**Automation Based Chatbot Model for Cricket**

**Abstract**

Day by Day Al (usage of latest technology) is being used in every field to simplify their work. "AI in Sports" is one of the implementations of A.I. is one of the hot topics in the market. So, we are trying to build an automation platform where the interested people can learn cricket without any investments. This project includes an automatic chatbot which can provide required rules and regulations of the game as well as it can build powerful training resumes for the user based on their fitness level, it can do side by side comparison with the actual plays of professional level players

**Key Words**: Sports Coach, Intelligent Game Tutor, Cricket Coach, Artificial Intelligence, Neural Networks

**OBJECTIVES**

1. To make an automated chatbot which can used to substitute a cricket coach.

2. The chatbot can give information about the rules of game, the styles of plays, how to play in different kind of situations,

3. It plays certain videos of some famous players who performs a certain plays or styles with detailed statistics in different angles to make the users learn visually as well.

4. The user may upload a video of themselves performing a certain play to the application and it will display both the videos side by side of a professional performing it and the user's video.

5. Install user feedback option with every program.

6. To build a program which can suggest a training resume the from inputs given by the user based on his/her physiology.

**Overview**

Our project is on “Automation based Chatbot model for cricket”, the chatbot is created using deep learning and python. Also, we will have an overview of machine learning with respect to deep learning and chatbot that we will be using to build our chatbot model. Also, we will be going through the chatbot intent file and what all information is needed to feed our chatbot model. The main purpose of using the python language is the vast libraries which are there, and which makes the task very easy to work upon. The libraries and the tools available make it possible to build the chatbot very quickly and easily instead of playing with the whole sort of algorithms manually which makes it complicated and tough to handle. With Machine Learning we are making a model such that it learns by itself after training and iterations. We feed our model with some sample data or more precisely we can say that training data which is collected in real-time or .csv files from various sources such that after learning it can work on test data and can give accurate results. The Machine Learning model is trained for several epochs such that the accuracy of the model can be increased, and the output is very accurate, and the loss is less.

**INTRODUCTION**

Cricket is very familiar sport which doesn’t require any introduction. Cricket fever is very high in India, day by day many people were choosing cricket as their profession. This fever intents is increasing every day. Cricket was brought to India by the British around early 1700s. In the early 1900s, some Indians joined the English Cricket Team. Cricket is regarded as a religion rather than a sport in India. The Indian Cricket team debuted in 1932 and has since risen dramatically and continues to dominate the Indian sports scene. Cricket is in the blood of every Indian, regardless of caste, creed, state, or religion. Though cricket is not a national sport, it has surpassed Hockey in terms of viewership and fan base. Famous cricketers such as Kapil Dev, Sunil Gavaskar, MS Dhoni, and Sachin Tendulkar are worshipped as Gods by the Indian people. Indians consider cricket to be an obsession. The Indian National Cricket Team, also known as "Team India," is known as the "Men in Blue" and is well-liked throughout the subcontinent. The Board of Control for Cricket in India, or BCCI for short, governs the Indian cricket team. The International Cricket Council currently ranks India first in One Day Internationals and third in Twenty20s.

Sport is a channel that is valued by both the wealthy and the poor in a society. As a result, it is prevalent for a nation to pursue glory in sports because all of its members can easily share in the resulting joy. Such victories reassure citizens about their place in the larger world. Cricket played this role in India, that also proceeded to develop as a sovereign nation in the post - revolutionary period. Cricket has the potential to bring about the kind of success that makes a prevalent Indian, who is constantly subject to enormous socioeconomic uncertainly, feel very proud and equitable to certain other peoples. Cricket may now be a part of living, more potent than India's traditional cultures, but this was not always the case. Apart from participants in other sports, cricketers, as described in this essay, could keep up with the shifting spirit of the times of the Indian Behemoth by scoring international victories at critical moments. Cricket could serve as a natural remedy even during liberation struggle, a booster during the country's early years, and a balm during the turmoil 1970s and 1980s. Then, beginning in 1987, cricket became the Indians' sole means of dictating terms to the rest of the world.

Chatbots are the future of customer service. Therefore, they are being used by businesses as a way to interact with their customers or potential customers. A chatbot can be used for various purposes and it can be programmed in several ways. While the traditional chatbots are programmed to respond to keyword matching, an automation-based chat-bot model for cricket can engage users with a more personalised response that is tailored for each user because it depends on how they answer specific questions posed by the bot. Automation based chat-bot model is a revolutionary idea that can change the way we learn cricket. With automation-based chat-bots, the user can simply type in “teach me cricket” and will be able to know What are the famous styles of the famous cricketers, Types of matches, types of bowling. This bot also helps users with other information like creating their self-training resume. This bot automatically updates live score in real time when matches are underway. It provides latest score of both teams in an easy format that can be understood by everyone. Chat-bot technology helps users find out the upcoming matches as well this gives them enough time to plan their day schedule accordingly or even plan vacations if they are on a break from work or school at that The chat-bot system is also a practical solution to handle the changing and growing user base. Chat-bots are more flexible in an ever-changing environment, such as the one that cricket faces. They can adapt to the change in rules and regulations quickly without any impact on the quality of service they provide. One of the main reasons for cricket's popularity is its unpredictability. This means that it is difficult to predict when and which sport will be played next. While this may not be suitable for a traditional cricket management system, it does work well with a chat-bot model Chatbots are designed for repetitive tasks and can handle over 50% of them with minimal human intervention The chat-bot is an automation-based coach that stat working on machine learning and dialog framework to impart knowledge about the game of cricket. Chat-bot is a computer program that conducts conversation using artificial intelligence (AI) with one or more human participants on a text-based interface such as through instant messaging, chat rooms, email, or other forms of text input. The use of technology in coaching cricket is one of the most talked-about topics that has been discussed for quite a long time now. Coaching chat-bot model for cricket is an interesting topic to explore because it has an extreme potential to revolutionize the way people learn about cricket coaching, learning programs, education, and development. The coach chat-bot model is the first of its kind in cricket to automate a coaching session. It is designed to aid illiterate people.

There are three major benefits of this model:

1. Anyone can access to learn cricket
2. Bridges communication gap between remote coaches and players.
3. Helps players with personalized training sessions

**Literature Review**

Almost every major professional sport is being influenced by artificial intelligence. This is a convenient interruption of the business as media inclusion becomes increasingly significant as the primary source of income in elite athletics. It is clear from the progression of this pattern that fans want more access to their favourite game group, and innovation is a critical channel for meeting this need. Simulated intelligence is allowing fans to feel closer to the players and the game than at any other time in recent memory. After a while, increasingly personalised encounters and increasingly supportive computerised communications may lead to greater fan steadfastness and commitment. In this regard, automated reasoning in sports will be like its applications in media and programming for large part. Wearable technology is another application of artificial intelligence in sports that holds tremendous promise for future development.

Organizations recognise the need to go beyond simply providing the following information to transform it into critical bits of knowledge that truly assist competitors in meeting their presentation objectives. These items can appeal to both wellness enthusiasts and expert competitors, providing a broad market reach.

The A.I. consciousness is most likely influencing the sports industry. With advancement spreading across all zones of active space, it is attempting for each sector to reload and doing everything cutting-edge items to address issues in the territories of complex scenarios. In just about any instance, there is a lack of improvements pending in every area of the company, including the games showpiece. Notwithstanding, a most recent innovation, such as A.I., can aid in market dominance. Numerous A.I. advancement organisations are offering A.I. frameworks that enable the games industry to assist players and supporters [4]. The games industry has seen the significant effect of human-made consciousness through various one-of-a-kind ideas aimed at improving implementation and dedication. Humanmade consciousness plays an important role in advertising sports because its capacities continue to be fixed and raised. Pairing Information systems is a standout player among many other A.I. Advancement Companies in the world, with 18+ years of involvement in providing the most significant advancements with innovative techniques. We have decided to commit and eligible designers ready to implement A.I. frameworks in various spaces, including the gaming industry.

**METHODOLOGY**

Our project is on “**Automation based Chatbot model for cricket**”, the chatbot is created using deep learning and python. Also, we will have an overview of machine learning with respect to deep learning and chatbot that we will be using to build our chatbot model. Also, we will be going through the chatbot intent file and what all information is needed to feed our chatbot model. The main purpose of using the python language is the vast libraries which are there and which makes the task very easy to work upon. The libraries and the tools available make it possible to build the chatbot very quickly and easily instead of playing with the whole sort of algorithms manually which makes it complicated and tough to handle.

**Machine Learning:**

With **Machine Learning** we are making a model such that it learns by itself after training and iterations. We feed our model with some sample data or more precisely we can say that training data which is collected in real-time or .csv files from various sources such that after learning it can work on test data and can give accurate results. The Machine Learning model is trained for several epochs such that the accuracy of the model can be increased, and the output is very accurate and the loss is less.



**Deep Learning**

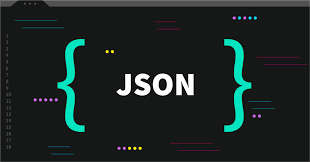
Also, **Deep Learning** which is the type of machine learning which incorporates the use of neural network which mimics the behaviour of a human brain. With only single layer the network can make approximate predictions, additional layers (i.e., hidden layers) can be used to optimize our model and to refine the accuracy of the model.

**Creating an Intents File**

Next task is to create an intents file (i.e., our training data). This file is used to tell our chatbot what has to be done. This training data is feed to the model to learn so that it knows that if okay I get this, I must do that. So, this intents file is sort of starting point through which the chatbot responds to a certain instruction or a certain text. So, we have built our intents file like a JSON file than we will put everything into an adjacent object and then inside that suggestion object we have a key that is called intents and inside that key we have a list of intents so that intends will be a list and it can be as long as you want which depends on the functionality that we want our chatbot to have.

The questions asked or the queries which our chatbot gets are to be group with the intent we have provided already such that to try figure out what the intention is, what are the user trying to speak, what the user want to know or achieve. The more are the details for the questions, the intents file going to be more useful and more accurate the chatbot going to be.

So, to begin with the intents file, we will have a couple of keys and variables and the first one is tag. For every intent there will be a tag and text are going to be like the high-level description of that intent. So, the very first intent we can have been for greeting. How our chatbot will greet if somebody is contacting our chatbot for the very first time. Like what normal people do, in the same way chatbot will say “Hii… How are you” or something like “hello”, “hey”, “hey, wassup” or anything which they feel to greet with, every different variation in which user can use to say hello. All these words our grouped under the tag of greetings. So, whenever our model encounters with any of this word, to respond to the queries a nice number of responses should be there so that our chatbot don’t look monotone and every time just repeating the same words “hello, how are you”. To make our chatbot more user friendly more and more responses should be given as possible. The response can vary depending on the culture where we are based, we must think of everything which a user can expect, or a user can ask.

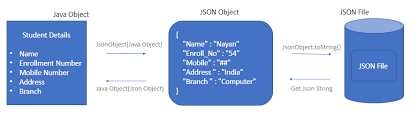


The intents file is the one where we need to focus more and more as it represents how our chatbot model will be responding to the queries. If we the junk data than it will be junk, if we put useless data all of it will be a crap. So, we need to be careful while creating the Intents file. This phase of the project is very critical and the most time consuming where we need to put efforts thinking in all different directions in which a user says something and, we need to tag it properly.

Putting the responses in the correct tag is very important. Putting the wrongs responses in the wrong tags might confuse the chatbot model. Like instead of saying “goodbye”, “thankyou”, the chatbot replies with "welcome”, “hello”. We need to spend majority of the time building the correct intent file with no errors.

Dividing and deciding what responses goes to which tag is a very tedious task to do and what are the patterns in terms what people would possibly say and what response should I give to the question. In fact, the patterns are the most important even more than the answers to the queries. Also, instead of only one response we can have multiple responses for the better quality of answers. All these responses go into a python list of strings. This is a list of strings and tag is just only one string that represents the tag. Since this a tag it contains only one word, if we want to add more words, we can attach it using underscore.

So, the chatbot which we are building is based on the cricket, the people who want to learn or have queries can directly reach and have their queries which our chatbot will be answering. The users who have no access to the academies can refer to the app and is much more convenient. Here, they can get information as well as practical knowledge for the training.



The chatbot can be easily expanded by expanding the intents file we can do much more. The flexibility of the model is dependent on the intents, tags which we are providing. Therefore, the chatbot model is very much scalable to our needs.

So, this is how we are building our intents. JSON file which is in python. Therefore, it is list of objects and inside which every single object is an intent such as tag, followed by pattern, followed by the responses. There can be any number of the tags and, the pattern and responses associated to that tag.

**‘Tag’** is just a string.

**‘Pattern’** is a list of strings.

**‘Responses’** are a list of responses.

These three things together are called as **intent** and there can be many intents as many as possible.

Next, we will go through the code to build our model and this is the main part.

So, before we get started there are couple of things which we need to install such that to build our model

**LIBRARIES USED**

1. NLTK library
2. NumPy library
3. Keras library
4. TensorFlow library
5. Flask
6. Random
7. JSON
8. Pickle

**NLTK: -** We need nltk library to sort of process the words which we get in order to put it through our coding. We want to use the natural language processor to tokenize each, and every pattern and the pattern are these ones which we have in our intents file. We need to lemmatize the word to break down to its specific meaning because what we need is to simplify the language as much as possible for the machine to understand. For example, if someone says I’m crying while the other says I cried, and another says that someone else cried. The core meaning is crying but context is different in different situations, so we need to lemmatize the words to get the exact meaning for the data.

* Wordnet – It is the large world database which is part of the nltk library which contains all the words like nouns, adjectives, adverbs and verbs. To use the wordnet, we have to first install the nltk library and then download the package.
* Punkt - By developing a model for words that begin a sentence, collocations, and abbreviations using an unsupervised method, this tokenizer extracts a list of sentences from a text. It must be trained on a substantial volume of plaintext in the target language before being used.
* WordNetLemmatizer – In the event that the input word cannot be discovered in WordNet, WordNetLemmatizer returns the term in its original form.

**NumPy: -** This library is used for working with arrays. NumPy array are much faster than the lists. There are many functions associated which make working with the nd array very easy.

Keras: - This library provides a python interface for ANN (artificial neural networks). Also, this library acts as an interface for the tensorflow library too. With Keras library we can:

* Iterate at the speed of thought
* Exascale machine learning
* Deploy it anywhere
* A vast ecosystem
* State-of-the-art research

The methods we have used within Keras library are:

* Sequential – A sequential approach is adequate for a straightforward stack of layers with precisely one input tensor and one output tensor for each layer. A sequential model is improper when the model seems to have a large number of inputs or outputs. Numerous inputs and outputs are included in every one of your layers.
* Dense
* Activation
* Dropout
* **TensorFlow:** A free and open-source library for artificial intelligence and machine learning is called TensorFlow. It may be used to a variety of applications, with a focus on deep neural network training and inference.
* **Flask: -** It is a micro-web framework in python. It is called as micro framework because it does not require any particular tools or libraries. It is used for developing web applications using python.

1. WSGI
2. WERKZEUG
3. JINJA2
4. Random
5. JSON
6. PICKLE

**AWS-EC2 Instance (Ubuntu 20.02): -**

**Step:1 Launch an instance:** Following the instructions in the following procedure, you can start a Linux Instance using the AWS Management Console. This report doesn’t address every possibility because its goal is to help you swiftly start your initial instance.

**To launch the instance:**

* Open the amazonec2 console

<https://console.aws.amazon.com/ec2/>

* For name, under name and tags, provide an identity that accurately describes this instance.

Background pattern

Description automatically generated with low confidence

* Complete the items under **Application and OS** images:
* In the **Quick Start** section, and then choose UBUNTU. This is the operating system for your instance.

Graphical user interface, application

Description automatically generated

* You can choose the processor setup of the instance below Instance type from the Instance type list. Select the t2. micro ec2 instance, as this is the default option. The free tier is available for the t2. micro instance type. You can utilise a t3. micro instance on the free tier in regions where t2. micro is not accessible.

Graphical user interface, text, application, email

Description automatically generated

* Select the key pair you made when setting up under Key pair (login) for Key pair name.

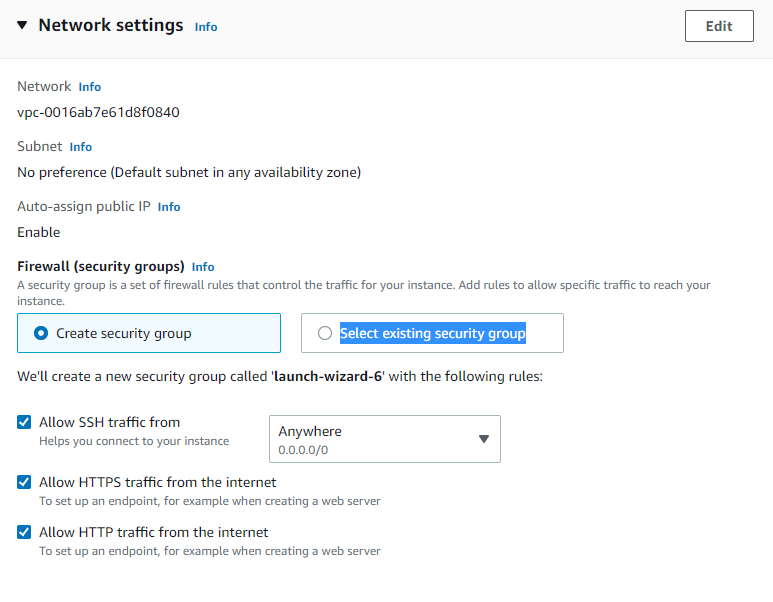
Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* Next click on Edit near to Network options. You can observe a new wizard formed and chose a security group for you under Security group name. Otherwise, you can choose the security group you created while setting up using the procedures below and utilise that instead of this security group:
* Pick **Select existing security group** from the menu.
* Select your security group from the list of available security groups.



* Change the default selections for the other configuration settings for your instance.
* Next choose required storage for your instance.

Graphical user interface, text, application, email

Description automatically generated

If you’re finished, select Launch instance after examination the summary of your instance config in the Summary box.

Graphical user interface, text, application, email

Description automatically generated

* Your instance launches and an affirmation screen appears. To close the confirmation screen and revert to console, select Display all instances.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* The launch’s status is visible on the Instances screen. Launching an instance just takes a brief while an instance’s first phase after launch is pending. The instance obtains a public DNS name when it begins, and its status is changed to operating.

A picture containing table

Description automatically generated

* The instance may not be ready for you to join for a few minutes. You may inspect this information in the **Status check** to make sure your instance successfully completed all progress inspections.

Graphical user interface, text, application

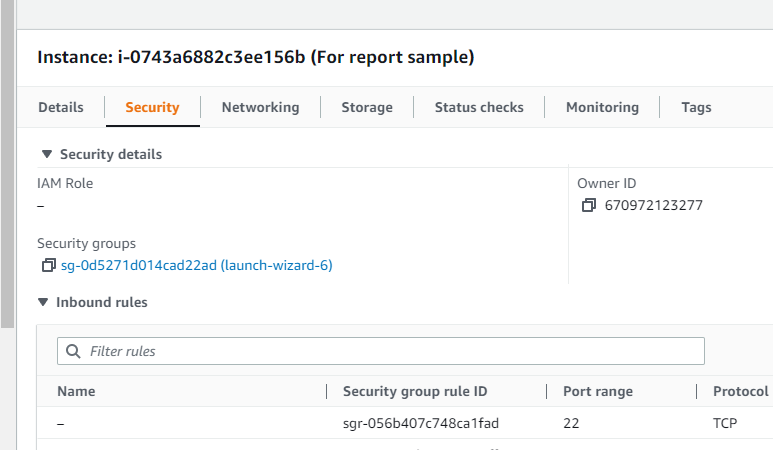
Description automatically generated

**Step:2 Connect to your Instance:**

Graphical user interface, text, application, email

Description automatically generated

You can use your own custom port by opening the security option under instances



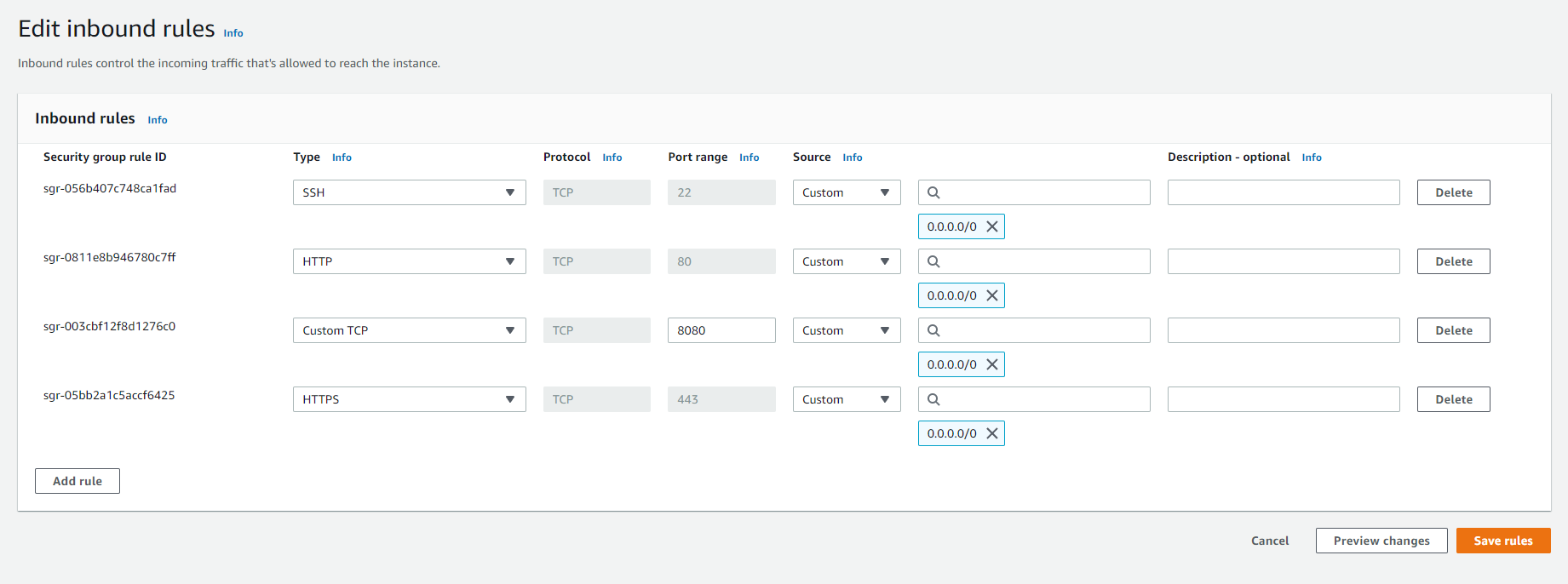
After that click on security group listed below, it opens a new box.

In that box click on edit inbound rules

Graphical user interface, text, application

Description automatically generated

It shows like this, Now select custom TCP and add wrking port range



**Code/Program**

Following, let’s move on to the coding part:

So, firstly the most important part is to import all the libraries needed to code our model.

**NLTK library**

We imported NLTK library which we used for classifying the intents and used natural language tokenizer to tokenize each pattern

**WordNetLemmatizer**

Then we used WordNetLemmatizer to break down the words to the basic meaning such that our machine can understand it. Because in case there can be many words whose basic meaning can be same but for different contexts the meaning changes.

Also, we have used the Bag of Words to keep track of occurrence in the text\document.

**Training Data**

After completing this task, next is to go with the training data. Training data is what we’re going to fit into our model. So, for fitting the data we have used a very model i.e., Sequential Model which is imported from the Keras library. There are many other models too nut using the Sequential model is one of the basic models that is easy and simple to get through.

**Fitting the Data**

After that we are going to fit our model with training data and tell our model what that information means. So, the intents which we have created manually helps us here to train our model by telling the model if you are getting this then you can reply with the following response.

So, while the user chats with the model but the word used might not be the same than it will look for the keywords and answer the query.

We also integrated the cricket game rules and regulations with chatbot so it can guide well.

This is an essence for what a chatbot really does…

So, this is the code for the model. After building the model save the file as chatbot.py.

**Processing the Data**

For processing the data also, we have file called processor.py where we put in the questions and all the processing and implementing is done so that to get an answer.

Basically, first step is to clean up the sentence like if the user is asking a question “Hii. How are you”. Now, we will lemmatize the sentence and break down the sentence into words so that our model can understand the meaning of the words.

Then we will be having a predicting class where the prediction of the intent is done after lemmatizing the sentence. We check the words in the intent file to match the word within it so that chatbot can figure out what the user is asking. The model looks for the word in the intents where the word fits into what can be the pattern and then what will be the best response to the query.

This is all about the processing part where we process the queries from the user and then match the tokenized words with the intents already have to give back the response.

**Front-end (Web Development)**

Next, we have done is the front-end using Flask.

Here we have only two files: first one is **app.py** where we define the flask application and then other one, we have is **index.html** where we have the templates. This is the minimal code which we can have to create an application, therefore, flask is called as the lightweight web application framework because it only requires very few lines of the code.

After we have done deployment, our chatbot is ready to answer all the queries of the user whatever he/she may ask.

**Procedure**

Procedure for running our model is

1. Open command prompt in pem location folder and connect it by passing the ssh (secure shell) client
2. Install all the required libraries

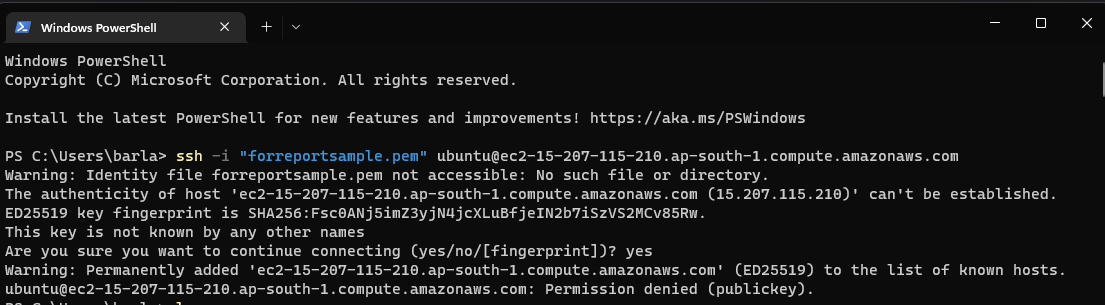
Text

Description automatically generated

1. Run the chatbot.py file to create the model using command **python chatbot.py**

* It will create files: chatbot\_model.h5, words.pkl and class.pkl
* H5 file to create large amounts of data
* Pickle file to serialize and deserialize the data

1. Run the App to create a Flask front end on any port the app is pointing python3 app.py.

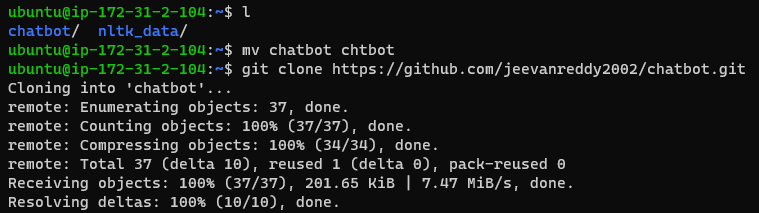


Run ls(list) command to know where the location is and use cd (current directory) to open any file.

1. Now to check every module use pip list
2. Next run training model for the chatbot with python3 in front to the file (example- Python3 chatbot.py)

Text

Description automatically generated



Text

Description automatically generated

1. Now run the main file with python3 and u will be able to see that your application will be running

Text

Description automatically generated

1. After running the application, the applications copy the public IP Address in ec2 dashboard for the instances and paste it in a new tab with your own custom port which you have created earlier.

Graphical user interface, text, application, email

Description automatically generated

Now the website is working, …

Text

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

**Step:3 Stop the Instance:** After using instance you built for this project, you should terminate it to complete cleaning up. If you wish to use this instance in a different way before cleaning it up.

Graphical user interface, text, application

Description automatically generated

1. To terminate your instance
2. Instances can be selected from the navigation pane. Choose the instances from the list.
3. Choose Instance state, Terminate instance.

Graphical user interface, application

Description automatically generated

1. When requested for confirmation, choose Terminate.
2. Your instance is terminated by Amazon EC2. Following the termination of your instance, the entry is immediately erased from the console after a brief period. The terminated instance cannot be manually removed from the console display.

**Conclusion**

AI chatbot innovations and business products are developing rapidly, as are the number of research studies on these technologies. Finally, because increased adaptation of chatbots for diverse populations is expected, future research must consider equity and fairness when creating and planning chatbot initiatives. Therefore, we have successfully developed a Chatbot which can assist users who wish to learn cricket.

In our opinion, our software serves as the easiest way for all rural people to know basics about cricket, if they are very serious about selecting cricket as their career option, from basics like rules and regulations about cricket, different styles and plays in batting as well as bowling, our website will provide them with complete assistance.