

Информационная система прогнозирования собираемости платежей в отделениях почтовой связи «Почта России» с использованием машинного обучения

В.В. Бадашев, А.Н. Скоба

Южно-Российский государственный политехнический университет (НПИ) им М.И. Платова, Новочеркасск

Аннотация: В данной статье рассматривается прогнозирование собираемости платежей в отделениях почтовой связи, с учетом сезонности и применением машинного обучения. Разработан алгоритм построения расчетной модели, которая предоставляет возможность для аналитиков почты РФ делать ежемесячный прогноз собираемости платежей для каждого УФПС (Управление федеральной почтовой связи) с учетом сезонности. Данная модель также позволяет получать отклонения от нормы для прогнозируемых и реально получаемых значений собираемых платежей и более точно корректировать повышение тарифов на услуги. Разработанная информационная система реализована в виде веб-сайта с помощью фреймворка ASP.NET Core и библиотеки для машинного обучения ML.NET.

Ключевые слова: математическое моделирование, прогнозирование с учетом сезонности, собираемость платежей, машинное обучение, нейронная сеть.

Для решения задачи прогнозирования собираемости платежей в отделениях почтовой связи «Почта России» был использован метод машинного обучения Singular spectrum analysis (SSA), подробное описание которого приведено в работах [1,2]. Детальная постановка задачи прогнозирования собираемости платежей приведена в работе [3]. Использование метода SSA для построения прогнозных моделей включает два основных этапа: встраивание и сингулярное разложение. Суть данных этапов заключается в следующем.

Шаг 1. Встраивание. Шаг встраивание — это начальный шаг в алгоритме SSA. Этот метод преобразует наблюдаемые временные ряды в многомерную векторную последовательность. Техника встраивания отображает исходный временной ряд путем формирования $K=N-L+1$ векторов вложения, имеющих размерность L :

$$X_i = (f_{i-1}, \dots, f_{i+L-2})^T, \quad 1 \leq i \leq K.$$

траекторная матрица – это матрица вида:

$$X = (x_{ij})_{i,j=1}^{L,K} = \begin{pmatrix} f_0 & f_1 & f_2 & \dots & f_{K-1} \\ f_1 & f_2 & f_3 & \dots & f_K \\ f_2 & f_3 & f_4 & \dots & f_{K+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{L-1} & f_L & f_{L+1} & \dots & f_{N-1} \end{pmatrix}. \quad (1)$$

Можно заметить, что в уравнении (1) матрица X имеет одни и те же элементы на анти-диагоналях. Матрица такого типа называется матрицей Ганкеля [1].

Шаг 2. Сингулярное разложение. В результате данного шага осуществляется сингулярное разложение траекторной матрицы ряда. Пусть $S = XX^T$. Обозначим $\lambda_1, \dots, \lambda_L$ собственные числа матрицы S , взятые в неубывающем порядке $\lambda_1 \geq \dots \geq \lambda_L \geq 0$ и U_1, \dots, U_L – ортонормированную систему собственных векторов матрицы S , соответствующих собственным числам. Пусть $d = \max\{i: \lambda_i \geq 0\}$. Пусть $V_i = X^T U_i \sqrt{\lambda_i}, i = 1, \dots, d$, то сингулярное разложение матрицы X может быть записано в виде:

$$X = X_1 + \dots + X_d, \quad (2)$$

где $X_i = \sqrt{\lambda_i} U_i V_i^T$. Каждая из матриц X_i имеет ранг 1, поэтому их называют элементарными матрицами. Набор $(\sqrt{\lambda_i} U_i V_i^T)$ называют i -й собственной тройкой сингулярного разложения (2).

Сконструированная таким образом матрица X содержит всю необходимую информацию для получения прогнозных значений собираемости платежей с учётом сезонности [3]. Разработанная и программно-реализованная информационная система, на основе метода прогнозирования SSA, для решения задачи прогнозирования собираемости платежей в отделениях почтовой связи «Почта России» является примером применения распределенных систем обработки информации (СОИ) [3]. Данная распределённая СОИ была реализована с использованием

комбинации фреймворка ASP.NET Core и библиотеки машинного обучения ML.NET, которые являются мощными инструментами для разработки надежных программных приложений [4]. Фреймворк, основанный на языке программирования C# и включающий механизм представления Razor [4], обеспечивает надежную основу для создания сложных веб-систем. Для получения качественных и количественных оценок при прогнозировании необходимо учитывать множественные факторы и связи [5, 6]. Используя ML.NET для прогнозирования платежей методом SSA, можно получать всю необходимую информацию о тенденциях изменения платежей и принимать более обоснованные решения о минимизации финансовых потерь. При этом были рассмотрены варианты её программной реализации, как на базе двухуровневой архитектуры «клиент-сервер», так и на базе трехуровневой клиент-серверной архитектуры [7]. При разработке системы использовался предметно-ориентированный подход [8]. Рассмотрим более подробно программную реализацию системы. На первом шаге осуществляется сбор данных с региональных Управлений федеральной почтовой связи (УФПС) «Почта России» для реализации процесса обучения модели прогнозирования. Модель сущности для хранения данных для обучения за определенную дату по УФПС в базе данных представлена на рис.1.

```
public class UfpsStatistic : Entity
{
    [Key]
    Ссылка: 0
    public override int Id { get; set; } // Первичный ключ
    Ссылка: 10
    public int UfpsId { get; set; } // Id УФПС
    [ForeignKey("UfpsId")]
    Ссылка: 4
    public virtual REFUFPS Ufps { get; set; } // УФПС
    Ссылка: 5
    public int CountPayment { get; set; } // Кол-во принятых платежей
    Ссылка: 6
    public float SumTotal { get; set; } // Сумма платежей
    Ссылка: 28
    public DateTime Date { get; set; } // Дата
}
```

Рис. 1. – Модель для хранения собранных данных в базе данных

Сбор данных осуществляется через Аpi платежной системы «Почта России». На втором шаге полученные данные преобразуются в модель данных (см. рис.1) и записываются в базу данных. Процесс сбора данных представлен на рис.2. На третьем шаге формируется модель, на которой и будет обучаться нейронная сеть (рис.2.) [9].

```
public class UfpsPaymentAnalysisData
{
    public float Year; // Год
    public float Month; // Месяц
    public float Day; // День
    Ссылка: 9
    public string Date { get; set; } // Дата
    Ссылка: 4
    public string NameFull { get; set; } // Наименование УФПС
    Ссылка: 4
    public float CountPayment { get; set; } // Кол-во платежей
    Ссылка: 4
    public float SumTotal { get; set; } // Сумма платежей
}
```

Рис. 2. – Модель данных для прогноза

После того, как модель для построения прогноза сформирована – необходимо определить пайплайн обучения нейронной сети с помощью метода ForecastBySsa из ML.NET для прогнозирования временных рядов SSA [4]. Пайплайн представлен на рис.3.

```
IEstimator<ITransformer> forecastEstimator = mlContext.Forecasting.ForecastBySsa(
    outputColumnName: nameof(UfpsPaymentCountTimeSeriesPrediction.ForecastedPaymentCountUnits),
    inputColumnName: nameof(UfpsPaymentAnalysisData.CountPayment),
    windowSize: 3,
    seriesLength: ufpsStatistics.Count,
    trainSize: ufpsStatistics.Count,
    horizon: 1,
    confidenceLevel: 0.95f,
    confidenceLowerBoundColumn: nameof(UfpsPaymentCountTimeSeriesPrediction.ConfidenceLowerBound),
    confidenceUpperBoundColumn: nameof(UfpsPaymentCountTimeSeriesPrediction.ConfidenceUpperBound),
    variableHorizon: true);
```

Рис. 3. – Пайплайн прогнозирования модели методом SSA.

Необходимо также отметить, что успешная реализация разработанной модели прогнозирования во многом зависит от правильно подобранных

значений параметров. Подробное описание каждого из параметров модели прогнозирования представлено в работе [10]. Так, например, параметр "Размер окна" особенно важен, поскольку он определяет точность сценарной модели путем разложения данных временного ряда на сезонные и шумовые компоненты, а поэтому он был установлен в значение 3 с учетом того, что сезон в рассматриваемых данных меняется каждые 3 месяца. Использование параметров доверительного интервала обеспечивает дополнительный уровень надежности и помогает снизить риски, связанные с чрезмерной зависимостью от прогнозов. Метод ForecastBySsa, который используется для прогнозирования применяет алгоритмы машинного обучения, чтобы автоматически определять параметры модели для прогнозирования. Таким образом, ML.NET Forecast BySsa позволяет прогнозировать собираемость платежей в разных регионах и учитывать сезонность, праздники и выходные дни. Для того, чтобы провести обучение модели на собранных данных о платежах и суммах с УФПС можно использовать метод Fit объекта PredictionEngine [10], который принимает в качестве аргументов наборы данных для обучения и тестирования модели. После обучения можно также провести оценку качества модели, используя для этого различные метрики. После того, как модель обучена и ее качество оценено, её можно использовать для прогнозирования будущих значений о собранных платежах и их суммах с УФПС. Для этого можно использовать метод Predict() [9], который принимает на вход временной ряд, для которого нужно сделать прогноз, и возвращает предсказанные значения на заданный интервал времени. Фактически временной ряд можно сформировать путем получения объекта, реализующего интерфейс IDataView, например, считав массив данных, состоящий из данных модели представленной на рис.2. с помощью метода LoadFromEnumerable() из свойства Data класса MLContext.

Для графического отображения прогноза с наложением реальных данных были использованы графики, построенные с помощью Kendo UI для ASP.NET Core [10]. Графики, построенные с помощью Kendo UI, могут быть полезными при визуализации результатов прогнозирования и оценки точности прогноза. Kendo UI предоставляет различные типы графиков, такие как линейные, столбчатые, круговые и другие, которые могут использоваться для отображения временных рядов и сравнения прогнозных значений с фактическими данными. Например, для прогнозирования количества собранных платежей с УФПС можно построить линейный график, где по оси X будут расположены даты, а по оси Y - количество принятых платежей. Затем целесообразно добавить на этот график две линии: одну для отображения фактических значений собранных платежей, а другую - для отображения прогнозных значений и соответственно сравнить прогноз с фактическими данными для оценки его точности. Так график количества собранных платежей в УФПС по одному из регионов России, за период шести месяцев, представлен на рис.4.

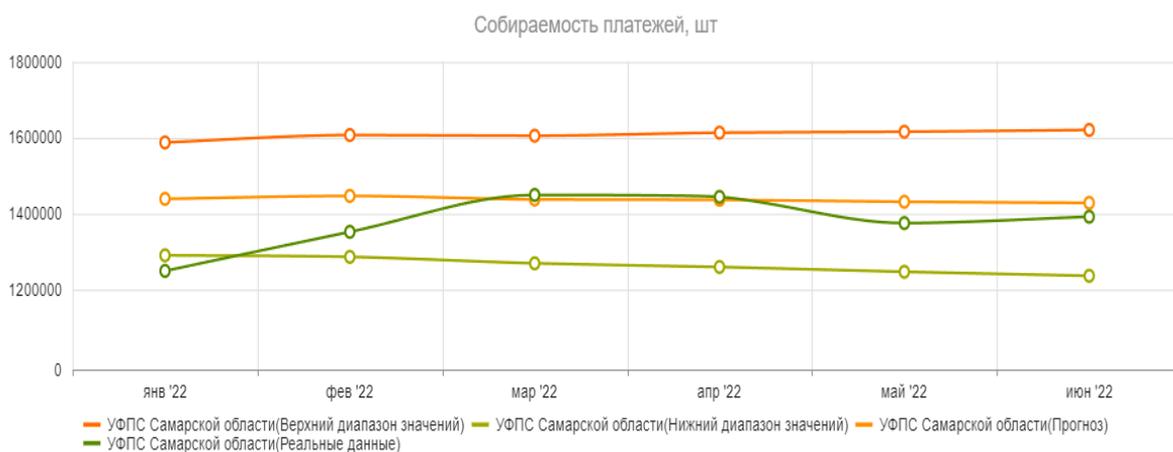


Рис. 4. – График количества собранных платежей

Практическая апробация разработанной и программно реализованной прогнозной модели по данному региону показала: точность модели прогнозирования количества принятых платежей - 95,85%; точность модели

прогнозирования суммы принятых платежей - 90,65%. Полученные результаты позволяют утверждать о высокой эффективности построенной модели прогнозирования. Соответственно, разработанную и программно-реализованную распределённую СОИ, использующую данный метод прогнозирования, можно рекомендовать для практического использования для получения достоверных прогнозов о собранных платежах каждого из УФПС «Почта России».

Литература

1. Голяндина Н.Э. Метод “Гусеница” -SSA: анализ временных рядов: СПб: Университет СПбГУ, 2003 — 87 с.
2. Golyandina, N., Nekrutkin, V., & Zhigljavsky, A. (2017). Singular Spectrum Analysis: A New Tool in Time Series Analysis. Chapman and Hall/CRC — 69 p.
3. Скоба А.Н., Михайлов В.К., Айеш Ахмед Нафеа Айеш. Решение задачи обеспечения оптимальной эффективности функционирования распределенных систем обработки информации // Инженерный вестник Дона. 2021. №6. URL: ivdon.ru/ru/magazine/archive/n6y2021/7053.
4. Чамберс Джеймс, Пэкетт Дэвид, Тиммс Саймон Ч-17 ASP.NET Core. Разработка приложений. — СПб.: Питер, 2018. — 464 с.
5. Wooldridge J.M. Introductory Econometrics: A Modern Approach. South Western: Cengage Learning, 2013. 912 p.
6. Цвиль М.М., Великанова Е.С. Прогнозирование объемов таможенных платежей с использованием фиктивных переменных // Инженерный вестник Дона, 2020, № 3. URL: ivdon.ru/ru/magazine/archive/N4y2020/6401.
7. Мартин Р. М29 Чистая архитектура. Искусство разработки программного обеспечения. — СПб: Питер, 2021. — 352 с.

8. Предметно-ориентированное проектирование (DDD): структуризация сложных программных систем.: Пер. с англ. - М.: 000 "И.Д. Вильямс", 2011. - 448 с.

9. Лебедева И.М., Федорова А.Ю., Макроэкономическое планирование и прогнозирование. Под ред. А.Ю. Федоровой. – СПб: Университет ИТМО, 2016. – 54 с.

10. de la Torre, C., Gandhi, P., & Mukherjee, S. (2020). Hands-On Machine Learning with ML.NET: Build custom machine learning models using C# or F#. Packt Publishing — 296 p.

References

1. Goljandina N.Je. Metod “Gusenica” SSA: analiz vremennyh rjadov [Caterpillar SSA method: Time series analysis]: Ucheb. posobie. SPb: SPbGU, 2000, 87 p.

2. Golyandina, N., Nekrutkin, V., & Zhigljavsky, A. 2017. Singular Spectrum Analysis: A New Tool in Time Series Analysis. Chapman and Hall (CRC), 69 p.

3. Skoba A.N., Mihajlov V.K. Inzhenernyj vestnik Dona, 2021, №6. URL: ivdon.ru/ru/magazine/archive/n6y2021/7053.

4. Chambers Dzhejms, Pjekett Djevid, Timms Sajmon Ch 17 ASP.NET Core. Razrabotka prilozhenij. [ASP.NET Core. Application development], SPb.: Piter, 2018, 464 p.

5. Wooldridge J.M. Introductory Econometrics: A Modern Approach. South Western: Cengage Learning, 2013, 912 p.

6. Csvil' M.M., Velikanova E.S. Inzhenernyj vestnik Dona, 2020, № 3. URL: ivdon.ru/ru/magazine/archive/N4y2020/6401.

7. Martin R. Chistaja arhitektura. Iskusstvo razrabotki programmogo obespechenija. [Clean architecture. The art of software development], SPb.: Piter, 2021, 352 p.



8. Predmetno-orientirovannoe proektirovanie (DDD): strukturizacija slozhnyh programmnyh sistem. [Subject-oriented design (DDD): structuring of complex software systems]: Per. s angl, M.: 000 "I.D. Vil'jams", 2011, 448 p.

9. Lebedeva I.M., Fedorova A.Ju., Makroekonomicheskoe planirovanie i prognozirovanie [Macroeconomic planning and forecasting]. Pod red. A.Ju. Fedorovoj, SPb: Universitet ITMO, 2016, 54 p.

10. De la Torre, C., Gandhi, P., & Mukherjee, S. (2020). Hands-On Machine Learning with ML.NET: Build custom machine learning models using C# or F#. Packt Publishing, 296 p.