

Прогнозирование продаж с помощью среды программирования Python (на основе примера базового прогноза временных рядов)

Приходько Артур Николаевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассматривается прогнозирование продаж с помощью среды программирования Python на основе примера прогноза временных рядов. Практическим результатом является график прогноза продаж, с возможностью сравнения готового прогноза с имеющимися данными.

Ключевые слова: python, прогнозирование, временные ряды

Forecasting sales using the Python programming environment (based on a time series forecasting example)

Prikhodko Arthur Nikolaevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article explores sales forecasting using the Python programming environment based on a time series forecasting example. The practical result is a sales forecast graph, with the ability to compare the finished forecast with the available data.

Keywords: python, forecasting, time series

Экономическая сфера жизни общества, как и любая другая, с развитием информационных технологий приобретает новые возможности.

Одним из таких нововведений является возможность осуществлять прогнозирование результатов торговли своего предприятия с помощью сред программирования, основываясь на математических формулах.

Прогнозирование представляет собой предсказание будущих показателей, основанное на накопленном опыте и текущем положении дел [1].

Прогнозирование продаж тесным образом связано с временными рядами.

Временными рядами называют собранные в разные моменты времени статистические материалы, в которых отражены значения тех или иных параметров исследуемого процесса [5].

Как итог мы получаем наглядные результаты прогноза в виде графика, значения которого близки к реальным показателям. Получаемый прогнозируемый материал позволяет сформировать ожидаемую статистику

продаж товара, тем самым предприятие может скорректировать план будущих поставок/производства и оценить рентабельность того или иного реализуемого товара [3].

Следует отметить, что прогноз является приблизительным, т.к. в программе на данный момент невозможно учесть влияние тех или иных форс-мажорных обстоятельств (неожиданная дестабилизация в стране, пандемии и т.д.), поэтому расчет ведётся в условиях, где торговле не препятствуют такого рода обстоятельства [4].

По теме статьи имеется обширный исследовательский материал. Например, математическое объяснение прогнозирования временных рядов представлено в книге «Анализ временных рядов и прогнозирование» за авторством Н.А. Садовниковой и Р.А. Шмойловой [5]. Что касается программно-технической реализации по теме, здесь можно найти различного рода статьи других авторов, например «Программно-математическое обеспечение прогнозирования локальных показателей» за авторством О. К. Куклиной, Е. А. Михайловой А. С. Яхиной [4]; «Методы краткосрочного прогнозирования финансовых временных рядов с малыми объёмами выборки» А.И. Передриенко, Т.П. Лютой, И.М. Харитонова, И.В. Степанченко [3]; «Методы машинного обучения в малом бизнесе: содержание и управление» С. А. Тищенко, М. А. Шахмурадяна [2] и «Программная реализация долгосрочного прогнозирования временных рядов на языке Python» В.Н. Семеновой, П.А. Гатина [1].

Цель данной статьи разобрать на примере имеющегося датасета процесс прогнозирования продаж товара.

В качестве изначальных статистических данных будет использован датасет, в котором представлены продажи шампуня за период в 3 года (<https://www.kaggle.com/guangningyu/sales-of-shampoo>).

Датасет о продажах шампуня за период в 3 года включает в себя количество продаж за каждый месяц (рис. 1).

Month	# Sales
1-01	266.0
1-02	145.9
1-03	183.1
1-04	119.3
1-05	180.3
1-06	168.5

Рисунок 1 – Часть датасета

На основе него составим базовый прогноз в среде Python со сравнением с реальными показателями, который в будущем можно модифицировать для разработки более сложных моделей прогнозирования.

В качестве метода прогнозирования будет использован метод устойчивости (или «наивный» метод). Его главные преимущества – лёгкость в разработке и точность прогноза для товаров с регулярным спросом, что идеально подходит для рассматриваемого датасета [2].

Для начала, разберемся наглядно в изначальных данных.

На основе имеющихся данных построим график временных рядов, используя следующий код (рис. 2):

```
from pandas import read_csv
from pandas import datetime
from matplotlib import pyplot

def parser(x):
    return datetime.strptime('190'+x, '%Y-%m')

series = read_csv('shampoo-sales.csv', header=0, parse_dates=[0], index_col=0, squeeze=True, date_parser=parser)
series.plot()
pyplot.show()
```

Рисунок 2 – Код для построения графика

В результате получаем следующий график (рис. 3):

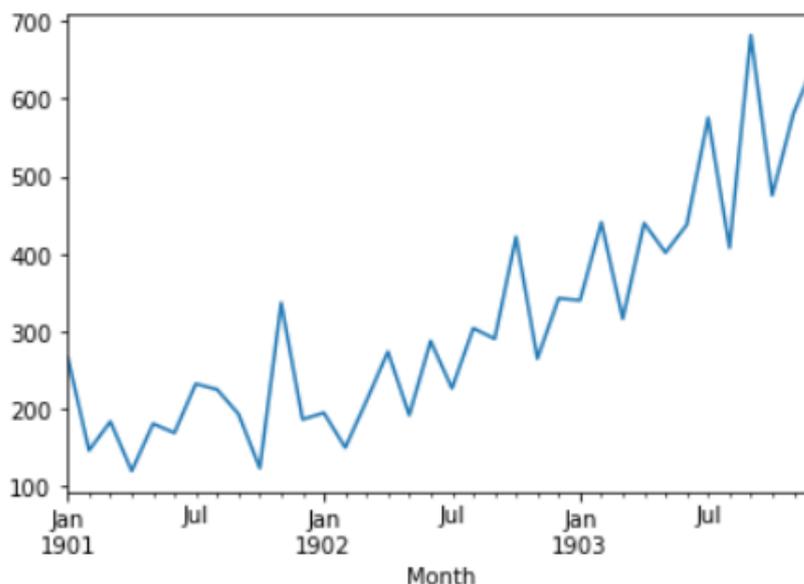


Рисунок 3 – График временных рядов

Для реализации метода устойчивости в Python нам нужно пройти 4 этапа:

1. Преобразовать одномерный массив данных в контролируемый набор для обучения.

2. Разделить датасет в 2 набора: набор для машинного обучения и набор для прогнозирования.

3. Определить модель устойчивости в среде Python.

4.Реализовать прогноз на основе модели устойчивости и имеющихся данных.

Шаг 1: преобразуем массив в контролируемую зону обучения.

При помощи кода, представленного на рисунке 4, изначальный набор данных преобразуется в 2 столбца, где $t-1$ – это входная переменная X , а $t+1$ – выходная переменная Y . Также данный код печатает первые 5 строк преобразованного датасета для наглядности (рис. 5).

```
from pandas import read_csv
from pandas import datetime
from pandas import DataFrame
from pandas import concat
from matplotlib import pyplot
from sklearn.metrics import mean_squared_error

def parser(x):
    return datetime.strptime('190'+x, '%Y-%m')

series = read_csv('shampoo-sales.csv', header=0, parse_dates=[0], index_col=0, squeeze=True, date_parser=parser)
# Преобразуем датасет
values = DataFrame(series.values)
dataframe = concat([values.shift(1), values], axis=1)
dataframe.columns = ['t-1', 't+1']
print(dataframe.head(5))
```

Рисунок 4 – Преобразование массива

	t-1	t+1
0	NaN	266.0
1	266.0	145.9
2	145.9	183.1
3	183.1	119.3
4	119.3	180.3

Рисунок 5 – Часть преобразованного массива

Шаг 2: разделяем датасет на наборы для машинного обучения и прогнозирования.

Разделим датасет на 2 набора. 66% данных дадим программе для обучения, оставшиеся 34% будут использованы для прогнозирования (рис. 6). Таким образом на завершающем шаге мы сможем сравнить, насколько точно справилась программа с прогнозом, по сравнению с имеющимися данными.

```
# Разделяем датасет на наборы
X = dataframe.values
train_size = int(len(X) * 0.66)
train, test = X[1:train_size], X[train_size:]
train_X, train_y = train[:,0], train[:,1]
test_X, test_y = test[:,0], test[:,1]
```

Рисунок 6 – Код для разделения массива

Шаг 3: определяем модель устойчивости.

Для этого мы определяем модель устойчивости в качестве функции, которая будет возвращать значения, предоставленные как входные данные.

```
# Определяем модель устойчивости
def model_persistence(x):
    return x
```

Рисунок 7 – Код для определения модели устойчивости

Шаг 4: реализовываем прогноз на основе модели устойчивости.
Применяем код (рис. 8) и получаем следующие результаты (рис. 9).

```
# Прогноз
predictions = list()
for x in test_X:
    yhat = model_persistence(x)
    predictions.append(yhat)

# Прогноз и исходные данные на одном графике
pyplot.plot(train_y)
pyplot.plot([None for i in train_y] + [x for x in test_y])
pyplot.plot([None for i in train_y] + [x for x in predictions])
pyplot.show()
```

Рисунок 8 – Код для реализации прогноза

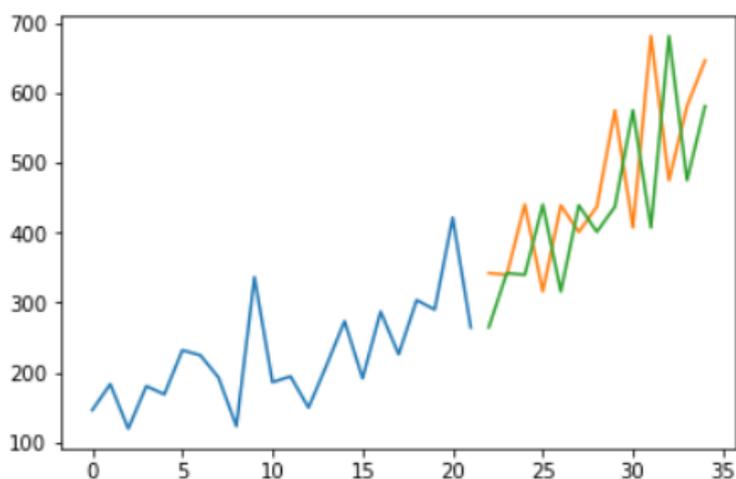


Рисунок 9 – Итоговый прогноз

Как видим, итоговый прогноз (отмечен оранжевым) по сравнению с исходными данными для сравнения (отмечены зелёным) имеют погрешность в один шаг, что позволяет в данном случае назвать применённые методы прогнозирования достаточно точными. Метод устойчивости в своем прогнозе показал примерно такие же показатели повышения уровня продаж, как и в данных, приведенных в датасете.

Таким образом, с помощью среды программирования Python удалось реализовать прогнозирование продаж на основе примера прогноза временных рядов.

Библиографический список

1. Куклина О. К., Михайлова Е. А. Яхина А. С. Программно-математическое обеспечение прогнозирования локальных показателей // Вестник Забайкальского государственного университета. 2015. №11. С. 97-104.
2. Передриенко А.И., Лютая Т.П., Харитонов И.М., Степанченко И.В. методы краткосрочного прогнозирования финансовых временных рядов с малыми объёмами выборки // Инженерный вестник Дона. 2020. №5. С. 9.
3. Тищенко С. А., Шахмурадян М. А. Методы машинного обучения в малом бизнесе: содержание и управление // Вестник Российского экономического университета им. Г.В. Плеханова. 2019. №6. С. 83-95.
4. Семенова В.Н., Гатин П.А. Программная реализация долгосрочного прогнозирования временных рядов на языке Python // Информатизация образования и науки. 2019. № 4 (44). С. 117-133.
5. Садовникова Н.А., Шмойлова Р.А. Анализ временных рядов и прогнозирование. М.: Издательский дом Университета "Синергия", 2016. 152 с.