iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VII **Month of publication:** July 2021

DOI: https://doi.org/10.22214/ijraset.2021.36718

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

Chatbot for Career Counselling

Mr. Mandeep Katre[1], Samya Manchanda[2], Rashi Agarwal[3]
[1]*Professor,* [2,3]*Student, Dept. of Computer Science and Engineering, Inderprastha Engineering College, Ghaziabad, India*

Abstract: ***A chatbot is a computer application that is developed to prevaricate and process conversation between humans and digital devices offering the users an experience as if they are communicating with a person in real life instead of a digital device. With today's customers expecting immediacy and actualization in their interactions with devices, the addition of chatbots as a communication channel has become critical to growing demand. This paper presents the design of an expert system for educational guidance for students available after SSC & HSC. This project is aimed to implement a web-based chatbot using Rasa NLU and machine learning algorithms that analyze user's queries and understand the user's messages. The bot is designed to address the queries of the user related to the courses they should opt for and colleges that offer the course. The user just has to query through the bot and based on the inputs taken from the user the chatbot will accordingly predict a suitable course for the user. The system responds by showcasing the predicted course for the user. The user can access the bot without any pre-registration.***

Keywords: Chatbots, Rasa NLU, Machine Learning, Career guidance.

I. INTRODUCTION

Nowadays, it is a major thing to have basic knowledge about various career fields that are high in demand and available across the world. Career affects all aspects of our life - Personal, Social and Professional. Selecting a wrong career will put you in an inappropriate situation which will not only affect your professional life but will also have an unfavourable impact on your personal life too. Most people often try to secure a position anywhere they get as early as possible after their graduation. There are only a few who take time before deciding a correct career option. If you jump into any profession, then in the initial years you could feel good about having a job but later you will find it stressful. That is when you will realize it is not the right profession for you.

It is of no surprise that the students that are currently enrolled in various colleges and universities claim that they would have selected a different course to pursue if they would have been aware about it earlier. Career counselling[6,7] bots are highly useful for this. Chatbots[10] are preferably taken into use numerous times in the past few years as it offers effortless communication between user and system. Also, there is a soaring reach for chatbots becoming more popular in the coming years.

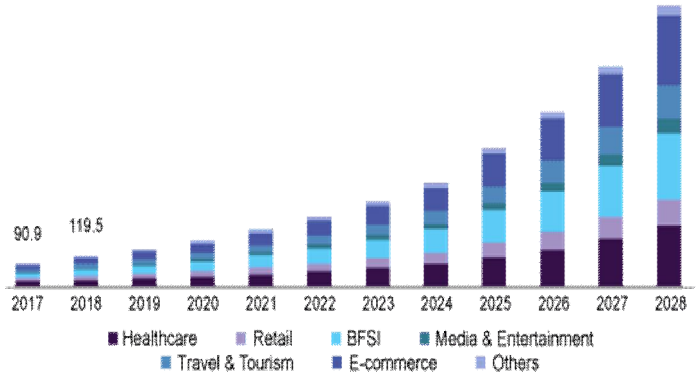


Fig 1: Growth in use of Chatbot over years

Our chatbot is designed using Rasa NLU and machine learning algorithms.

Our system offers a one-step task that provides career guidance[4,5] and allows the users to get all answers about the courses one has available as an option based on their interest and thus, picking you up out of the major struggle. The system delivers the most suitable predicted courses to the user on the basis of their ratings provided to the bot. Our well-designed infographics and thorough researched data on ample courses will aid the user to get complete information about the courses and the colleges that offer those courses. It is just like having your own exclusive admission counsellor with you whensoever.

©IJRASET: All Rights are Reserved

II. LITERATURE REVIEW

A. Existing System

A chatbot [3] is aimed to find new, easy & innovative ways to help the user by ditching the old age forms. There already exists many websites and bots that assist the user regarding their career related information. Students these days do effective online searching before opting for the fair career option. But many of the websites and bots ask the user to buy some kind of subscription or pay a certain amount so as to get assistance any further failing to which many users could not get the necessary assistance of making the right career move.

Archana S. Parab,Siddhesh Palkar and Satish Maurya [1] basically refers to the use of Natural Language Processing and have also developed a knowledge base which comprises unstructured and structured information which assists in answering complex queries of the user, also helping them in developing an intelligent chatbot.

Faraaz Ansari, Shaikh Saad, and Shaikh Shareem [2] refers to Fuzzy inference system along with some artificial intelligence approaches which helps them in making AI based expert systems. The approach is to extract some tokens from the sentence in order to find the goal of the sentence by matching the input user.

Nangineni Vijaya Lakshmi, Mandadapu Hema, Kondepati Gowthami & Mrs. Jaladi Praveena [8] have also used NLP (Natural Language Processing) and some of the artificial intelligence techniques that have helped them in developing an intelligent career counselling chatbot.

Y. Wu, G. Wang, W. Li, and Z. Li, [11] refers to some parallel computing and artificial intelligence techniques in developing the automatic chatbot.

Vijayakumar R, Bhuvaneshwari B, Adith S, Deepika M [9] have developed an AI based academic chatbot using Machine learning techniques, NLP and also using the python programming language to develop the user interface and MySqL database for storing relevant information which helps them in answering the queries of the user.

B. Rasa NLU

Rasa is an open-source machine learning framework for building AI assistants and chatbots. Rasa NLU is a library for NLU which takes the user input and tries to infer the intent and elicit the available entities and helps the bot to make sense of what the user is saying. Rasa Open-Source framework is the most flexible and transparent solution for conversational chatbot offering complete control over building a chatbot for assistance of the user.

C. Chatbot Concept

1) *Intent:* It can be defined as the objective or target of the user input. More precisely it is the action a user aspires to take that they anticipate the chatbot to facilitate.

"The VERBS in your dialog"

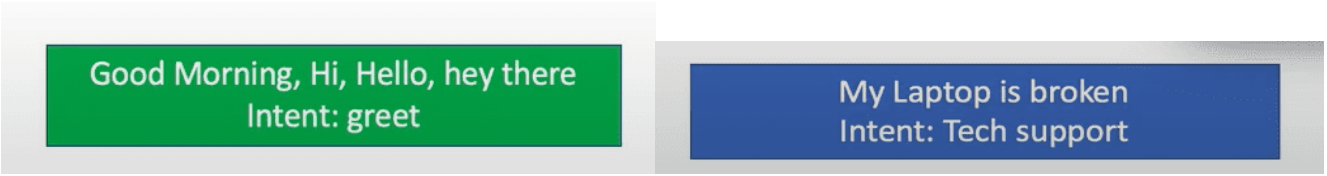


Fig 2: Intent Classification

2) *Entity:* It can be defined as the useful information given by the user that can be extricated.

"The NOUNS in your dialog"

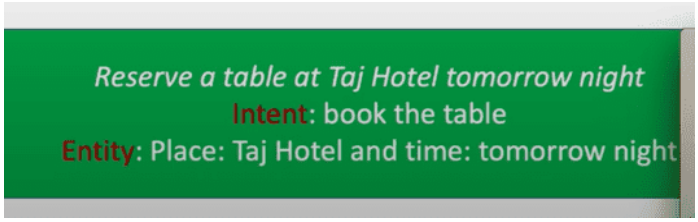


Fig 3: Entity Classification

©IJRASET: All Rights are Reserved

3) *Actions:* It is an operation that a bot can perform. Actions could either be a just a reply to something in return or querying a database. Actions also include any other operation possible by the code.
4) *Stories:* Stories are defined as the sample interactions between the bot and the user, defined in particular of the intents captured and actions performed. These can be used to train dialogue management models.
5) *Domain:* Domain knowledge is used to define configuration for conversation sessions between the bot and the user. Domain describes the entities, intents, forms, actions etc. that the bot should have knowledge about.

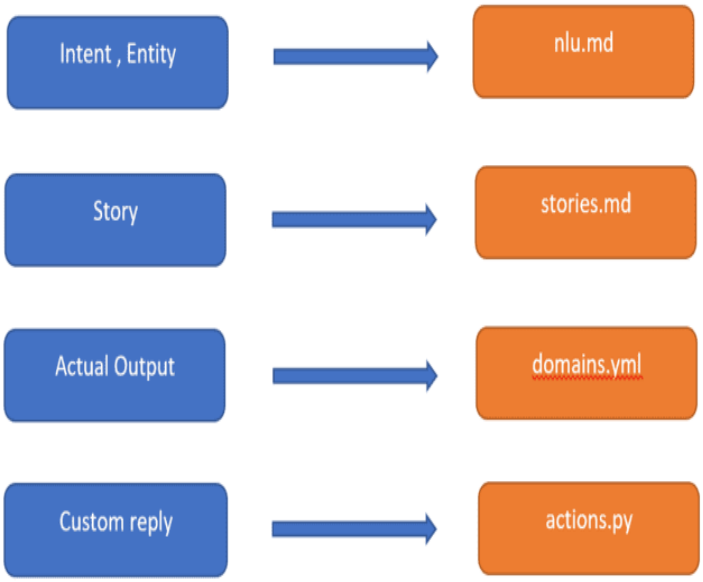


Fig 4: Defined Chatbot Concept in Code

III. PROPOSED SYSTEM

The proposed system is selected for helping many students that are facing confusion about selecting their career related problems. This system is having knowledge about various courses and the colleges offering those courses. The user is supposed to rate different subjects on the basis of his/her interest in it. Based on the ratings of the different subjects, our chatbot uses a machine learning model to suggest a suitable course to the user. This system is voice enabled and responses to the user in both text and speech in order to make the conversation attractive and interactive.

Advantages

1) This helps to resolve career related queries.
2) Users will be suggested courses on the basis of their interests rather than subjective marks only.
3) The designed and processed algorithms result in predicting the right course.

Comparisons between various systems gave a clarity and better understanding of our project. Advancement in technology has brought in many such systems which are already in place today but, all of these systems have their own set of limitations. Our chatbot aims to provide a comprehensive solution to the existing limitations by yielding better accuracy in terms of performance and computational time and cover a wide area of courses as compared to the preexisting systems and developing a scalable, robust and user-friendly guide.

The algorithms used in our proposed system for predicting the right set of courses are as follows:

A. Support Vector Machine

Support Vector Machine or SVM is the most commonly used machine learning algorithm for labelled data i.e.
Supervised Learning. This classification is widely used to handle multi-classification problems. Note that we have created our own dataset by collecting samples using forms, surveys and rigorous interactions. Following steps have been followed to implement SVM on Arts_dataset.

©IJRASET: All Rights are Reserved

1) *Step 1* - Fit svm classifier using sklearn library
2) *Step 2* - Making a non-linear algorithm for a model. Default kernel RBF has been selected. RBF stands for Gaussian Radial Basis function. It has been used as we have very little knowledge about our dataset. Mathematical formula for RBF is -

$$k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$$

3) *Step 3* - Train the non-linear model.
4) *Step 4* - Predict from the trained model
5) *Step 5* - Model evaluation via conclusion matrix and accuracy.

SVM (Support vector machine) classifier

```
In [28]: from sklearn import svm

In [29]: clf = svm.SVC()
         clf.fit(X_train, y_train)

Out[29]: SVC()

In [30]: svm_y_pred = clf.predict(X_test)

In [31]: svm_cm = confusion_matrix(y_test,svm_y_pred)
         svm_accuracy = accuracy_score(y_test,svm_y_pred)

In [32]: print("confusion matrics=\n",svm_cm)
         print("  ")
         print("accuracy=",svm_accuracy*100)

confusion matrics=
 [[ 8  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  8  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0]
 [ 0  0  7  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  1  5  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  8  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  1  0  0  4  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  5  0  0  0  0  0  0  0  0  1  0]
 [ 0  0  0  0  0  0  0  9  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 10  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  1  0  0  1  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  5  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 10  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  7  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  7  0  0]
 [ 0  0  0  0  0  0  1  0  0  0  0  0  0  2  0 12  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  5]]

accuracy= 92.62295081967213
```

Fig 5: Arts_dataset accuracy using SVM

©IJRASET: All Rights are Reserved

B. Decision Tree

Decision tree is a supervised learning algorithm to deal with large datasets in order to predict an output. It uses multiple features to split and form a tree. Following steps have been followed to implement the Decision tree on Arts_dataset -

1) *Step 1* - Load required modules and libraries
2) *Step 2* - Split the dataset into training-testing data. Testing data is 25% of the total dataset.
3) *Step 3* - Import DecisionTreeClassifier from sklearn to make a model . Create an instance of the class to fit the training data to the model.
4) *Step 4* - Predict from the tree.
5) *Step 5* - Model evaluation via conclusion matrix and accuracy.

Decision Tree Classifier

```
In [13]: from sklearn import tree
         from sklearn.model_selection import train_test_split
         from sklearn import preprocessing
         from sklearn.metrics import accuracy_score

In [14]: # Split the data into training and testing sets
         X_train, X_test, y_train, y_test = train_test_split(X_normalized, y, test_size=0.25, random_state=0)

In [15]: clf = tree.DecisionTreeClassifier()
         clf = clf.fit(X_train, y_train)

In [16]: from sklearn.metrics import confusion_matrix,accuracy_score

In [17]: y_pred = clf.predict(X_test)

In [18]: y_pred

Out[18]: array([ 4, 12,  6,  0, 11, 15, 15,  0, 14,  4,  0, 15,  8, 15, 14,  6, 11,
                 5,  1,  1, 12,  1, 15, 10,  6,  3, 12,  1,  0, 15, 13,  4, 12,  5,
                11,  7,  8, 15,  1,  7, 15,  4,  8, 11,  3,  0,  4,  0, 12, 12,  2,
                 1,  2,  8,  7,  7, 16, 15,  7,  9,  5, 15, 14, 14, 14,  1, 16, 12,
                 8, 15,  6, 12,  4,  2,  3,  6,  7, 16, 14, 13,  4,  7, 11, 12,  4,
                 3, 15,  8, 14,  4,  8,  4, 13, 15, 12,  8,  9,  5, 13, 13, 16,  0,
                 7,  3, 16, 16,  3, 10, 14,  3,  3, 10,  3, 16,  1, 15, 15, 13,  6,
                 7,  1,  0])

In [19]: cm = confusion_matrix(y_test,y_pred)
         accuracy = accuracy_score(y_test,y_pred)

In [20]: print("confusion matrics=\n",cm)
         print("  ")
         print("accuracy=",accuracy*100)

confusion matrics=
 [[ 8  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  9  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  3  4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  2  2  0  0  0  0  0  0  0  0  0  1  0  1]
 [ 0  0  0  0  8  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  1  0  4  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  5  0  0  0  0  0  0  0  0  1  0]
 [ 0  0  0  0  0  0  0  9  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  2  0  0  0  0  8  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  5  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 10  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  6  0  0  1]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  7  0  0]
 [ 0  0  0  0  0  0  1  0  0  0  0  0  0  0  0 14  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  5]]

accuracy= 88.52459016393442
```

Fig 6: Arts_dataset accuracy using decision tree

©IJRASET: All Rights are Reserved

C. Xg Boost

XG Boost is an open-source gradient boosting framework.

The following steps have been followed to implement

XgBoost on Arts_dataset -

1) *Step 1* - Load required modules and libraries
2) *Step 2* - Split the dataset into training-testing data. Testing data is 25% of the total dataset.
3) *Step 3* - Import DecisionTreeClassifier from sklearn to make a model . Create an instance of the class to fit the training data to the model.
4) *Step 4* - Predict from the tree.
5) *Step 5* - Model evaluation via conclusion matrix and accuracy.

```
In [36]: xgb_y_pred = model.predict(X_test)

In [37]: xgb_y_pred

Out[37]: array([ 4, 12,  6,  0, 11, 15, 15,  0, 14,  4,  0,  6,  8, 15, 14,  6, 11,
                 5,  1,  1, 12,  1,  6, 10,  6,  2, 12,  1,  0,  6, 13,  4, 12,  5,
                11,  7,  8, 15,  1,  7,  6,  4,  8, 11,  8,  0,  4,  0, 12, 12,  2,
                 1,  2,  8,  7,  7, 11, 15,  7,  9,  5, 15, 14, 14, 14,  1, 16, 12,
                 8, 15,  6, 12,  3,  2,  8,  6,  7, 16,  2, 13,  3,  7, 11, 12,  4,
                 2, 15,  8, 14,  4,  8,  4, 13, 15, 12,  8,  9,  5, 13, 13, 16,  0,
                 7,  3, 16, 16,  5, 10, 14,  3,  3, 10,  2,  3,  9, 15, 15, 13,  6,
                 7,  1,  0])

In [38]: xgb_cm = confusion_matrix(y_test,xgb_y_pred)
         xgb_accuracy = accuracy_score(y_test,xgb_y_pred)

In [39]: print("confusion matrics=\n",xgb_cm)
         print("  ")
         print("accuracy=",xgb_accuracy*100)

confusion matrics=
 [[ 8  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  8  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0]
 [ 0  0  6  1  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  1  5  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  8  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  5  0  0  0  0  0  0  0  0  1  0]
 [ 0  0  0  0  0  0  0  9  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 10  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  5  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 10  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  1  0  6  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  7  0  0]
 [ 0  0  0  0  0  0  5  0  0  0  0  0  0  0  0 10  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  5]]

accuracy= 91.80327868852459
```

Fig 7: Arts_datset accuracy using XgBoost

IV. METHODOLOGY

A. Algorithm

1) In the initial step, the system will take the input from the user regarding the interested stream.
2) The system will consider the user's input and based on that it will ask the user to give ratings to different subjects that could be the potential ones for future scope.
3) The system will then process all the gathered information from the user end by taking in use the already trained data.
4) The system will now predict an appropriate course using machine learning algorithms for the user based on the input received in both text as well as speech format.

©IJRASET: All Rights are Reserved

B. Module

1) The Processing: The processing in our career counselling chat bot goes from show and ask subject ratings to displaying course prediction. It follows the following procedure to predict the course-

a) Step 1- Begin with greetings and ask for the user's name for further use in the conversation. Based on the inputs, the Rasa model identifies the intent of the user responses and maps to the corresponding actions.

b) Step 2 - Then the bot asks the user for its willingness for any career guide. If the response is positive, it further asks if it can ask for further questions. If the intent is positive, it asks for the user's stream and maps this intent to the corresponding action. Action triggers to ask user's subject ratings. The subjects asked are those that can be opted in the selected particular stream.

c) Step 3 - The user's ratings on their level of interest can be between 0 to 10. The ratings are used to set the slot of the subjects and append to the ratings array.

d) Step 4 - After taking all the ratings, the bot utters if the user wants a predicted course. If yes, processing of input data takes place.

e) Step 5 - The ratings array is passed to prediction functions where the best accuracy model is called. The ratings array data is normalized i.e. scaling data between 0 and 1. Formula of normalization is -

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

f) Step 6 - The normalized data is used by machine learning models to give a numerical value that maps to the predicted course

g) Step 7 - The bot gives the predicted course along with a link to find a favourite college offering that course.

V. EXPERIMENTAL RESULTS & ANALYSIS

A. ML Model best Accuracy Results

The accuracies achieved by our subject datasets in various machine learning models are as follows-

1) PCM_dataset - 79.23%

```
[42]: import matplotlib.pyplot as plt

[43]: left=[1,2,3]
accuracy_percent = [accuracy*100,svm_accuracy*100,xgb_accuracy*100]
tick_label = ['Decision tree', 'SVM', 'Xgboost']

plt.figure(figsize = (10,8))
plt.bar(left,accuracy_percent , tick_label = tick_label,
        width = 0.9,color = ['red', 'yellow','green'])
for i in range(len(accuracy_percent)):
    plt.annotate(str(accuracy_percent[i]), xy=(left[i],accuracy_percent[i]),
                 ha='center', va='bottom')

plt.title("Accuracy comparison")
plt.ylabel("Accuracy in PCM")
plt.xlabel("ML classifiers")

plt.show()
```

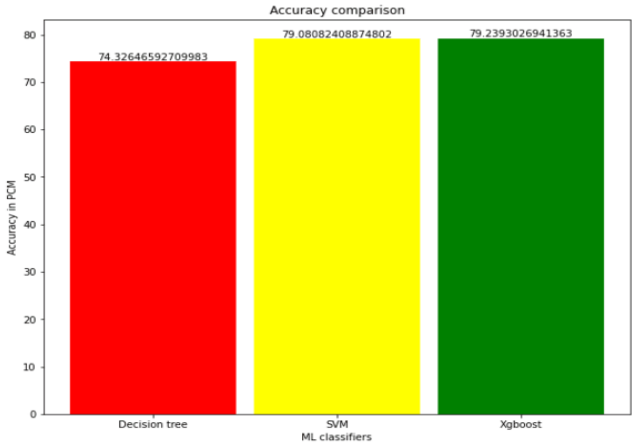


Fig 8: Accuracy comparison for PCM_dataset

©IJRASET: All Rights are Reserved

2) PCB_dataset - 97.09%

```
In [41]: import matplotlib.pyplot as plt

In [42]: left=[1,2,3]
accuracy_percent = [accuracy*100,svm_accuracy*100,xgb_accuracy*100]
tick_label = ['Decision tree', 'SVM', 'Xgboost']

plt.figure(figsize = (10,8))
plt.bar(left,accuracy_percent , tick_label = tick_label,
        width = 0.9,color = ['red', 'yellow','green'])
for i in range(len(accuracy_percent)):
    plt.annotate(str(accuracy_percent[i]), xy=(left[i],accuracy_percent[i]),
                 ha='center', va='bottom')

plt.title("Accuracy comparison")
plt.ylabel("Accuracy in PCB")
plt.xlabel("ML classifiers")

plt.show()
```

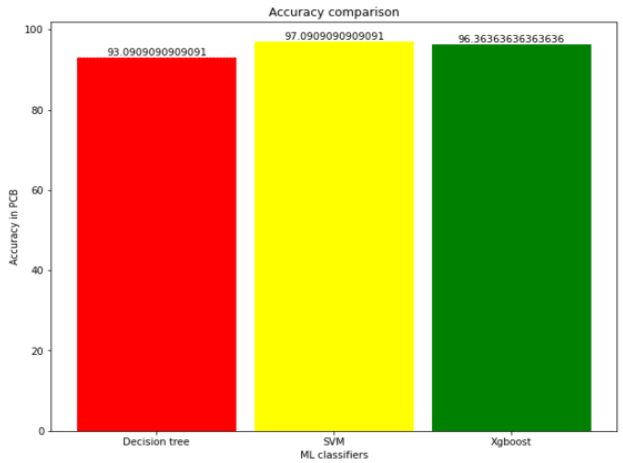


Fig 9: Accuracy comparison for PCB_dataset

3) Commerce_dataset - 99.6%

```
In [41]: import matplotlib.pyplot as plt

In [42]: left=[1,2,3]
accuracy_percent = [accuracy*100,svm_accuracy*100,xgb_accuracy*100]
tick_label = ['Decision tree', 'SVM', 'Xgboost']

plt.figure(figsize = (10,8))
plt.bar(left,accuracy_percent , tick_label = tick_label,
        width = 0.9,color = ['red', 'yellow','green'])
for i in range(len(accuracy_percent)):
    plt.annotate(str(accuracy_percent[i]), xy=(left[i],accuracy_percent[i]),
                 ha='center', va='bottom')

plt.title("Accuracy comparison")
plt.ylabel("Accuracy in Commerce")
plt.xlabel("ML classifiers")

plt.show()
```

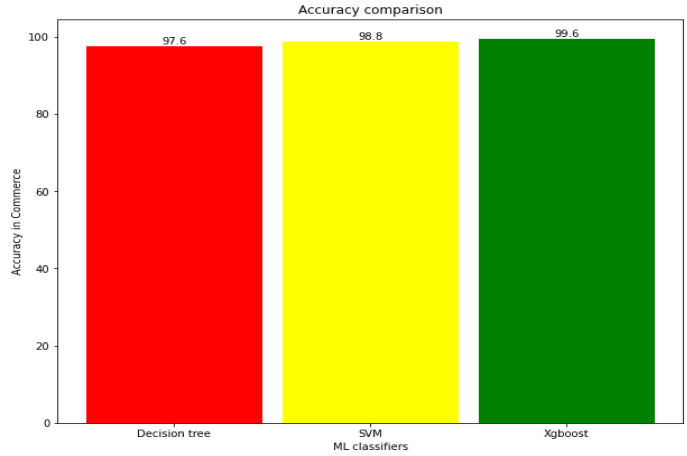


Fig 10: Accuracy comparison for Commerce_dataset

©IJRASET: All Rights are Reserved

4) Arts_dataset - 92.62%

```
[41]: import matplotlib.pyplot as plt

[42]: left=[1,2,3]
accuracy_percent = [accuracy*100,svm_accuracy*100,xgb_accuracy*100]
tick_label = ['Decision tree', 'SVM', 'Xgboost']

plt.figure(figsize = (10,8))
plt.bar(left,accuracy_percent , tick_label = tick_label,
        width = 0.9,color = ['red', 'yellow','green'])
for i in range(len(accuracy_percent)):
    plt.annotate(str(accuracy_percent[i]), xy=(left[i],accuracy_percent[i]), ha='center', va='bottom')

plt.title("Accuracy comparison")
plt.ylabel("Accuracy")
plt.xlabel("ML classifiers")

plt.show()
```

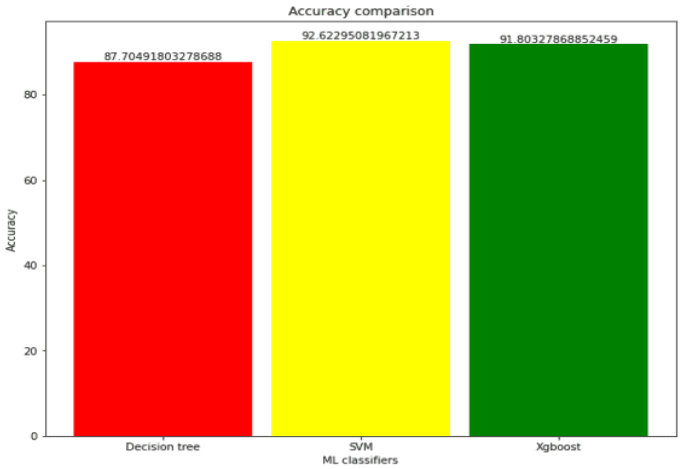


Fig 11: Accuracy comparison for Arts_datset

B. Exemplary Rasa based Chatbot Conversation Results

Below shown is an exemplary conversation of our designed chatbot system.

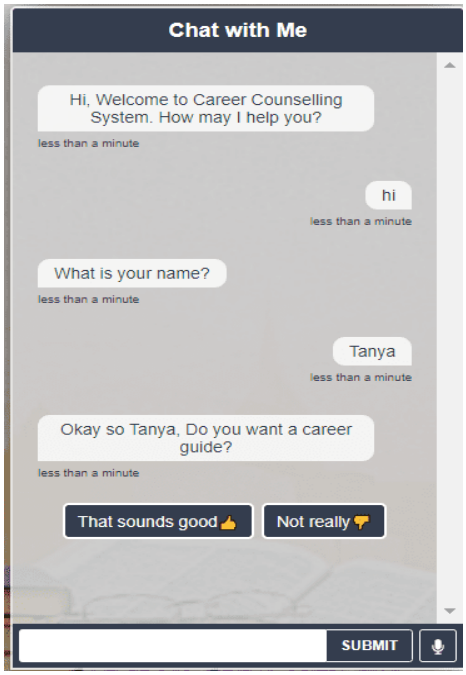


Fig 12: Exemplary chatbot greeting conversation

©IJRASET: All Rights are Reserved

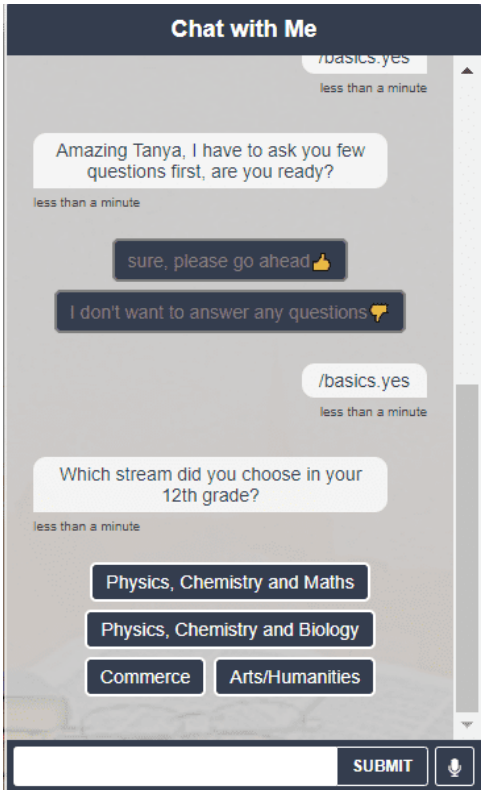


Fig 13: Exemplary chatbot input conversation

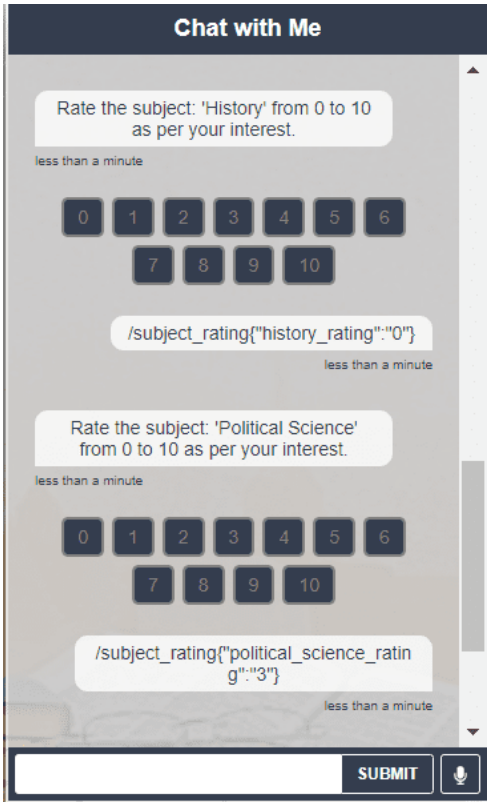


Fig 14: Exemplary chatbot input rating

©IJRASET: All Rights are Reserved

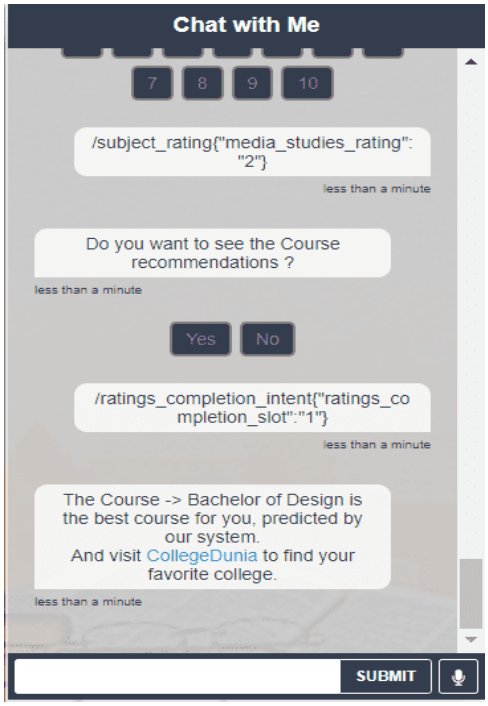


Fig 15: Exemplary Chatbot result conversation

The Rasa NLU module has helped us to build an intelligent system that offers the feature of conversation with the users on the basis of their responses. Their intentions as intents are being mapped to the actions under stories in Rasa and results in constructing a meaningful pattern for the chatbot.

Further, the user's will provide ratings between 0 to 10 for each subject as per their interests and strengths that forms an unlabelled dataset that is required to be predicted. For the same, various machine learning models are taken in use to predict the best suited course as the target to the input unlabelled dataset.

Finally , the predicted course will be recommended as output to the user .

VI. CONCLUSION & FUTURE WOR

The designed system will act as a real counsellor and will assist the user to plan and pursue a ‘true career’ based on his/her interest, choices and primary specifications. This system aims to help the students who are at their major crossways in life, struggling to make the right career move by identifying the right available choices for them.

The bot mainly assists the users who have passed their SSC or HSC and even some other random users to know in which stream and domain their strength lies.

In future reference, the system will implement more career fields, domains and branches that could be the potential choices so that the user can have even more options to make decisions over. Moreover, the system will cover more subjects that should be considered for rating after the selection of a particular stream. The target audience and covered branches will enfold a wider area in future.

REFERENCES

[1] Archana S. Parab,Siddhesh Palkar and Satish Maurya “An Intelligent Career Counselling Bot A System for counselling”, International Research Journal of Engineering and Technology (IRJET) , Volume: 04 Issue: 03 | Mar -2017 .

[2] Faraaz Ansari, Shaikh Saad, and Shaikh Shareem ,”Career Counselling Chatbot”, International Research Journal of Engineering and Technology (IRJET) , Volume: 06 Issue: 04 | Apr 2019.

[3] R. Moore, G. Gibbs, “Emile:Using a Chatbot Conversation to enhance the learning of Social Theory” Univ. of Huddersfield, Huddersfield, England,(2002) Ausburg, Germany, University of Ausburg. 2002.

[4] JoAnn Harris-Bowlsbey, Ed.D., Kuder Research Faculty “Overview of Career Guidance: Its Foundations, Objectives, and Methodology”, White Paper, Kuder.

[5] Ohm, Akshansh & K, Bhavani. (2019). Chatbot for Career Guidance Using AI. International Journal of Computer Sciences and Engineering. 7. 856-860. 10.26438/ijcse/v7i6.856860.

[6] D’Silva, Godson & Jani, Megh & Jadhav, Vipul & Bhoir, Amit & Amin, Prithvi. (2020). Career Counselling Chatbot Using Cognitive Science and Artificial Intelligence. 10.1007/978-981-15-3242-9_1.

©IJRASET: All Rights are Reserved

[7] Danyal Zaidi,Sahil Raza and Lalit Sharma,"Artificial Intelligence Based Career Counselling Chat Bot A System for counselling ", Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 6, 2021, Pages. 11732 – 11735

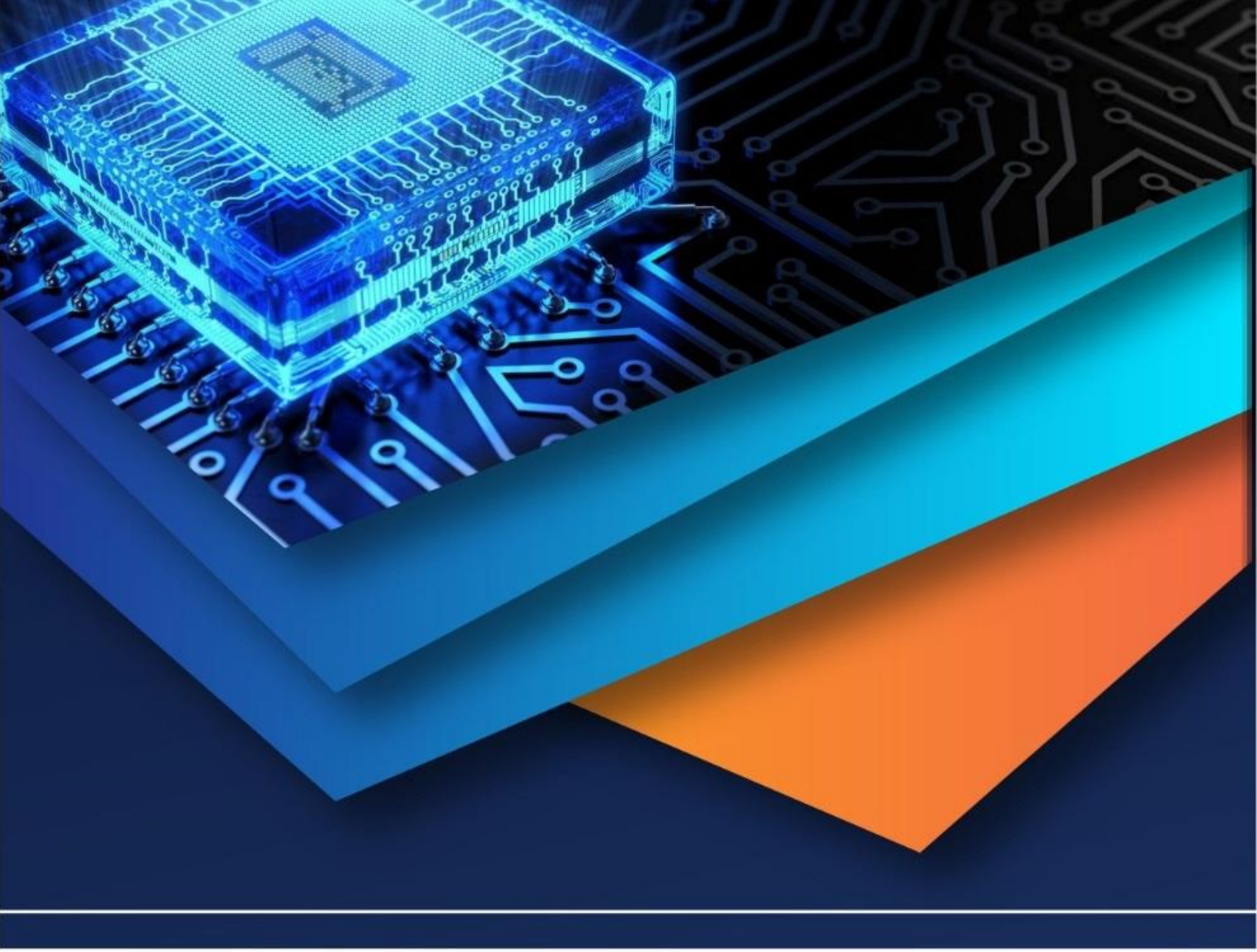
[8] Nangineni Vijaya Lakshmi, Mandadapu Hema, Kondepati Gowthami & Mrs. Jaladi Praveena ,"Artificial Intelligence Based Career Guidance Chatbot", Journal of Engineering, Computing and Architecture Volume 10, Issue 5, 2020 ISSN NO: 1934-7197

[9] Vijayakumar R, Bhuvaneshwari B, Adith S, Deepika M ,"AI-Based Student Bot for Academic Information System using Machine Learning".

[10] Khanna, Anirudh & Pandey, Bishwajeet & Vashishta, Kushagra & Kalia, Kartik & Bhale, Pradeepkumar & Das, Teerath. (2015). A Study of Today's A.I. through Chatbots and Rediscovery of Machine Intelligence. International Journal of u- and e-Service, Science and Technology. 8. 277-284. 10.14257/ijunesst.2015.8.7.28.

[11] Y. Wu, G. Wang, W. Li, and Z. Li, "Automatic Chatbot Knowledge Acquisition from Online Forum via Rough Set and Ensemble Learning," Proc. IEEE of 2008 IFIP International Conference on Network and Parallel Computing, 2008, pp. 242-246, doi:10.1109/NPC.2008.24.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)