

Прогнозирование спроса и управление запасами с помощью машинного обучения

А.А. Аббакумов, С.А. Ямашкин, В.Г. Карнетов, Д.С. Рощихин

Мордовский государственный университет им. Н.П.Огарёва

Аннотация: Данная статья посвящена изучению возможностей технологии машинного обучения для прогнозирования спроса на товары. В исследовании анализируются различные модели и возможности их применения в рамках задачи предсказания будущих продаж. Наибольшее внимание акцентировано на современных методах анализа временных рядов, в частности нейросетевые и статистические подходы. Полученные в ходе исследования результаты наглядно демонстрируют преимущества и недостатки разных моделей, степень влияния их параметров на точность прогноза в рамках задачи прогнозирования спроса. Практическая значимость полученных выводов определяется возможностью использования полученных результатов в анализе схожего набора данных. Актуальность исследования обусловлена необходимостью точного прогнозирования спроса на товары для оптимизации запасов и сокращения издержек. Использование современных методов машинного обучения позволяет повышать точность предсказаний, что особенно важно в условиях нестабильного рынка и изменяющегося потребительского спроса.

Ключевые слова: алгоритмы машинного обучения, оценка спроса, точность прогнозирования, анализ временных последовательностей, предсказание объемов продаж, Python, авторегрессионная интегрированная скользящая средняя, случайный лес, градиентный бустинг, нейронные сети, долгая краткосрочная память.

Введение

Текущая ситуация в мире на рынке логистики требует от компаний точных прогнозов и оперативной реакции на изменение спроса на различные товары, чтобы избежать хранения излишков на складе, но при этом не привести к дефициту [1]. В данной сфере ошибки критичны и могут привести к серьезным финансовым потерям [2], в то время как традиционные методы анализа и прогнозирование не учитывают всю сложность поведения конечного потребителя. С решением данной проблемы помогают справиться алгоритмы машинного обучения [3], которые позволяют производить наиболее точный анализ данных.

Современный рынок диктует новые правила, такие как переменчивость потребительских предпочтений, скоротечность трендов и нестабильная экономическая ситуация. В таких условиях бизнес наиболее остро нуждается

в инструментах способных не только к анализу большого количества данных потребителей, но и к адаптации постоянно изменяющейся среды [4].

Цель данного исследования заключается в определении наиболее эффективных моделей машинного обучения, в рамках выполнения задачи прогнозирования спроса на различные товары. В рамках статьи важно не просто определить прогноз какой из моделей окажется наиболее точным, но и рассмотреть какой подход будет наиболее предпочтительным в различных ситуациях.

Актуальность данного исследования обусловлена необходимостью поиска наиболее точных и гибких подходов, в рамках задачи прогнозирования спроса, которые в свою очередь позволят компаниям наладить эффективное управление запасами минимизировав при этом риски.

Методы прогнозирования спроса

Прогнозирование спроса — это не только расчет вероятности продаж, но и поиск закономерностей, которые определяют выбор тех или иных товаров покупателем [5]. На данный момент в рамках задачи прогнозирования спроса существуют три основных подхода [6]: статистические методы, машинное обучение и нейросетевые модели. Каждый из них обладает своими преимуществами и недостатками.

Статистические методы являются одним из первых инструментов прогнозирования. Их концепция заключается в построении прогноза, исходя из прошлых данных, предполагая при этом, что будущее будет похоже на прошлое. К примеру, модели авторегрессии позволяют изучать, как значение спроса в предыдущем влияют на текущую ситуацию. В свою очередь скользящая средняя помогает сгладить случайные колебания и выявить основные тренды. Данные методы наиболее эффективны в момент, когда спрос имеет сезонный или трендовый характер, но их применение

нецелесообразно в случае наличия резких изменений, обусловленных влиянием внешних факторов.

Методы машинного обучения способны анализировать десятки факторов одновременно, при этом выявляя скрытые зависимости [7]. Например, градиентный бустинг строит последовательность моделей, каждая из которых стремится минимизировать ошибки предыдущих повышая при этом точность прогноза. В то же время случайный лес основан на объединении предсказаний множества независимых деревьев решений [8]. Данные методы показывают большую эффективность в сравнении с статистическими моделями, поскольку помимо исторических данных позволяют учитывать внешние факторы

Нейросетевые модели отличаются возможностью обработки огромного количества данных [9], например, глубокие рекуррентные нейронные сети [10], такие как долгая краткосрочная память (Long Short-Term Memory - LSTM) способны запоминать длительные зависимости во временных рядах [11]. Сверточные нейросети, несмотря на изначальную ориентированность на обработку изображений, хорошо зарекомендовали себя в анализе временных данных и поиске сложных закономерностей. Эти технологии позволяют моделям обрабатывать огромные массивы информации и анализировать нестандартные данные, такие как отзывы в соцсетях и формировать предсказания с высокой точностью.

Сравнительный анализ методов прогнозирования спроса

Во время выполнения задачи прогнозирования редко удастся получить идеальные данные для работы, поэтому их важно приводить к удобному виду и очистить от шума. В рамках исследования мы используем сгенерированные тестовые данные, имитирующие продажи товара в течение нескольких лет. В рамках данных мы учитываем сезонные колебания, случайные и трендовые изменения спроса. Данные включают в себя дату, объем продаж, а также

случайные всплески спроса, отражающие рыночные колебания. График данных изображен на рис. 1. Моделирование данных выполним на языке программирования Python и приведем код (табл. 1) [12].

Таблица № 1

Программный код на Python для моделирования тестовых данных

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

def generate_sales_data(days=1000):
    np.random.seed(42)
    dates = pd.date_range(start='2020-01-01', periods=days, freq='D')
    trend = np.linspace(50, 200, days)
    seasonality = 20 * np.sin(np.linspace(0, 10 * np.pi, days))
    noise = np.random.normal(scale=10, size=days)
    sales = trend + seasonality + noise
    sales = np.maximum(sales, 0)
    return pd.DataFrame({'date': dates, 'sales': sales})

def prepare_data(df):
    df['day'] = df.index.dayofyear
    df['year'] = df.index.year
    X = df[['day', 'year']]
    y = df['sales']
    return train_test_split(X, y, test_size=0.2, random_state=42)

sales_data = generate_sales_data()
sales_data.set_index('date', inplace=True)
sales_data.index = pd.to_datetime(sales_data.index)
sales_data = sales_data.asfreq('D')
```

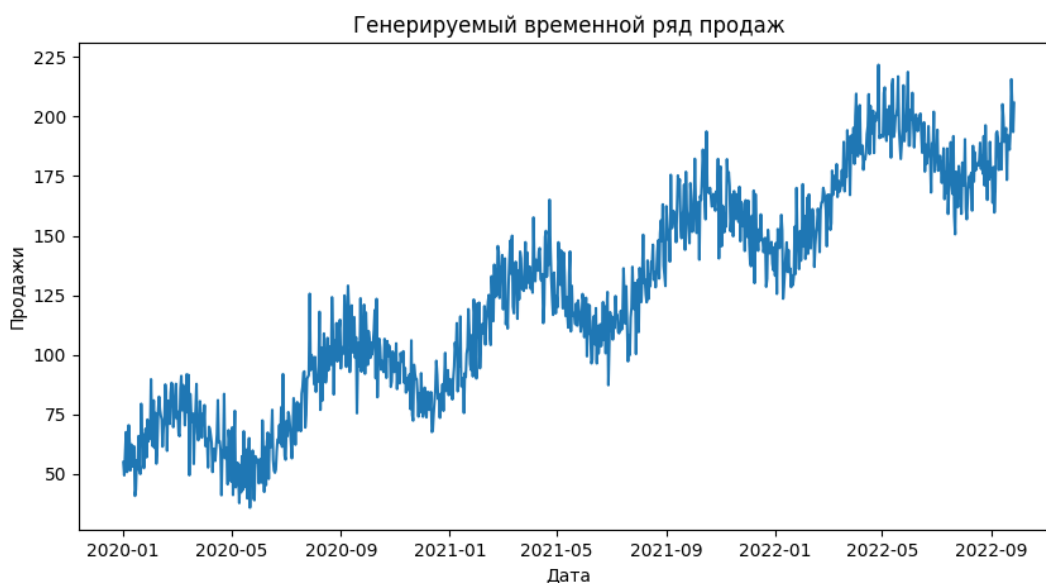


Рис. 1. – Результат выполнения программного кода для генерации тестовых данных

В результате выполнения программного кода мы получаем данные за несколько лет, с присущими им шумами. Далее нам необходимо протестировать как с нашими данными будут работать различные методы прогнозирования. Программный код для тестирования работы метода прогнозирования авторегрессионной интегрированной скользящей средней (Autoregressive Integrated Moving Average – ARIMA) (табл. 2).

Таблица № 2

Программный код модели ARIMA на языке Python

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from statsmodels.tsa.arima.model import ARIMA

sales_data = generate_sales_data()
sales_data.set_index('date', inplace=True)
sales_data.index = pd.to_datetime(sales_data.index)
sales_data = sales_data.asfreq('D')

X_train, X_test, y_train, y_test = prepare_data(sales_data)

train_size = int(len(sales_data) * 0.8)
train, test = sales_data.iloc[:train_size], sales_data.iloc[train_size:]

model = ARIMA(train['sales'], order=(5,1,0))
model_fit = model.fit()
predictions = model_fit.forecast(steps=len(test))
plt.figure(figsize=(10,5))
plt.plot(sales_data.index[-len(y_test):], y_test, label='Фактические значения')
plt.plot(sales_data.index[-len(y_test):], predictions, label='ARIMA прогноз',
linestyle='dashed')
plt.legend()
plt.title('Прогноз ARIMA')
```

Результат применения алгоритма ARIMA к тестовым данным представлен на рис. 2.

Исходя из полученного графика, изображенного на рис. 2, мы видим, что статистический алгоритм не подходит для наших тестовых данных, поскольку мы имеем случайные изменения спроса в наших тестовых данных,

что пагубно влияет на результат прогноза с использованием статистических методов.

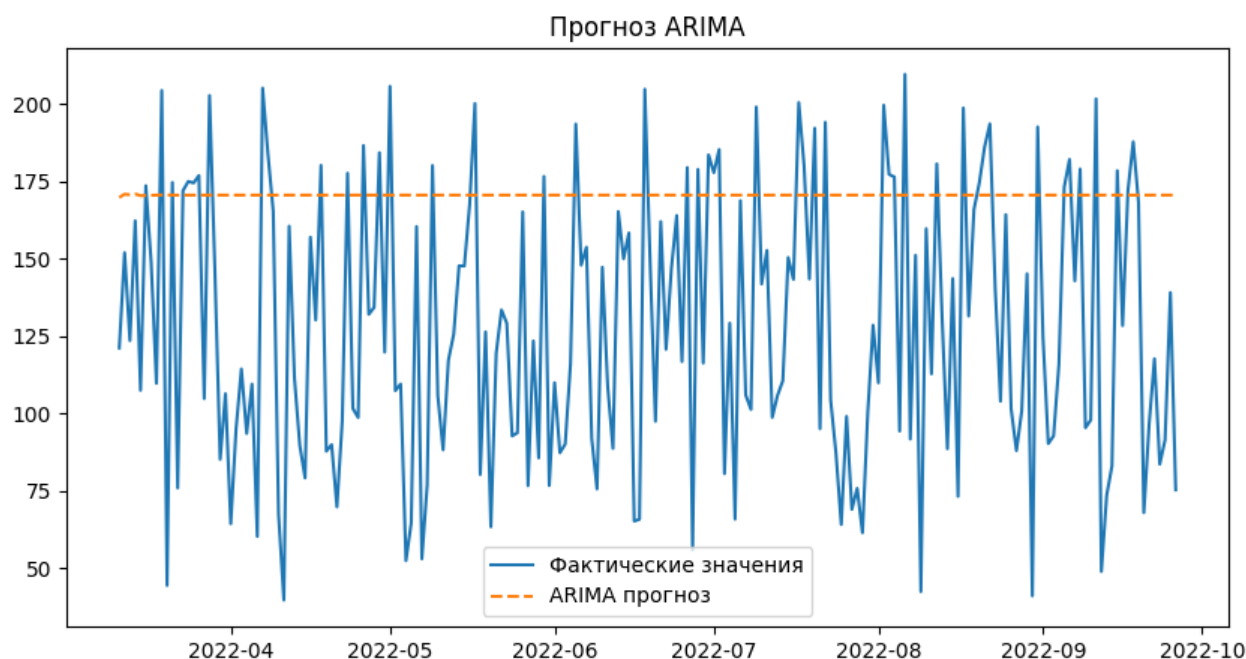


Рис. 2. – Результат выполнения программного кода для обработки тестовых данных с помощью ARIMA

Далее применим к нашим тестовым данным методы машинного обучения, такие как случайный лес и градиентный бустинг (табл. 3).

Таблица № 3

Программный код модели случайный лес и градиентный бустинг на языке Python

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.metrics import mean_absolute_error

sales_data = generate_sales_data()
sales_data.set_index('date', inplace=True)
sales_data.index = pd.to_datetime(sales_data.index)
sales_data = sales_data.asfreq('D')

X_train, X_test, y_train, y_test = prepare_data(sales_data)

rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
rf_predictions = rf_model.predict(X_test)
```

```
gb_model = GradientBoostingRegressor(n_estimators=100, learning_rate=0.1,
random_state=42)
gb_model.fit(X_train, y_train)
gb_predictions = gb_model.predict(X_test)

plt.figure(figsize=(10,5))
plt.plot(range(len(y_test)), y_test.values, label='Фактические значения',
color='black')
plt.plot(range(len(rf_predictions)), rf_predictions, label='Random Forest прогноз',
linestyle='dashed', color='blue')
plt.plot(range(len(gb_predictions)), gb_predictions, label='Gradient Boosting
прогноз', linestyle='dashed', color='red')
plt.legend()
plt.title('Прогноз моделей деревьев решений (Random Forest & Gradient Boosting)')
plt.xlabel('Дни')
plt.ylabel('Продажи')
```

Результат применения алгоритмов случайного леса и градиентного бустинга представлен на рис. 3.

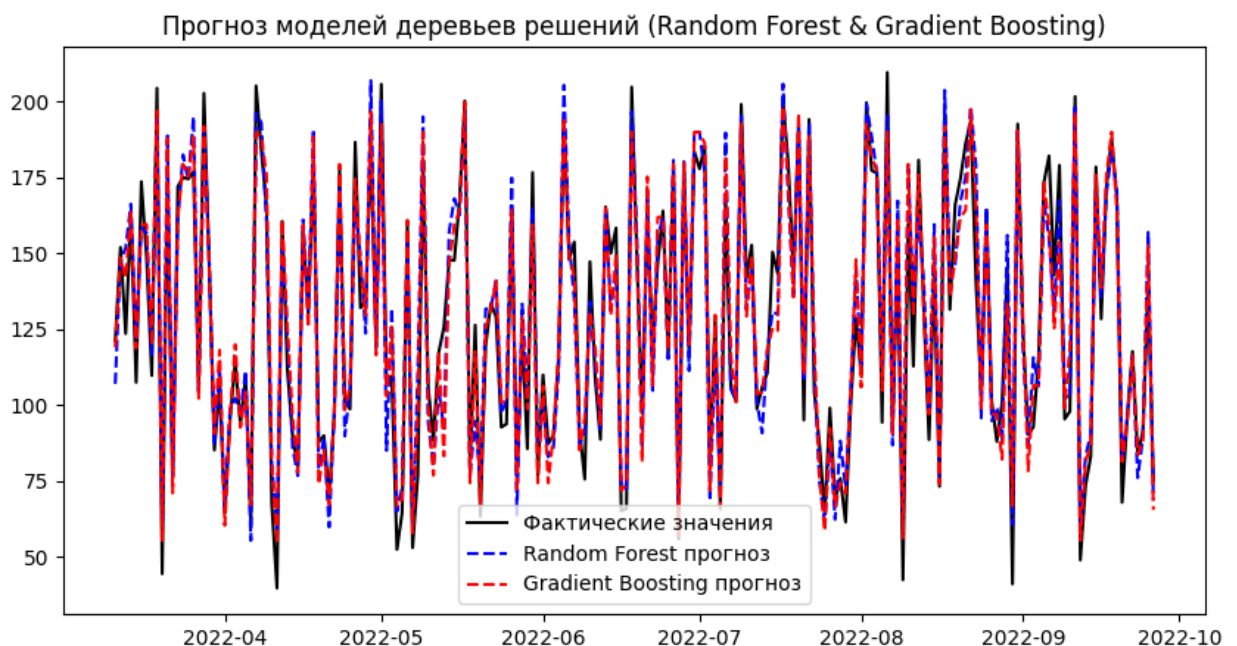


Рис. 3. – Результат выполнения программного кода для обработки тестовых данных с помощью случайного леса и градиентного бустинга

Проанализировав данный нам график на рис. 3, мы видим, что при использовании методов машинного обучения результирующий прогноз практически полностью коррелирует со входными данными, в отличие от применения статистического метода. Далее рассмотрим результат обработки наших входных данных с помощью LSTM (табл. 4).

Таблица № 4

Программный код модели LSTM на языке Python

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from sklearn.preprocessing import MinMaxScaler

sales_data = generate_sales_data()
sales_data.set_index('date', inplace=True)
sales_data.index = pd.to_datetime(sales_data.index)
sales_data = sales_data.asfreq('D')

X_train, X_test, y_train, y_test = prepare_data(sales_data)

scaler = MinMaxScaler()
scaled_sales = scaler.fit_transform(sales_data[['sales']])

def create_sequences(data, seq_length=10):
    X, y = [], []
    for i in range(len(data) - seq_length):
        X.append(data[i:i+seq_length])
        y.append(data[i+seq_length])
    return np.array(X), np.array(y)

X_lstm, y_lstm = create_sequences(scaled_sales)
X_lstm_train, X_lstm_test, y_lstm_train, y_lstm_test = train_test_split(X_lstm,
y_lstm, test_size=0.2, random_state=42)
lstm_model = Sequential([
    LSTM(50, activation='relu', return_sequences=True, input_shape=(X_lstm.shape[1],
1)),
    LSTM(50, activation='relu'),
    Dense(1)
])
lstm_model.compile(optimizer='adam', loss='mse')
lstm_model.fit(X_lstm_train, y_lstm_train, epochs=20, batch_size=16,
validation_data=(X_lstm_test, y_lstm_test))
lstm_predictions = lstm_model.predict(X_lstm_test)

plt.figure(figsize=(10,5))
plt.plot(range(len(y_lstm_test)), y_lstm_test, label='Фактические значения')
plt.plot(range(len(lstm_predictions)), lstm_predictions, label='LSTM прогноз',
linestyle='dashed')
plt.legend()
plt.title('Прогнозирование спроса с помощью LSTM')
plt.savefig("sales_plot4.png")
```

Результат применения алгоритмов случайного леса и градиентного бустинга представлен на рис. 4.

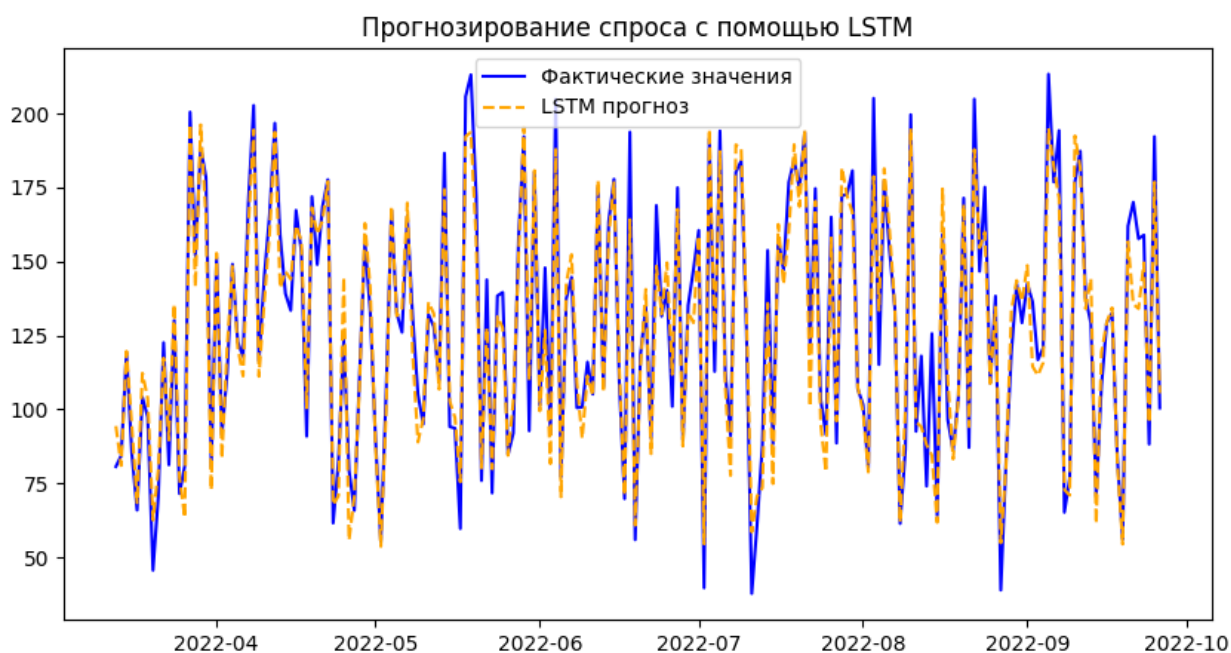


Рис. 4. – Результат выполнения программного кода для обработки тестовых данных с помощью LSTM

По результатам обработки тестовых данных с помощью LSTM, на рис. 4 можно наблюдать так же приближенные к тестовым данным прогнозы, схожие с методами машинного обучения.

В результате проведения сравнительного анализа можно прийти к выводу, что наименее подходящими являются статистические методы, они показали худший результат на тестовых данных. Что касается нейросетевых моделей и моделей, основанных на методах машинного обучения, мы получили результаты схожие с реальными. Однако, стоит отметить, что при гораздо больших объемах тестовых данных нейросетевые модели показывают себя наилучшим образом в рамках задачи прогнозирования спроса.

Заключение

В ходе исследования были рассмотрены различные методы прогнозирования спроса, включая статистические подходы, алгоритмы машинного обучения и нейросетевые модели. Мы сгенерировали тестовый

набор данных, приближенный к реальным данным, на котором и проверили все методы.

Результаты демонстрируют, что наибольшую точность прогнозов мы получили, используя модели машинного обучения и нейросетевые модели. Лучшие результаты из проанализированных моделей показал градиентный бустинг. Нейросетевые модели также продемонстрировали высокую точность, однако обучение таких моделей требовало больших ресурсов и времени.

Исходя из результатов, можно сделать вывод, что выбор метода прогнозирования обусловлен условиями, в которых он будет использован: для получения наибольшей точности без ограничения по ресурсам лучшими вариантами являются градиентный бустинг и LSTM. Если данные не содержат большого количества резких изменений и не требуют учета большого количества факторов, то статистические методы также применимы. Полученные результаты можно использовать в реальных сценариях прогнозирования спроса для оптимизации процессов закупки и хранения товаров.

Литература

1. Бунькина Н.И., Шевченко К.К. Разработка математической модели для решения задачи управления складом готовой продукции методом динамического программирования // Роль бизнеса в трансформации общества 2024 : сб. матер. XIX Междунар. конгр., Москва, 08–14 апр. 2024 г. Москва: Моск. фин.-пром. ун-т "Синергия", 2024. С. 326-330.

2. Карнетов В.Г., Ямашкин С.А. Применение искусственного интеллекта в логистике // IV Междунар. науч. конф. по междисциплинарным исследованиям : сб. ст., Екатеринбург, 15 дек. 2023 г. Екатеринбург: ООО "Институт Цифровой Экономики и Права", 2023. С. 414-418.

3. Забайкин Ю.В., Лютягин Д.В. Применение машинного обучения для прогнозирования спроса на строительные материалы и оптимизации цепочек поставок // Инновации и инвестиции, 2024. № 5. С. 499-503.

4. Тимохин М.Ю., Шаранин В.Ю. Искусственный интеллект и теория принятия решений: современные тенденции // Инженерный вестник Дона, 2023. №10. URL: ivdon.ru/ru/magazine/archive/n10y2023/8746/.

5. Дитман Т.А., Богатырева Я.А., Нордин В.В. Прогнозирование спроса на основе анализа временных рядов // Балтийский морской форум. Калининград, 2019. С. 628-634.

6. Makridakis S., Spiliotis E., Assimakopoulos V. The M4 Competition: 100,000 time series and 61 forecasting methods // International Journal of Forecasting, 2019. Vol. 36, № 1. P. 54–74.

7. Кротов А.Д., Яхонтова И.М. Прогнозирование спроса на товары и услуги с использованием алгоритмов машинного обучения // Проблемы развития науки и образования в эпоху модернизации: Материалы VIII Всерос. науч.-практ. конф., Ростов-на-Дону, 20 апр. 2023 г. Ростов-на-Дону: ООО "Издательство "Манускрипт", 2023. С. 110-112.

8. Пичугин О. Н., Прокофьева Ю. З., Александров Д. М. Деревья решений как эффективный метод анализа и прогнозирования // Нефтепромышленное дело, 2013. № 11. С. 69-75.

9. Ямашкина Е.О., Никулин В.В., Ямашкин С.А. Использование библиотеки TENSORFLOW для обучения нейронных сетей с целью анализа больших массивов пространственных данных // XLVIII Огарёвские чтения: Материалы науч. конф. Саранск, 06–13 дек. 2019 г. Ч. 1. Саранск: Национальный исследовательский Мордовский государственный университет им. Н.П. Огарёва, 2020. С. 505-510.

10. Федюшкин Н. А., Ямашкин С. А. Исследование репозиторий моделей нейронных сетей // Научно-технический вестник Поволжья, 2021. № 3. С. 28-30.

11. Nurfaizi A., Hasanuddin Ticket M. Prediction Using LSTM on a GLPI System // International Journal of Open Information Technologies, 2023. No. 7. P. 118-121.

12. Пономарев Д.С. Разработка web-приложения для предобработки данных с использованием библиотек Python // Инженерный вестник Дона, 2024. №8. URL: ivdon.ru/ru/magazine/archive/n8y2024/9414/.

References

1. Bunkina N.I., Shevchenko K.K. Razrabotka matematicheskoy modeli dlya resheniya zadachi upravleniya skladom gotovoj produkcii metodom dinamicheskogo programmirovaniya [Development of a mathematical model for solving the problem of managing a warehouse of finished products by dynamic programming]. Rol' biznesa v transformacii obshhestva 2024 : sb. mater. XIX Mezhdunar. kongr., Moskva, 08–14 apr. 2024 g. Moskva: Moskovskij finansovo-promyshlennyj universitet "Sinergiya", 2024. pp. 326-330.

2. Karnetov V.G., Yamashkin S.A. IV Mezhdunar. nauch. konf. po mezhdisciplinarnym issledovaniyam: sb. st., Ekaterinburg, 15 dek. 2023 g. Ekaterinburg: OOO "Institut Cifrovoy E`konomiki i Prava", 2023. pp. 414-418.

3. Zabaykin Yu.V., Lyutyagin D.V. Innovacii i investicii, 2024. No. 5. pp. 499-503.

4. Timokhin M.Yu., Sharanin V.Yu. Inzhenernyj vestnik Dona, 2023. No. 10. URL: ivdon.ru/ru/magazine/archive/n10y2023/8746/.

5. Ditman T.A., Bogatyreva Ya.A., Nordin V.V. Baltijskij morskoy forum: Materialy VII Mezhdunar. Baltijskogo morskogo foruma, Kaliningrad, 07-12 okt. 2019 g. Kaliningrad: Baltijskaya gos. akademiya rybopromyshlennogo flota, 2019. pp. 628-634.

6. Makridakis S., Spiliotis E., Assimakopoulos V. International Journal of Forecasting, 2019. Vol. 36, No. 1. pp. 54-74.
7. Krotov A.D., Yakhontova I.M. Problemy razvitiya nauki i obrazovaniya v epokhu modernizatsii: Materialy VIII Vseros. nauch.-prakt. konf., Rostov-na-Donu, 20 apr. 2023 g. Rostov-na-Donu: ООО "Izdatel'stvo "Manuskript", 2023. pp. 110-112.
8. Pichugin O.N., Prokof'eva Yu.Z., Aleksandrov D.M. Neftepromyshlovoe delo, 2013. No. 11. pp. 69-75.
9. Yamashkina E.O., Nikulin V.V., Yamashkin S.A. Ispol'zovanie biblioteki TENSORFLOW dlya obucheniya nejronnykh setej s tsel'yu analiza bol'shikh massivov prostranstvennykh dannykh [Using the TENSORFLOW library for training neural networks to analyze large amounts of spatial data]. XLVIII Ogaryovskie chteniya: Materialy nauch. konf., Saransk, 06–13 dek. 2019 g. Saransk: NI Mordovskij gos. universitet im. N.P. Ogaryova, 2020. pp. 505-510.
10. Fedyushkin N.A., Yamashkin S.A. Nauchno-tehnicheskij vestnik Povolzh'ya, 2021. No. 3. pp. 28-30.
11. Nurfaizi A., Hasanuddin Ticket M. International Journal of Open Information Technologies, 2023. No. 7. pp. 118-121.
12. Ponomarev D.S. Inzhenernyj vestnik Dona, 2024. No. 8. URL: ivdon.ru/ru/magazine/archive/n8y2024/9414.

Дата поступления: 22.04.2025

Дата публикации: 25.05.2025