Annals of "Dunarea de Jos" University of Galati
Fascicle I. Economics and Applied Informatics
Years XXIX – nº2/2023
ISSN-L 1584-0409 ISSN-Online 2344-441X
www.eia.feaa.ugal.ro

DOI https://doi.org/10.35219/eai15840409345

Python for e-Commerce

Maria Cristina ENACHE*

ARTICLE INFO

Article history:
Accepted July 2023
Available online July 2023
JEL Classification
K22, M21

Keywords:
Python, e-commerce, business

ABSTRACT

Python is widely used and has a strong presence in the e-commerce industry. Python's popularity and usage in e-commerce have been steadily growing over the years Predicting the exact future evolution of e-commerce and the role of Python is challenging. However, there are several trends and areas where Python is likely to continue making an impact in the e-commerce domain. In this article I tried to highlight the most common uses of python in e-commerce, with examples of scripts for analyzing customer behavior, but also some hints on how it could be used in the future.

1. Introduction

Today, e-commerce is a growing, multibillion-dollar part of the economy. As the online market began to expand more and more quickly, this gradually began to be seen in the development of e-commerce, but it really exploded with the widespread use of mobile phones, and customers use as the first option online shopping. Ease of use has led to a great increase in popularity for Python and PHP languages to create e-commerce websites. Entrepreneurs nowadays prefer Python development services more to develop their business as it is more advantageous than other programming languages for e-commerce development.

The development of a business in the online environment implies that the entrepreneur can call on technologies for rapid, efficient development and with diverse and yet customer-oriented functions, for a personalized shopping experience. E-commerce web development with Python attracts the attention of many entrepreneurs and choosing programming language like Python is a programming language that stands out for its efficiency, due to its simplicity, suitable for beginners because it is efficient enough to allow them to learn quickly but also to quickly create complex applications.

Currently, there are approximately 140,284 functional websites that have Python code behind them. We can note that this programming language was used for the development of 2.65% of the first 10,000 websites on the Internet. Thus, the Python language is currently the most sought-after technology, especially in electronic commerce. It is estimated that at least 24.5% of the total sales volume until 2025 will take place using e-commerce platforms and these will be accessed from smart devices such as smartphones. Thus, the construction of an e-commerce website accessible from devices such as smartphones will require the use of the right technologies and that is why development with the Python language is a good choice in terms of flexibility and scalability already proven in the management of the specific activities of an e-commerce store.[3]

A major advantage of using Python for e-commerce is that it is a powerful programming language that appeals through the simplicity of its code. More and more companies prefer Python for e-commerce because there are many advantages that reinforce the idea that Python is one of the programming languages that stand out by [1]:

- **Saves development time and effort** - because Python does not require developers to create many lines of code to describe concepts, unlike other programming languages. With Python, many complex e-commerce applications can now be created without writing many lines of code to help customers reduce both development costs and the time required to update and develop these applications.
- **It has many libraries and extensions** - Web development of e-commerce applications with Python with efficient, faster, and secure but scalable frameworks must be accompanied by the existence of easily available libraries and solutions.
- **Transport methods -** An essential component of any e-commerce business is shipping. No matter how good the products are, delivery options can easily damage the reputation of an online business. A customer will be hostile to the business if the purchased products are damaged or delivered late.

*Dunarea de Jos University of Galati, Romania. E-mail address: mpodoleanu@ugal.ro.

Python facilitates the integration of delivery services or offering customers the possibility of making specialized deliveries, depending on the size of their order.

- **Create a scalable online store -** Expanding the business, increasing the number of users can be a challenge for already developed websites. Python can help the e-commerce platform to increase the number of accesses, without decreasing the loading speed of the platform or the functionality of the site.

2. Phyton for e-commerce

In recent years, deep learning has had a significant impact on e-commerce, changing the way companies interact with customers and how they make purchasing decisions. Deep learning is a field of artificial intelligence that uses machine learning algorithms to identify complex patterns in input data. By using these models, companies can better understand customer behavior and preferences and provide a more personalized shopping experience.

One of the most important advantages of deep learning in e-commerce is its ability to create personalized recommendations. By using machine learning algorithms, companies can analyze a customer's previous buying behavior and make relevant recommendations for products or services that would suit them best. For example, Amazon uses a deep learning-based recommendation system that analyzes customers' purchase history and preferences to make personalized product recommendations [2].

In addition, deep learning can also be used to improve the overall customer shopping experience. Companies can use machine learning algorithms to analyze feedback left by customers and identify problems or areas for improvement in the buying process. For example, an online store can use sentiment analysis to identify customer comments complaining about delivery times or problems with the website, and then take steps to improve those problems.

Here is an example of a script that an online store could use to identify customer comments through sentiment analysis:

```
pythonCopy code
import pandas as pd
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
# Import the csv file
customer_reviews = pd.read_csv("client_review.csv")
# Create an instance of the VADER sentiment analyzer
sia = SentimentIntensityAnalyzer()
customer_reviews['Sentiment Score'] = customer_reviews['Comentariu'].apply(lambda x:
sia.polarity_scores(x)['compound'])
```

```
# Label each comment as positive, neutral, or negative
client_review['Sentiment Label'] = client_review['Sentiment Score'].apply(lambda x: 'positive'
if x > 0 else ('neutral' if x == 0 else 'negative'))
# Export data to a CSV file
client_review.to_csv("client_review_sentiment.csv", index=False)
```

This script imports a dataset containing comments left by customers of an e-commerce store and analyzes the sentiment of each comment using the VADER sentiment analyzer. The code will add a column for the sentiment scores of each comment and label each comment as positive, neutral, or negative based on the sentiment score. After the data is processed, the script exports the data to a CSV file for later analysis. This file can be loaded into an analysis program to identify problems or areas for improvement in the buying process and act accordingly.

It is important to note that this script is only an example and should be adapted to the specific needs and data of each online store. Security and customer data protection aspects must also be considered to avoid any breach of their privacy.

Deep learning can also be used to identify credit card fraud. By using machine learning algorithms, companies can identify suspicious patterns in purchase transactions and block transactions that appear to be fraudulent. For example, PayPal uses a fraud detection system based on deep learning that analyzes transactions and customer behavior to identify suspicious transactions.

However, there are also challenges associated with using deep learning in e-commerce. One of the main issues is ensuring the security of customer data. Businesses must be mindful of how customer data is collected, stored, and used and take steps to ensure that it is protected against unauthorized access or loss.

To create a personalized offer for a customer based on a deep learning approach, we need a range of data about the customer and their preferences. This may include information such as purchase history, website browsing behavior, feedback left by previous customers, personal preferences, and other relevant data. We will use this data to build a deep learning model that will be able to predict the customer's individual preferences and needs and provide a personalized offer for it.

Here is a possible script for such an approach:

```
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow.keras import layers
# Read data
client_data = pd.read_csv('client_data.csv')
# Preprocess the data
# Define the deep learning model
model = tf.keras.Sequential([
layers.Dense(64, activation='relu', input_shape=[len(client_data.columns)]),
layers.Dense(64, activation='relu'),
layers.Dense(1)
])
# Compile the model
model.compile(loss='mean_squared_error',
 optimizer=tf.keras.optimizers.Adam(0.001))
# Train the model
X_train = client_data.drop(columns=['oferta'])
y_train = client_data['oferta']
model.fit(X_train, y_train, epochs=100, batch_size=32)
# Make a prediction
client_data = client_data.iloc[0].values.reshape(1, -1)
oferta = model.predict(client_data)
print("Oferta personalizată pentru clientul dat este: ", oferta)
```

This script assumes that we have customer data in a CSV file called client_data.csv, which contains a column for the offer and several columns for data about the customer and their preferences. The deep learning model used is a simple model with two dense layers and a ReLU activation function. In addition, a mean squared error (MSE) loss and an Adam optimizer are used to compile the model. After the model has been trained, we use the data about a particular customer to make a prediction about the personalized offer for that customer.

ReLU (Rectified Linear Unit) is a non-linear activation function used in artificial neural networks. This is defined as relu(x) = max (0, x), where x is the input.

The ReLU function is a simple and efficient function that can be used in many types of neural networks, such as convolutional neural networks and fully connected neural networks. ReLU is preferred compared to other activation functions such as sigmoid function or tanh function because it reduces the gradient problem that occurs while training deep neural networks.

By using the ReLU function, negative values in the input are transformed to zero, while positive values are left unchanged. This makes the ReLU function simple, fast, and computationally efficient because it eliminates the need to use the exponential in the calculations.

The ReLU function is often used in the hidden layers of neural networks because it can help speed up network training and help prevent overfitting. However, using ReLU can cause the gradient to vanish for negative input values, which can slow down neural network training for specific architectures. Therefore, other activation functions have been introduced, such as Leaky ReLU, which eliminates this problem by using a small slope for negative input values.

Here is an example of a deep learning-based script that analyzes the purchase history of a mobile phone store to recommend other products:

```
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, LSTM
from sklearn.preprocessing import MinMaxScaler
# Import history data
comenzi = pd.read_csv("comenzi.csv")
# Select produs column
comenzi_produs = comenzi['Produs']
# Transform the data for the LSTM model
scaler = MinMaxScaler(feature_range=(0, 1))
comenzi_produs = scaler.fit_transform(np.array(comenzi_produs).reshape(-1, 1))

# Create input and output for the LSTM model
seq_len = 3
X = []
y = []
for i in range(seq_len, len(comenzi_produs)):
   X.append(comenzi_produs [i-seq_len:i, 0])
   y.append(comenzi_produs [i, 0])
X = np.array(X)
y = np.array(y)
# Split the data into sets for training and test
train_size = int(len(X) * 0.8)
X_train, X_test = X[:train_size], X[train_size:]
y_train, y_test = y[:train_size], y[train_size:]
# Build the LSTM model
 model = Sequential()
 model.add(LSTM(units=50, return_sequences=True, input_shape=(X_train.shape[1], 1)))
 model.add(Dropout(0.2))
 model.add(LSTM(units=50, return_sequences=True))
 model.add(Dropout(0.2))
 model.add(LSTM(units=50))
 model.add(Dropout(0.2))
 model.add(Dense(units=1))
 # Compile and train the model
 model.compile(optimizer='adam', loss='mean_squared_error')
 model.fit(X_train, y_train, epochs=100, batch_size=32)
# Make product recommendations
last_seq = comenzi_produs [-seq_len:].reshape(1, seq_len, 1)
 next_product = scaler.inverse_transform(model.predict(last_seq))[0][0]
 produs_recomandat = comenzi[comenzi['Produs'] == next_produs]['Produs'].unique()
 print("Produse recomandate: ", produs_recomandat)
```

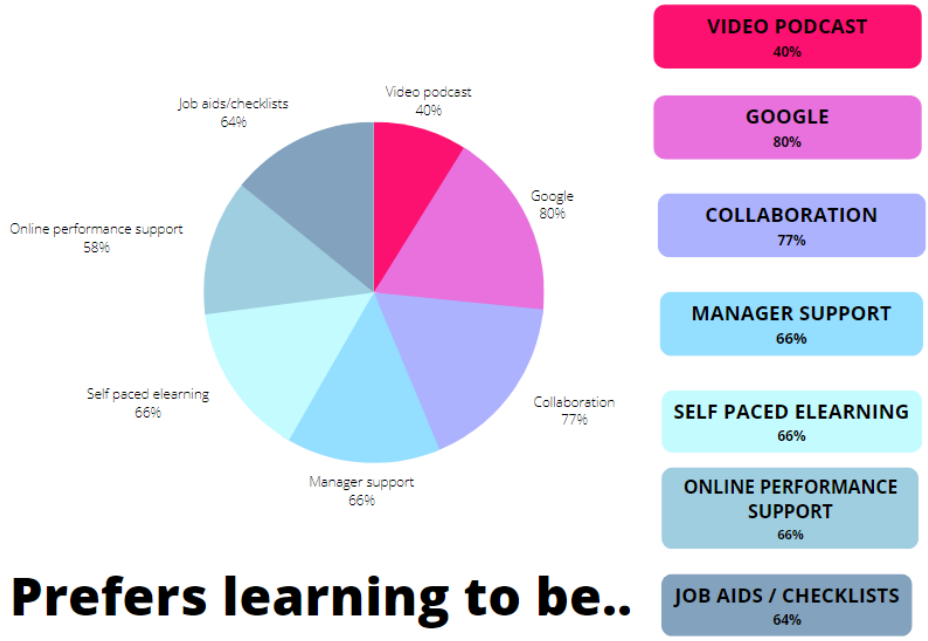
This script uses a Long Short-Term Memory model, i.e. a type of recurrent neural network used to process and analyze sequential data, to analyze the purchase history of an e-commerce store selling mobile phones and to make new product recommendations. In this example, we will only consider the last three products purchased to predict the next product that might be purchased. This model was first used by Hochreiter and Schmidhuber in 1997, and became well-known and widely used for its ability to solve gradient recurrence problems.

The LSTM model is built from LSTM units, which have three main gates: input gates, forget gates, and output gates. Input gates control the flow of data inside the LSTM unit, forget gates decide what information to keep or forget, and output gates determine what information to send to the next time step. LSTM units can retain important information for longer periods of time due to the use of forget gates. These gates allow LSTM units to decide which information to keep and which information to forget based on its meaning in the context of the input sequence.[5]

The LSTM model is used in many fields, among which we can highlight natural language processing and time series analysis. The LSTM model is more used in situations where the data must be analyzed in more detail, or when the analysis is done on some sequential data, as it has the ability to handle gradient recurrence situations and to retain important information for longer periods of time

Before training the LSTM model, the history data is processed using a Min - Max scaling and transformed into a suitable format for the LSTM model. The LSTM model is built using three LSTM layers and a dense layer with dropout to prevent overfitting.

After the model is trained, the script makes predictions for the next product to be recommended.



3. The role of Python in future evolution of e-commerce

Predicting the exact future evolution of e-commerce and the role of Python is challenging. However, there are several trends and areas where Python is likely to continue making an impact in the e-commerce domain:

- AI-Powered Personalization: Personalization is a key focus in e-commerce, and Python's robust machine learning and AI libraries will continue to play a significant role in building AI-powered recommendation systems, customer segmentation models, chatbots, and sentiment analysis tools. Python's libraries like scikit-learn, TensorFlow, and PyTorch will enable e-commerce businesses to deliver personalized and engaging experiences to their customers.
- Enhanced Security and Fraud Detection: As security threats and online fraud continue to be challenges in e-commerce, Python can be leveraged for developing robust security measures and fraud detection systems. Python's extensive libraries for cryptography, data analysis, and machine learning can be utilized to build intelligent fraud detection algorithms and secure payment gateways.
- Voice Commerce and Chatbots: With the rise of voice assistants and chatbots, Python can be utilized to build conversational interfaces for e-commerce. Libraries like NLTK and spaCy can be employed for natural language processing, enabling voice-based search, voice-activated shopping, and chatbot-based customer support.
- Data Analytics and Business Intelligence: Python's data processing and analysis libraries, such as pandas and NumPy, will continue to be instrumental in extracting insights from large volumes of e-commerce data. Python's integration with data visualization libraries like Matplotlib and Plotly will aid in generating meaningful visualizations and dashboards for data-driven decision-making in e-commerce.
- Progressive Web Applications (PWAs): PWAs are gaining popularity in e-commerce due to their cross-platform compatibility and enhanced user experience. Python-based web frameworks like Django and Flask can be utilized to build PWAs that provide seamless offline access, push notifications, and app-like experiences to customers.
- Cloud-Native Development: Cloud-native development, utilizing services like AWS, Azure, and Google Cloud, is becoming increasingly prevalent in e-commerce. Python's compatibility with cloud platforms, along with its libraries for cloud computing and containerization, will enable efficient deployment, scaling, and management of e-commerce applications in the cloud.

4. Conclusions

It's important to note that these are potential areas where Python may continue to evolve in e-commerce, but the actual evolution will depend on various factors, including industry trends, technological advancements, and business requirements. Python is already a language of the present, but it is also developing, because there is a special interest for this to happen, as a whole network of developers

contributes to its growth. It has already proven its usefulness in fields such as data science, creating applications for desktop computers, or web applications. With this article we wanted to show how useful it is for e-commerce. Within e-commerce platforms, the Python language is proven useful in that it can be used to add a recommendation component for customers and desired products, creating an app for the store and synchronizing it with the e-commerce platform, In a situation of technological elution and increased competitiveness Python offers many advantages, through the possibility of expanding the e-commerce platform with new functionalities.

References

1. *Yi-Ou, L.- Deep learning in NLP: Methods and applications. Journal of University of Electronic Science and Technology of China*
2. *Shardul Bhatt - Python for eCommerce Development -*
3. *https://www.botreetechnologies.com/blog/python-for-ecommerce-development-why-does-it-make-sense/*
4. *Wenhu Yu - Multi-level Deep Learning based E-commerce Product Categorization*
5. *https://ceur-ws.org/Vol-2319/ecom18DC_paper_2.pdf*
6. *Shipra Saxena-Learn About Long Short-Term Memory (LSTM) Algorithms*
7. *https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/*
8. *Zhou, L.- Product advertising recommendation in e-commerce based on deep learning and distributed expression. Electron Commer Res 20, 321–342 (2020). https://doi.org/10.1007/s10660-020-09411-*
9. *https://dl.acm.org/doi/abs/10.1145/3298689.3347006 A deep learning system for predicting size and fit in fashion e-commerce*
10. *Data Analysis and Prediction Modeling Based on Deep Learning in E-Commerce https://www.hindawi.com/journals/sp/2022/1041741/*
11. *Dhruv Patel - Why Choose Python for eCommerce Platform?*
12. *https://www.linkedin.com/pulse/why-choose-python-ecommerce-platform-dhruv-patel?trk=pulse-article*
13. *The Necessity of E-commerce website -*
14. *https://www.sovereignconsult.com/blog/need-for-ecommerce-website/*