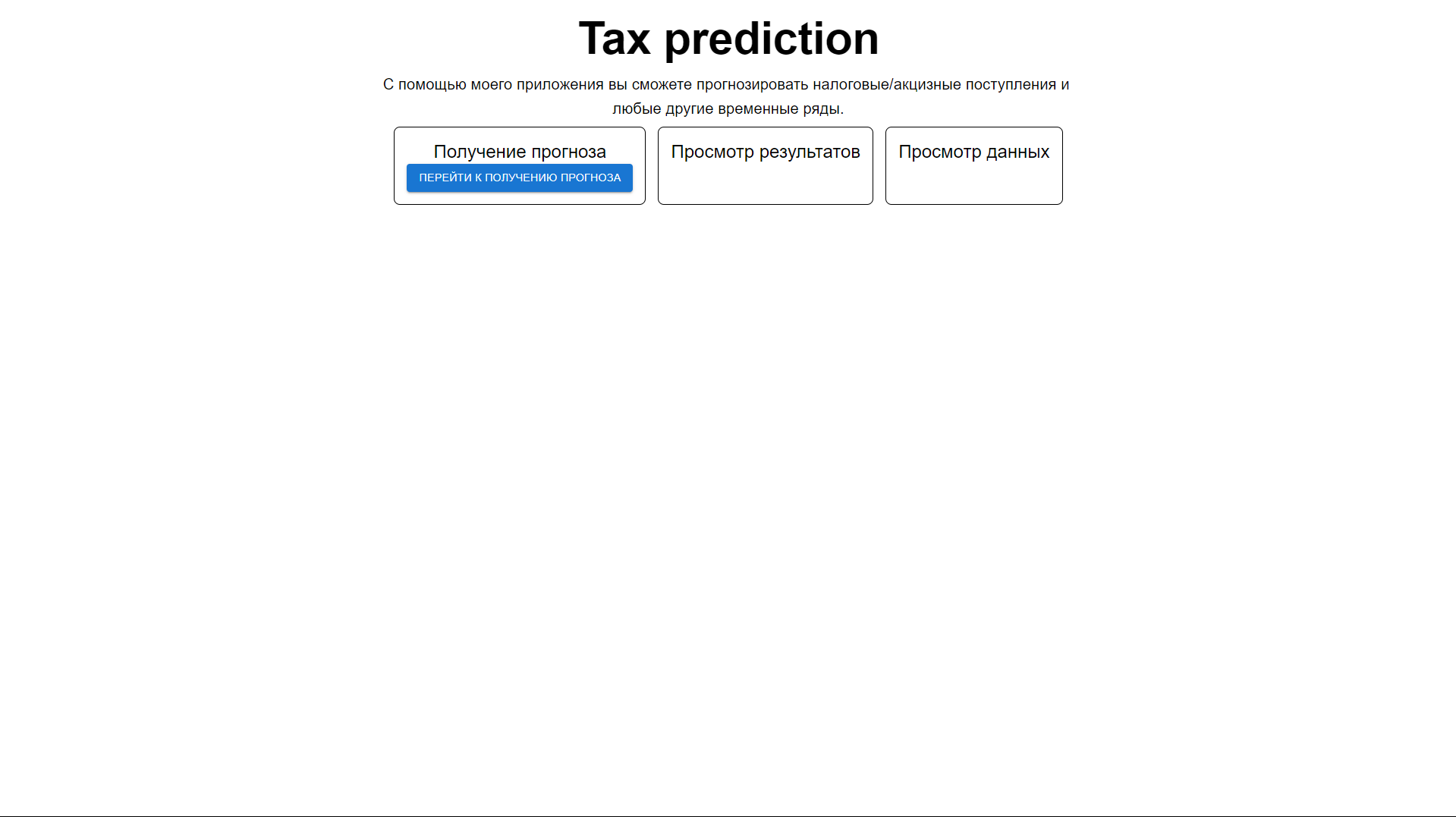


Рисунок 7 – Вид главной страницы

Здесь находится описание приложения, а также краткое описание возможностей с кнопками перехода на соответствующие страницы.

Внешний вид страницы выполнения прогноза:

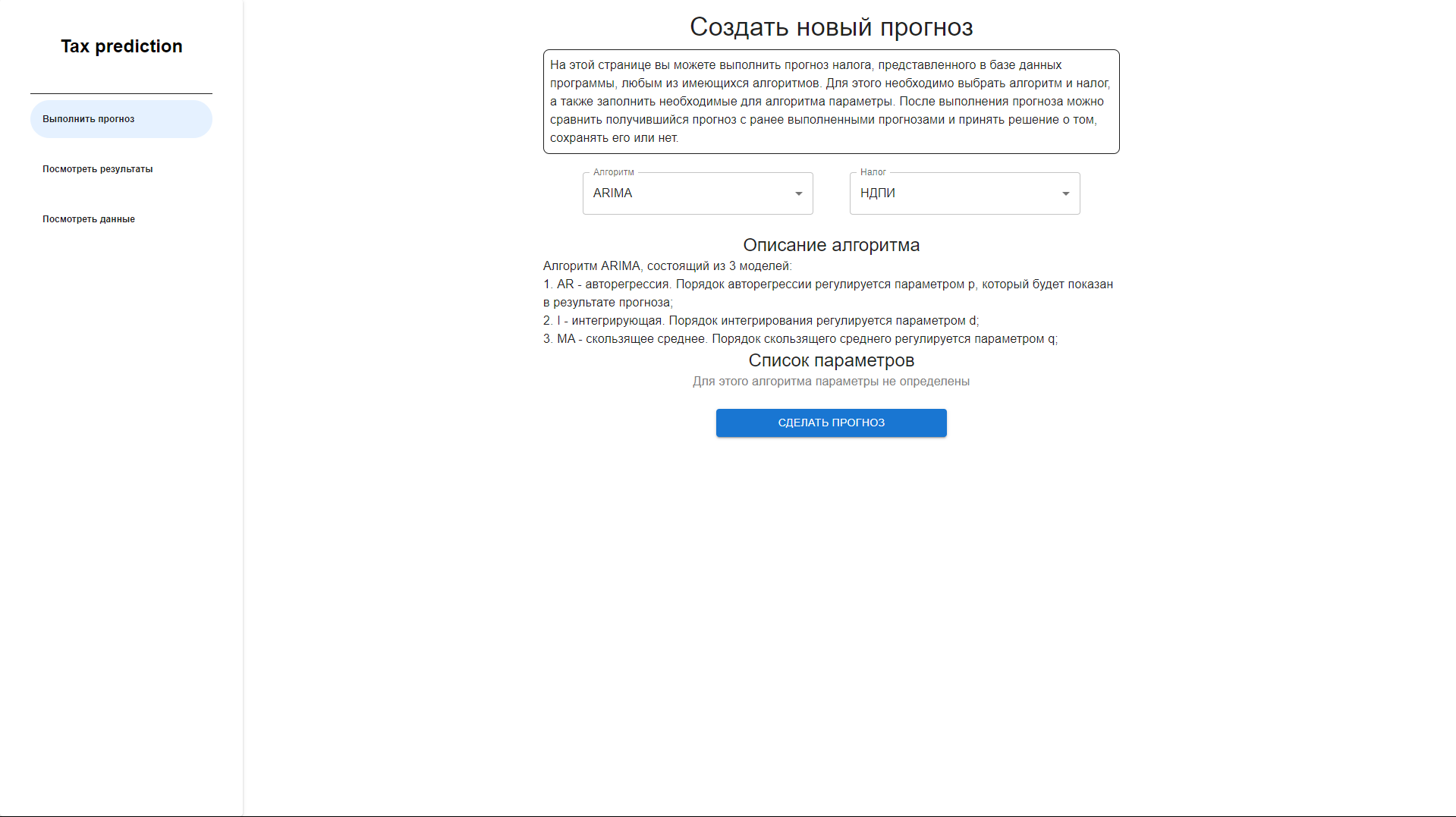


Рисунок 8 – Внешний вид страницы выполнения прогнозов

На этой и оставшихся страницах слева располагается меню для быстрого переключения между страницами. Помимо этого, на странице есть описание алгоритма получения прогноза. Для получения прогноза необходимо выбрать алгоритм прогнозирования и налог, для которого необходимо выполнить прогноз. После задания необходимых параметров, если таковые требуются, необходимо нажать кнопку «Сделать прогноз». Спустя какое-то время на экране появляется выполненный прогноз вместе с другими прогнозами, если таковые имеются:

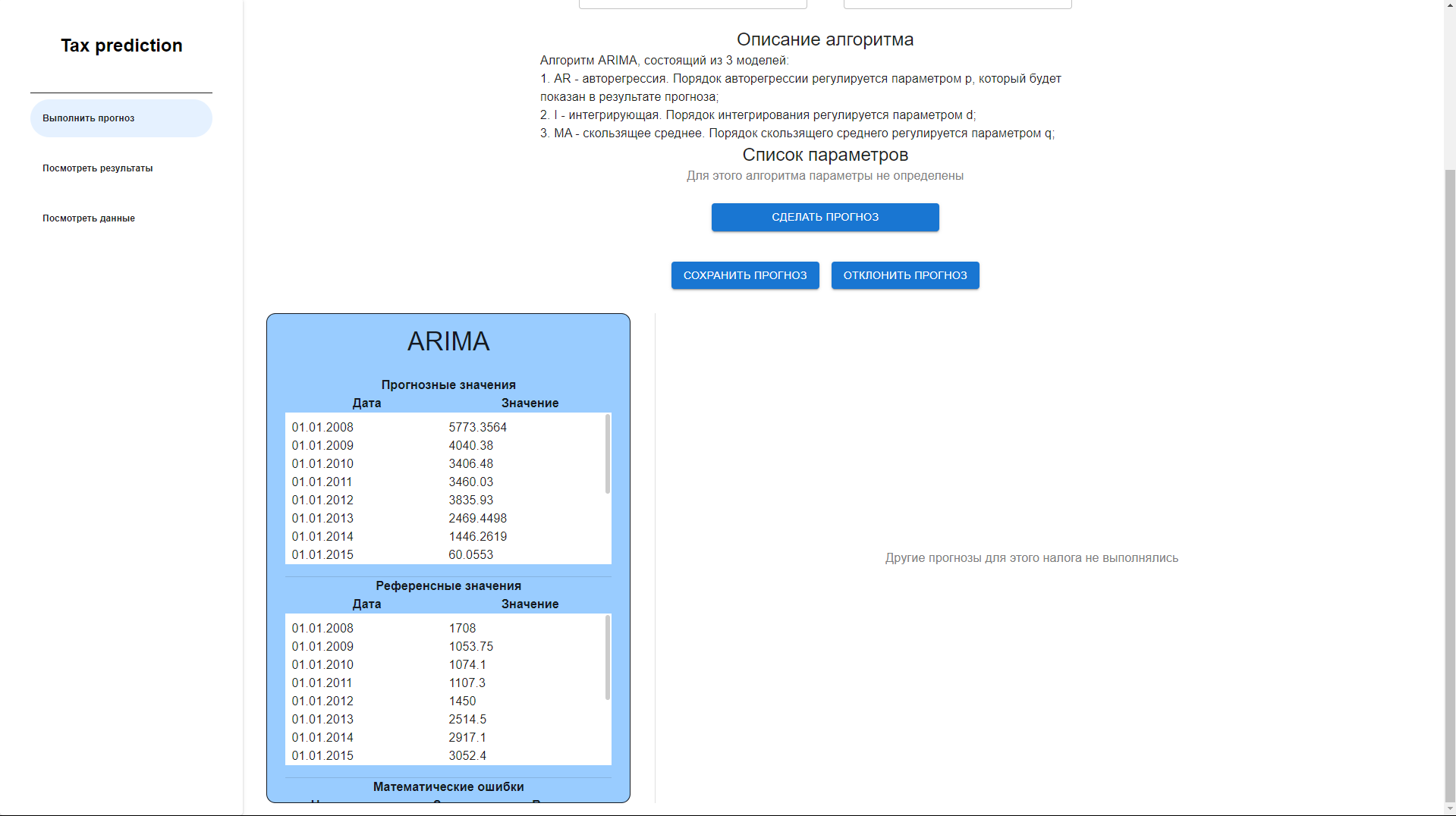


Рисунок 9 – Внешний вид страницы при получении прогноза

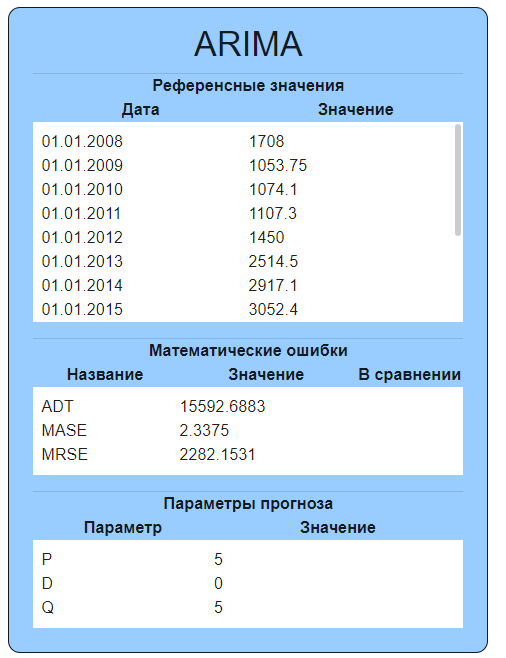


Рисунок 10 – Внешний вид одного из прогнозов

Возвращаемые результаты прогнозов показываются в краткой форме, достаточной для того, чтобы оценить его точность по сравнению с другими.

Внешний вид страницы просмотра выполненных прогнозов:

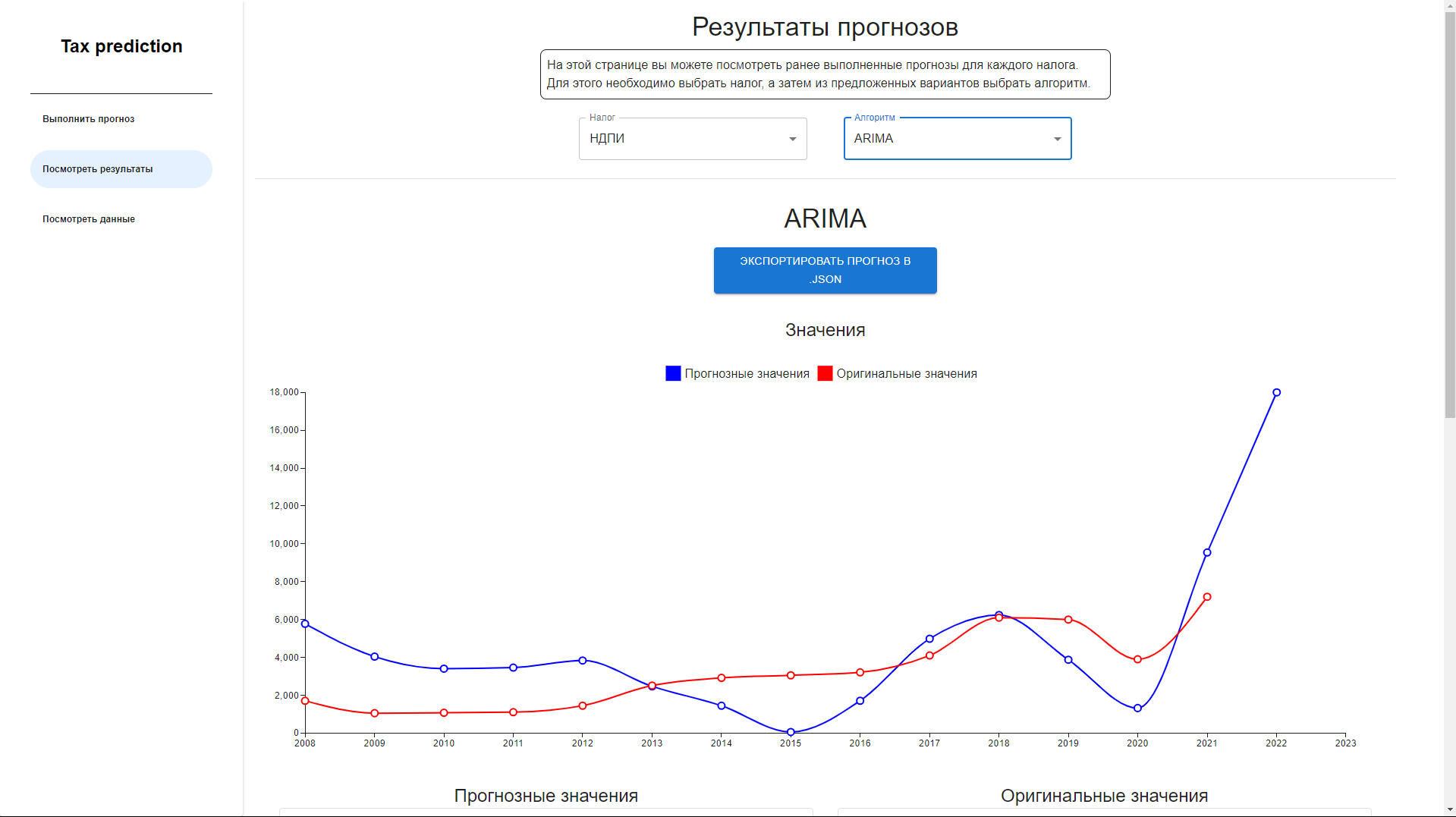


Рисунок 11 – Внешний вид страницы с результатами прогнозов

На этой странице, при выборе налога и алгоритма прогнозирования, показывается инфографика прогноза: график изменения значений, таблицы значений, таблицы параметров и метрик и так далее. На этой странице также есть возможность экспортировать выбранный прогноз в JSON формат, на случай если прогноз необходимо использовать где-то еще. После нажатия на кнопку «Экспортировать прогноз» скачается файл с названием «Названиеналога\_названиеалгоритма\_prediction.json». Выбор формата экспорта обусловлен популярностью формата и почти повсеместным его использованием в других системах.

Перейдем к внешнему виду страницы с данными по налогам и факторам. Данная страница содержит в себе 3 вкладки: для просмотра и изменения данных по налогам, для просмотра и изменения данных по факторам, для добавления новых данных. Внешний вид вкладки с данными по налогам:

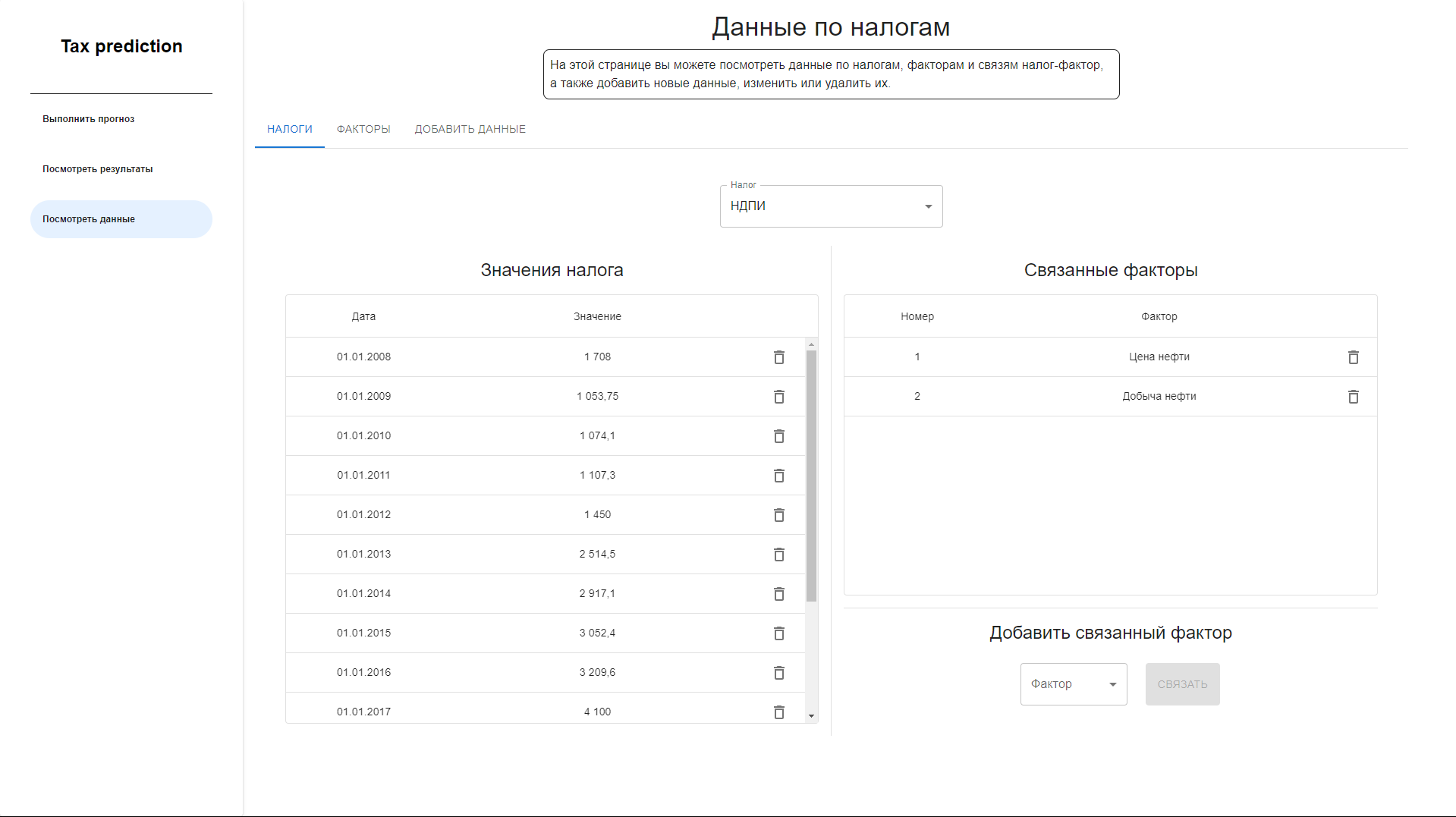


Рисунок 12 – Внешний вид страницы с данными по налогам

На этой вкладке необходимо выбрать налог, данные по которому необходимо посмотреть. После этого появится таблица с данными, в которой можно изменить или удалить данные, а также таблица связей этого налога с факторами.

Внешний вид вкладки с данными по факторам:

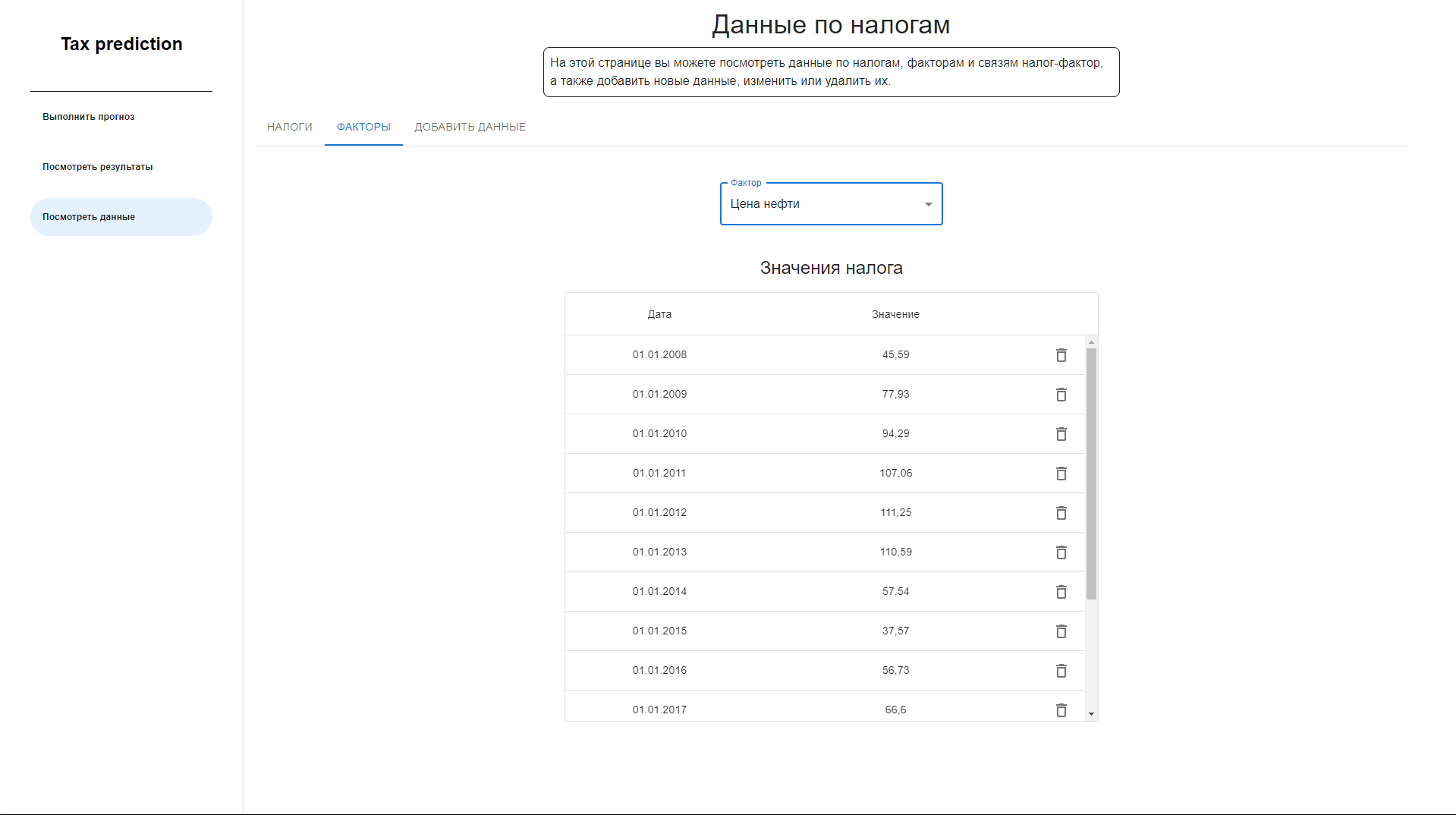


Рисунок 13 – Внешний вид вкладки с данными по факторам

На этой вкладке можно посмотреть, изменить или удалить данные по какому-либо фактору.

Внешний вид вкладки для добавления данных:

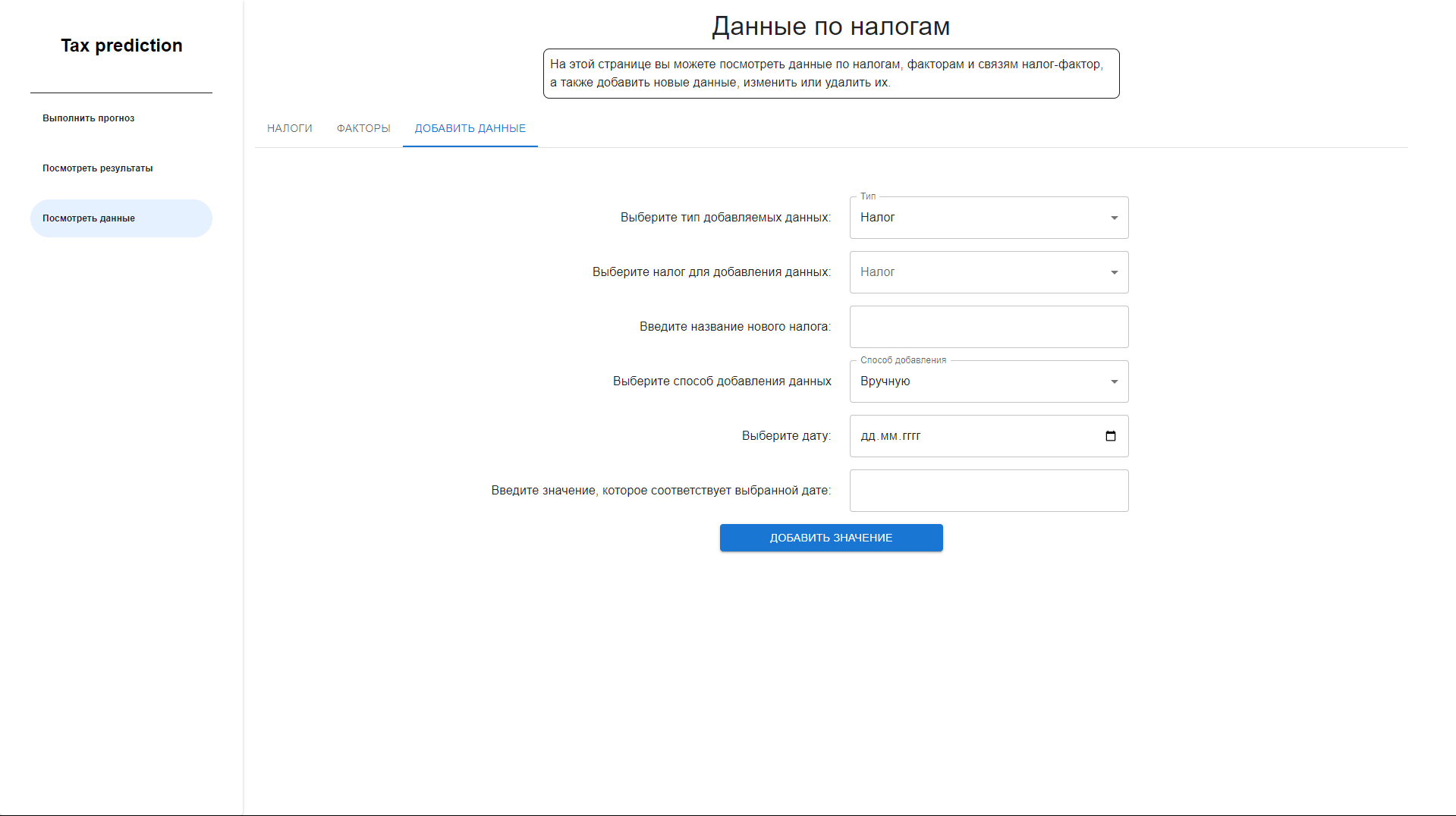


Рисунок 14 – Внешний вид вкладки для добавления данных

Для добавления данных необходимо сначала выбрать тип добавляемых данных. После этого в поле выбора налога/фактора появятся содержащиеся в базе данных налоги, к которым можно добавить новые данные. Также, если выбрать вариант «Новый налог», то появится поле для ввода названия нового налога.

После выбора налога, к которому будут добавляться данные, необходимо выбрать способ добавления данных: вручную или с помощью файла. При вводе данных вручную необходимо выбрать дату и значение, которое соответствует этой дате. При добавлении данных через файл, необходимо будет предоставить файл, который содержит в себе данные в виде множества csv строк формата ‘Дата;Значение’.

# 3. Анализ применения алгоритмов

Теперь применим разработанную систему для прогнозирования налогов и акцизов, а также для получения прогноза объёма федерального бюджета.

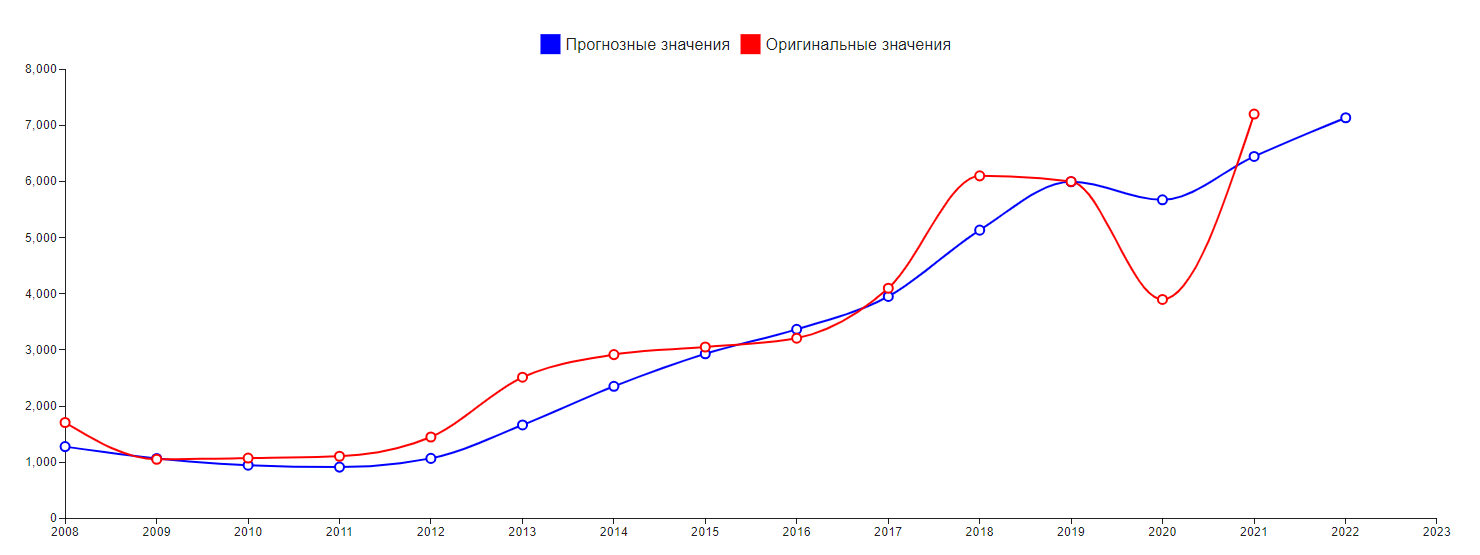
Для начала, необходимо добавить все необходимые данные в систему. После этого, на странице выполнения налогов на фронтенде, выполним прогноз каждым из реализованных алгоритмов для каждого из налогов и акцизов, а затем выберем лучший прогноз для каждого из них. Получив все необходимые прогнозы, мы сможешь выполнить прогноз федерального бюджета.

Теперь рассмотрим лучший прогноз для каждого из налогов и акцизов, а именно:

1. НДПИ;
2. Налог на прибыль организаций;
3. НДФЛ;
4. НДС;
5. Акциз на табачную продукцию;
6. Акциз на табак;
7. Акциз на автомобильный бензин;
8. Акциз на дизельное топливо;
9. Акциз на алкогольная продукция;
10. Акциз на авиационный керосин;
11. Акциз на природный газ;
12. Акциз на нефтяное сырье.

## 3.1. Прогноз для НДПИ

Для НДПИ лучший прогноз получился с использованием алгоритма двойного экспоненциального сглаживания:



Метрики прогноза:

* **ADT** – 1581,477. Для других алгоритмов эта метрика равна 15592.688 и 33015.265.
* **MASE** – 0,538. Для других алгоритмов эта метрика равна 2.237 и 4.153.
* **MRSE** – 661,889. Для других алгоритмов эта метрика равна 2282.153 и 4171.869.

Для этого алгоритма параметры прогноза не определены.

## 3.2. Прогноз для налога на прибыль

*Здесь будет результат прогноза*

## 3.3. Прогноз для НДФЛ

*Здесь будет результат прогноза*

## 3.4. Прогноз для НДС

*Здесь будет результат прогноза*

## 3.5. Прогнозы акцизов

*Здесь будут результаты прогнозов*

## 3.6. Прогноз объема федерального бюджета

*Здесь будет прогноз федерального бюджета*

# 4. Заключение

При выполнении данной работы я провел анализ налоговой системы Российской Федерации и выяснил, какие налоги и акцизы оказывают наибольшее влияние на объем федерального бюджета. Я создал мультиалгоритмическую расширяемую автоматизированную систему прогнозирования налоговых поступлений, которая хранит в себе все необходимые данные, а также для которой можно легко реализовать новые алгоритмы и с легкостью внедрить их в систему. С использованием этой системы я сделал прогнозы четырех видов налогов и восьми видов акцизов, которые затем были использованы при получении прогноза федерального бюджета.

# Список терминов и сокращений

1. **Бюджет** – это форма образования и расходования денежных средств, предназначенных для финансового обеспечения задач и функций государства и местного самоуправления.
2. **Госбюджет** является важным рычагом влияния на развитие производительных сил общества, реализации экономической политики государства, которая направлена на рост уровня жизни населения страны.
3. **Федеральный бюджет** выполняет следующие задачи:

* перераспределение ВВП;
* государственное регулирование и стимулирование экономики;
* финансовое обеспечение бюджетной сферы и осуществление социальной политики государства;
* контроль за образованием и использованием централизованных фондов денежных средств.

1. **Доходы федерального бюджета** – денежные средства, поступающие в бюджет в безвозмездном и безвозвратном порядке в соответствии с законодательством Российской Федерации в распоряжение органов государственной власти Российской Федерации, за исключением средств, являющихся источниками финансирования дефицита бюджета.
2. **Бюджетный кодекс –** кодифицированный нормативно-правовой акт, устанавливающий общие принципы бюджетного законодательства Российской Федерации, организации и функционирования бюджетной системы Российской Федерации, правовое положение субъектов бюджетных правоотношений, определяет основы бюджетного процесса и межбюджетных отношений в Российской Федерации, порядок исполнения судебных актов по обращению взыскания на средства бюджетов бюджетной системы Российской Федерации, основания и виды ответственности за нарушение бюджетного законодательства Российской Федерации.
3. **Налоговое законодательство** – совокупность нормативных правовых актов государства, регулирующих налоговое право.
4. **Бэкенд –** компонент системы, запускающийся без графического интерфейса, и отвечающий за взаимодействие с внешним миром и другими внутренними сервисами.
5. **Фронтенд –** графический компонент системы, которые взаимодействует с бэкендом.

# Список литературы

1. Аналитический портал ФНС России [Электронный ресурс] URL: <https://analytic.nalog.gov.ru/>
2. Данные по формам налоговой отчетности [Электронный ресурс] URL: <https://www.nalog.gov.ru/rn77/related_activities/statistics_and_analytics/forms/>
3. Описание алгоритма SSA [Электронный ресурс] URL: <http://www.machinelearning.ru/wiki/index.php?title=%D0%9F%D1%80%D0%BE%D0%B3%D0%BD%D0%BE%D0%B7%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%B2%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%BD%D1%8B%D1%85_%D1%80%D1%8F%D0%B4%D0%BE%D0%B2_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%BE%D0%BC_SSA_%28%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%29>
4. Описание алгоритма ARIMA [Электронный ресурс] URL: <https://www.mathnet.ru/links/8164e407c11fed7c9ba53b6a5f00ecd9/vyurv256.pdf>
5. Описание метода обратного распространения ошибки для обучения НС [Электронный ресурс] URL: <https://robocraft.ru/algorithm/560>
6. Выдержка из научной статьи по экономике на тему “Прогнозирования в экономике” [Электронный ресурс] URL: <https://cyberleninka.ru/article/n/prognozirovanie-v-ekonomike/viewer>
7. Статья о реализации компьютерной системы прогнозирования на базе MS Excel [Электронный ресурс] URL: <https://fundamental-research.ru/ru/article/view?id=39174>
8. Пример используемых методов прогнозирования экономики на основе ВВП, стр. 57-75 [Учебное пособие] URL: <https://sev.msu.ru/wp-content/uploads/2020/02/UMP-Socialno-jekonomicheskoe-prognozirovanie.pdf>
9. Описание алгоритма динамической трансформации [Электронный ресурс] URL: <http://www.machinelearning.ru/wiki/images/c/c3/Digital_Signal_Processing%2C_lecture_6.pdf>
10. Описание средней абсолютной масштабированной ошибки [International Journal of Forecasting] URL: <https://pure.eur.nl/ws/portalfiles/portal/47633436/A-note-on-the-MASE-Revision-for-IJF.pdf>

# Приложение

1. Репозиторий с кодом системы – <https://github.com/SlamperBOOM/Graduate-Work>