

ML Enabled WhatsApp Chatbot using IBM Watson

Manasi Chhibber* and Madhulika Bhadauria

Amity School of Engineering & Technology, Amity University, India

***Corresponding Author:** Manasi Chhibber, Amity School of Engineering & Technology, Amity University, India.

Received: May 04, 2022; **Published:** May 24, 2022

DOI: 10.55162/MCET.02.045

Abstract

A chatbot mimics human interaction through voice or text interactions. Chatbots are used by businesses to interact with consumers in addition to the traditional client care channels like via phones, mails or social media. The most basic type of chatbot system handles jobs by interpreting user input and then searching its directory for content related to particular words or phrases. In a nutshell, it works as a keyword-based content retrieval system. Watson Studio provides a collaborative environment and resources to work on data in order to solve business challenges. Users may select tools for analyzing and displaying their data, cleaning and shaping it and using it for constructing and training machine learning models. Watson Assistant is a business-oriented AI chatbot. This enterprise AI technology can be used to create conversational AI applications. Since, loan application process can be extremely cumbersome, so we have developed a chatbot using IBM Watson that could help in easing this process. The chatbot has been integrated with WhatsApp using Twilio. By answering a series of standard questions related one's application, one can see the risk percentage involved in taking the loan. This has been enabled using Machine Learning models.

Keywords: chatbot; machine learning; IBM Watson; loan risk prediction

Introduction

The main aim of the work was to build a chatbot that could be used easily by people for assistance [1] regarding loan applications. If done traditionally, this procedure can be really time-consuming that clients frequently migrate to other rivals, leading to the loss for the bank. However, we have employed machine learning technologies [10] to minimize the time it takes to make these judgments and raise the accuracy of the decisions made, resulting in a growth in our client base. Machine learning [8], when used correctly, may also help to reduce prejudice. Furthermore, chatbots may be used to boost productivity and deliver a better customer experience. Chatbots may improve the customer experience at multiple points by answering quickly and effectively to commonly requested inquiries or by smoothly elevating requests to a human agent with the full context of the enquiry [2]. The complete process for loan application has been automated using:

- Watson Studio to assist in the generation of ML pipelines and the training of the model to improve predictions [5, 7].
- IBM Cloud [9] Functions to allow execution of the application code without servers and automatically scale it.
- The Watson Assistant [12] chatbot's ability to connect to WhatsApp through Twilio. Watson Assistant also aids in the expansion of the user base by making the system more accessible.

In the past, there has been some research in this area. A M Rahman, Abdullah Al Mamun and Alma Islam (2017) studied various chatbot building platforms and explored their capabilities. Sasha Fathima Suhel, Vinod Kumar Shukla, Sonali Vyas and Ved Prakash Mishra (2020) researched about the possibility of Conversion to automation in banking sector through chatbots using Artificial Machine Intelligence [8, 11]. Sewoong Hwang and Jonghyuk Kim (2021) examined an AI chatbot that could improve one's financial stability and

provide policy solutions when required. Although these studies are relevant, but our solution is not just smart as it's build using ML, also it is very convenient to use.

Materials and Methods

Some prerequisites that were required are listed below:

- An IBM Cloud account
- A Twilio account

The step-by-step method followed for making the chatbot has been described below:

Creating the Watson Studio service

We first started by creating a Watson Studio [12] instance and associated a cloud storage instance with it as shown in Figure 1. Then, we created a new project as shown in Figure 2 and uploaded the dataset that would be required by the Machine Learning model [10] for training like in Figure 3.

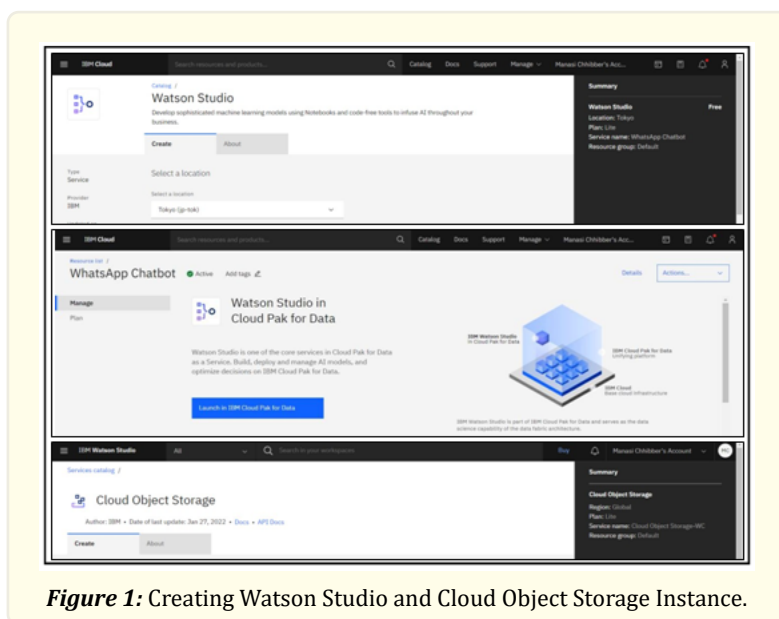


Figure 1: Creating Watson Studio and Cloud Object Storage Instance.

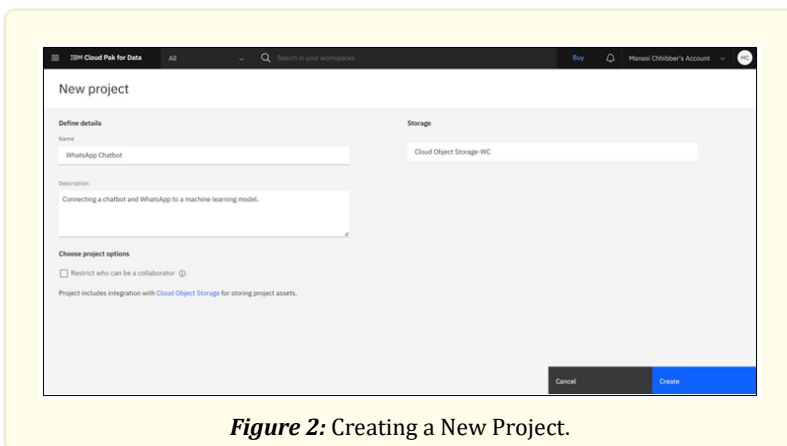


Figure 2: Creating a New Project.

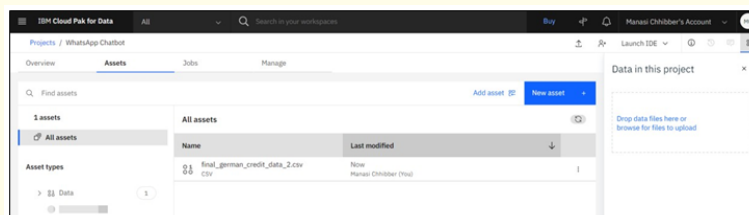


Figure 3: Uploading Dataset.

Creating and running an AutoAI experiment

We then moved on to the Machine Learning portion. We created a new experiment as depicted in Figure 4. We then set the target attribute as shown in Figure 5. The prediction type was chosen as shown in Figure 6. After finalizing the experiment settings, the model was trained on the dataset. Once trained, the metric chart and other model specifications could be viewed as shown in Figure 7. Here, we also visualized the performance of the model that gave us the most accurate results as displayed in Figure 8.

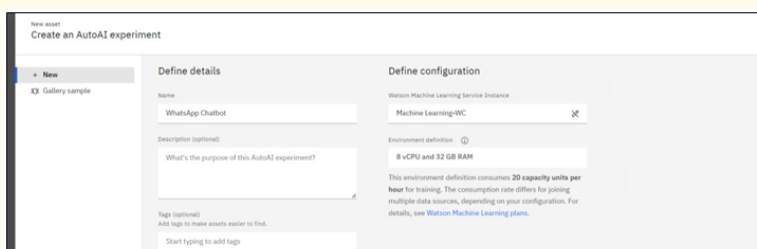


Figure 4: Creating a New Experiment.

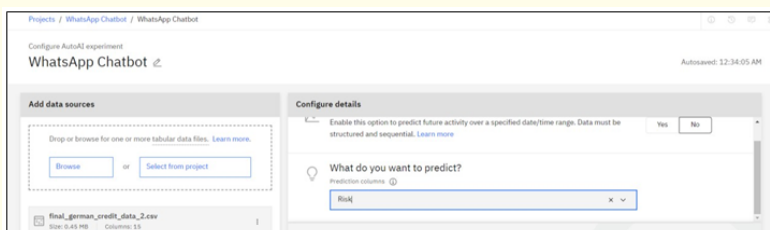


Figure 5: Setting Target Attribute.

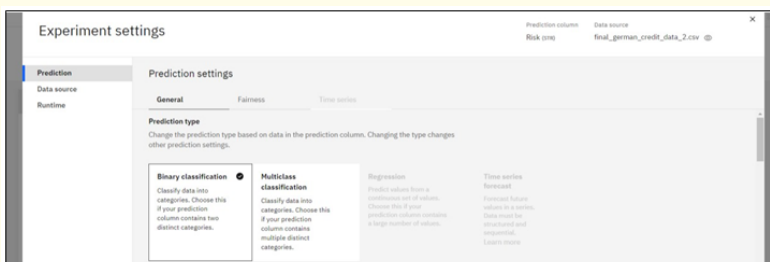
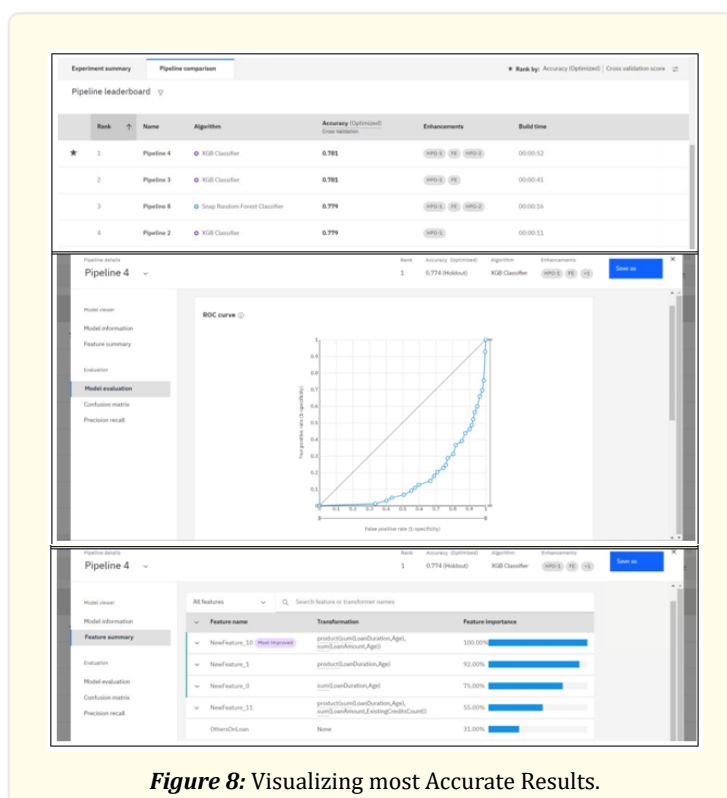


Figure 6: Choosing Classifier Type.



Creating a deployment and testing the model

As our model was ready, we then moved forward to the deployment phase. We create a deployment space first as shown in Figure 9.

The deployment of XGB model [13] was done, as it gave us the most accurate results. A direct online link for the endpoint of our model was generated as shown in Figure 10.

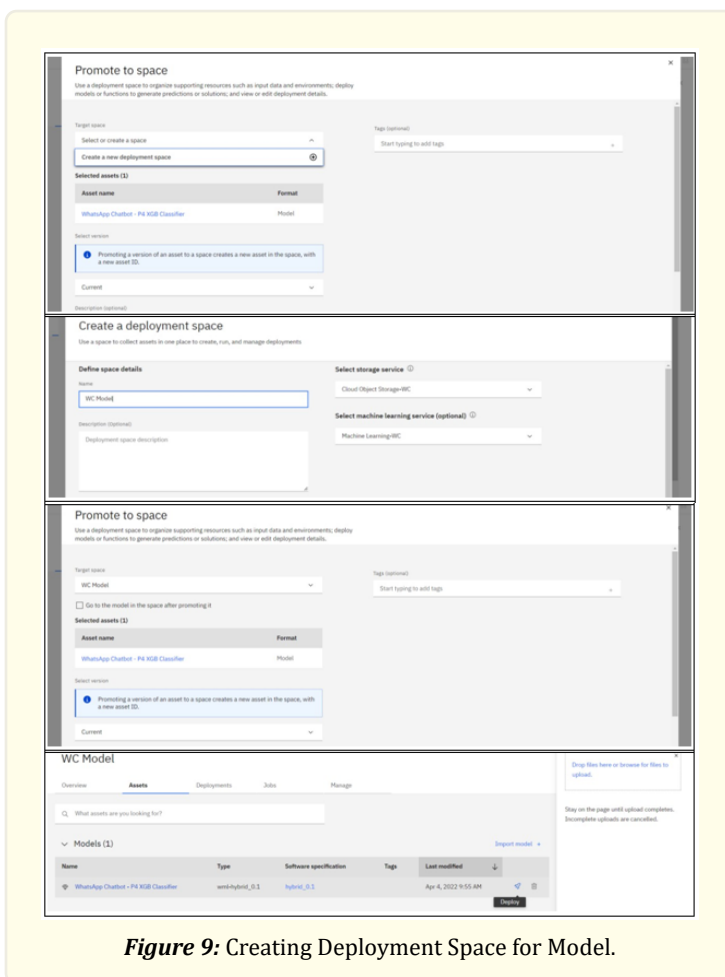


Figure 9: Creating Deployment Space for Model.

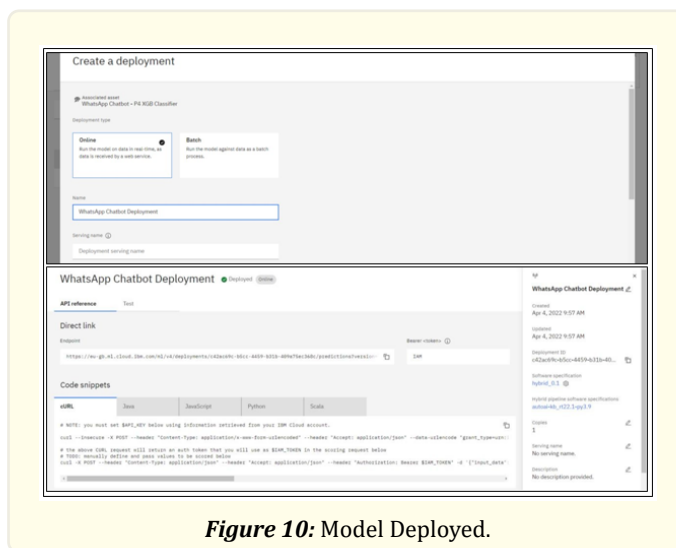


Figure 10: Model Deployed.

Setting up IBM Cloud functions

To use the chatbot service elsewhere, an API key was required. It was generated by hovering to the API keys option as shown in Figure 11. To integrate this with our chatbot, IBM Cloud Functions [6] were used as depicted step by step in Figure 12. The function code was written, and the API was linked as shown in Figure 13.

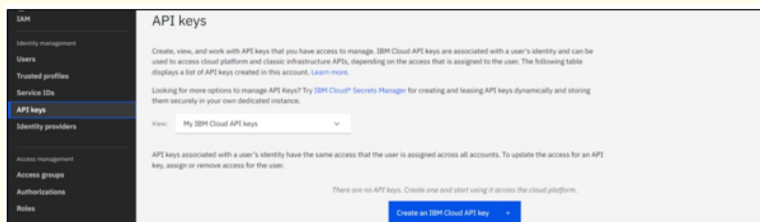


Figure 11: Creating an API Key.

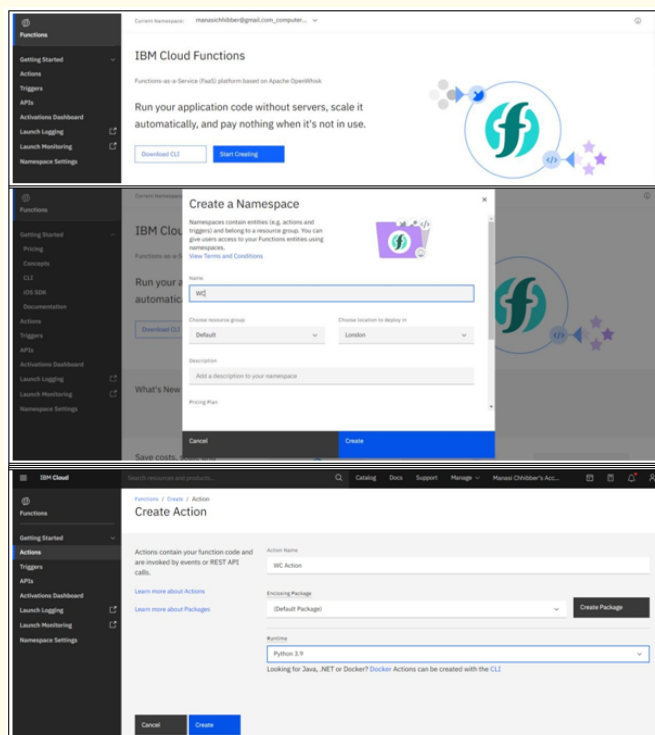


Figure 12: Setting up Cloud Function.

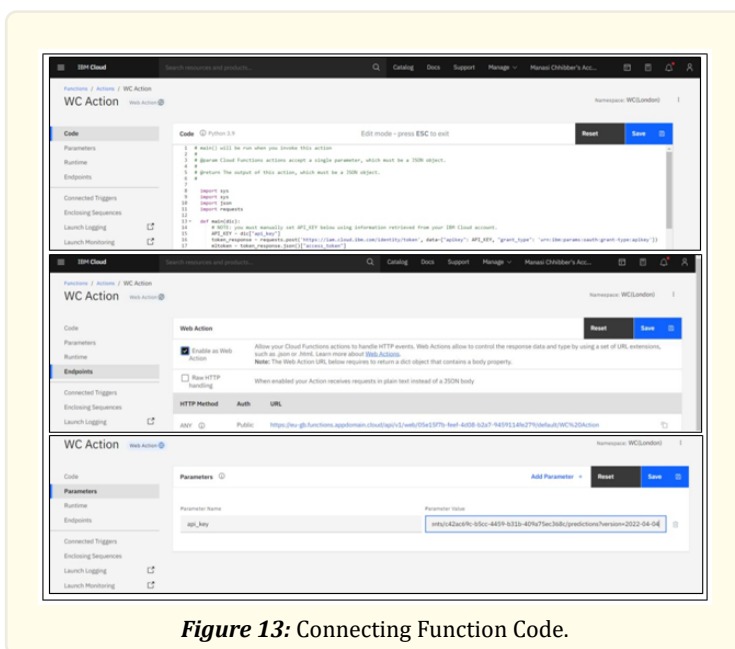


Figure 13: Connecting Function Code.

Integrating the machine learning model with Watson Assistant

A new Watson Assistant was created and a json file consisting of its dialog specifications was uploaded as shown in Figure 14.

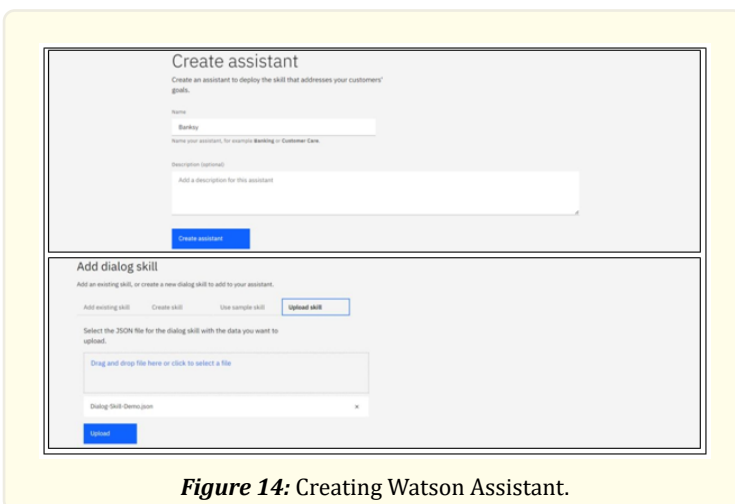


Figure 14: Creating Watson Assistant.

Integrating IBM Watson Assistant with WhatsApp

To set up WhatsApp, a series of simple steps like the ones shown in Figure 15 were followed. The specifications of the chatbot were entered and it was good to go.



Figure 15: Setting up WhatsApp.

Configuring and using Twilio sandbox

At last, we opened Twilio, entered our chatbot's API key as shown in Figure 16 and the phone numbers where we wanted the chatbot to reply to conversations and finally it was ready to be activated on WhatsApp.



Figure 16: Using Twilio to setup the Chatbot.

Results and Discussion

To use the chatbot, named Banksy, all you need to do is to ping on +1 415 523 8886 with code 'join directly- obtain' and you'll be set. After answer a series of question, you'll be able to see the risk probability of your application. Figure 17, Figure 18 and Figure 19 describe the complete process.

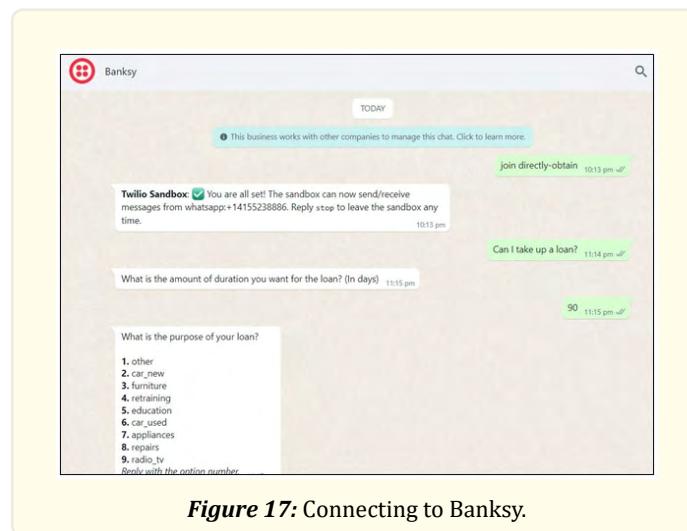


Figure 17: Connecting to Banksy.



Figure 18: User providing Loan Details.

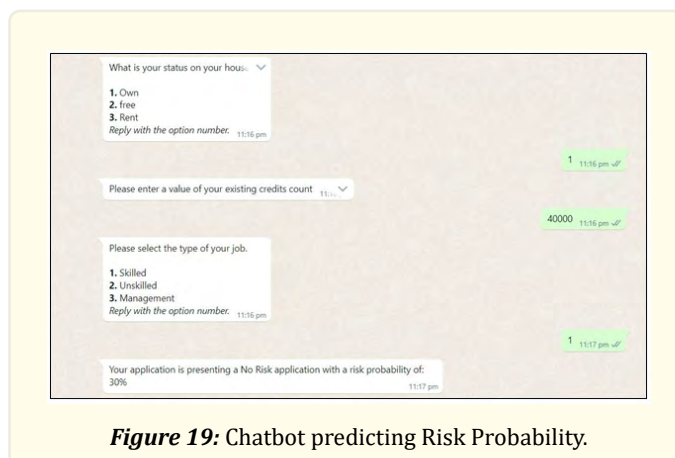


Figure 19: Chatbot predicting Risk Probability.

Conclusion

With the help of a wide range of IBM Watson services [4, 6], we have built a chatbot for loan application process assistance [3]. The chatbot can smartly deduce the risk probability that could be involved during the process. This has been done with the help of XGB classification model [13] in Machine Learning. The functionality to use the chatbot through WhatsApp was added using Twilio.

Acknowledgements

I would like to thank my research supervisor Dr. Madhulika Bhaduria. Without her assistance and dedicated involvement in every step throughout the process, this article would have never been accomplished. I would like to thank you very much for your support and understanding.

References

1. Adamopoulou Eleni and Lefteris Moussiades. "An overview of chatbot technology". IFIP International Conference on Artificial Intelligence Applications and Innovations. Springer, Cham (2020).
2. Cahn Jack. "CHATBOT: Architecture, design, & development". University of Pennsylvania School of Engineering and Applied Science Department of Computer and Information Science (2017).
3. Godse Neha Atul., et al. "Implementation of chatbot for ITSM application Using IBM watson". 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). IEEE (2018).
4. Ralston Kennedy., et al. "A voice interactive multilingual student support system using IBM Watson". 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE (2019).
5. Sabharwal Navin., et al. "Developing Cognitive Bots Using the IBM Watson Engine: Practical, Hands-on Guide to Developing Complex Cognitive Bots Using the IBM Watson Platform". Apress (2019).
6. Sabharwal Navin., et al. "Building your first bot using Watson assistant". Developing cognitive bots using the IBM Watson engine. Apress, Berkeley, CA (2020): 47-102.
7. Biswas Manisha. "IBM Watson Chatbots". Beginning AI Bot Frameworks. Apress, Berkeley, CA (2018): 101-137.
8. El Naqa Issam and Martin J Murphy. "What is machine learning?". machine learning in radiation oncology. Springer, Cham (2015): 3-11.
9. Hoyt Robert E., et al. "IBM Watson analytics: automating visualization, descriptive, and predictive statistics". JMIR public health and surveillance 2.2 (2016): e5810.
10. Miller James D. Hands-On Machine Learning with IBM Watson: Leverage IBM Watson to implement machine learning techniques and algorithms using Python. Packt Publishing Ltd (2019).

11. Bini Stefano A. "Artificial intelligence, machine learning, deep learning, and cognitive computing: what do these terms mean and how will they impact health care?". The Journal of arthroplasty 33.8 (2018): 2358-2361.
12. Stringer Roger. Twilio Cookbook. Packt Publishing Ltd (2013).
13. Chen Tianqi., et al. "Xgboost: extreme gradient boosting". R package version 0.4-2 1.4 (2015): 1-4.

Volume 2 Issue 6 June 2022

© All rights are reserved by Manasi Chhibber., et al.