Propensity Life Tracker – Ariadne Project

Final Report

12-04-2022

Team members:

Abhinandan Umakant Jawalekar

Jayarani Rajesh

Karthikeyan Suresh Kumar

Acknowledgment

We would like to thank a few people for helping make this research project possible.

Thank you to our client, Teresa Woods Snelgrove, for giving us this opportunity to work on this innovative and engaging project.

Finally, we would like to thank Thecla Oreme and Brunilda Xhaferllari, for their support and guidance throughout the project.

Contents

Acknowledgment 2

Our Team 5

Introduction 6

Project Description 6

Objective 6

Methodology 7

Rationale 7

Requirements 8

Research Findings 9

Outcomes and Benefits 10

- Deliverables 11

Process 11

- Process Visualization 12

Tools and Software 14

Deliverables Detailing 17

- Dataset Collection and Labelling 17
- Database Design and Architecture 18
- Classification Models 20
  - Sentiment Classification Model 20
  - Taxonomy Classification model 21
  - Tense Classification model 22
  - Subject Classification Model: 24
- Analysis Dashboard 25
  - Home Page 25
- Information Page 26
- Analytics Dashboard 27
- Sentiment Analysis Dashboard 28
- Mood Analysis Dashboard 29
- Tense Analysis Dashboard 30

Conclusion 30

References 31

Table of Figures

Figure 1:Requirements for phase 2 8
Figure 2: Cycle of our project 14
Figure 3:Home page of Propensity dashboard 25
Figure 4: Information page of propensity dashboard 26
Figure 5: Propensity dashboard 27
Figure 6: Sentiment analysis dashboard 28
Figure 7: Mood analysis dashboard 29
Figure 8: Tense analysis dashboard 30

Our Team

Abhinandan Umakant Jawalekar

I am a graduate with a bachelor's in Computer Applications. I have good experience in dealing with computers and various programming languages. I also have completed a Pro degree in Data Science which helped me enhance my analysis skills. Later I worked on a project with a bank which helped me develop my skills like documenting, analyzing, etc.

Jayarani Rajesh

I'm a B.Sc Statistics graduate who completed the program in 2020. I have excellent statistical and mathematical skills required for Data Science. I have good knowledge of statistical analysis and data cleaning of the datasets. I am good with the Univariate, Multivariate, Inferential techniques which are primarily used in the data cleaning process.

Karthikeyan Suresh Kumar

Bachelor of Engineering, Electronics, and Communication.

I have 3+ years of experience as a Software Developer in HTC Global Services. The major chunk of my work is focused on the coding, testing, and execution of web applications for various banks. I have expertise in programming languages – Java and python, Frameworks – Spring and Hibernate, SQL, Webservices – REST, and SOAP.

Introduction

Propensity-Life Tracker is a project idea by Teresa Woods Snelgrove and she is the project lead of this innovative project. Teresa is a successful serial entrepreneur who has had success in building numerous businesses.

Teresa is accompanied by John Pickard, who is an exceptional entrepreneur, whose expertise and inputs would bring additional value and insights to the project. Teresa's vision of creating this project would in turn changes the lives of many individuals who are suffering from mental illness and stress. This project is a tool that would have a wide reach of audiences as it is can be used for monitoring the well-being of the kids from school, monitoring old people in homes, in hospitals for recovering patients, and so on.

Project Description

Propensity is a tool that helps a user capture a succession of thoughts or memories through dialogues/conversations with an intelligent agent (Like Alexa or Siri). Our agent is called "Ariadne". Through this tool, people can have conversations with Ariadne, which in turn analyses those conversations and converts them into meaningful reports which highlight the sentiment, subject, tense, and mood analysis of a person.

The tool could have alternative uses: A mood monitor, mental health monitor, sentiment analysis, bias analysis, etc. It could be offered as a wearable device, a smartphone app, etc.

Objective

Further development of the project, from the end of Phase One. Phase One of the development of the project largely focused on the initial research and the design of the Machine Learning (ML) models. The team has developed two ML models: one for the classification of the Taxonomy and the other model is for the Sentiment Classification.

The Taxonomy classification model is based on the Naïve Bayes algorithm and this model has a prediction accuracy of 25%. The Sentiment classification model is based on the VADER (Valence Aware Dictionary for Sentiment Reasoning) model and it has a prediction accuracy rate of 70%.

Phase Two of the project involves improving the performance of the working ML models for the Taxonomy and Sentiment classification, developing new models for Subject and Tense classification, creating datasets for training and testing the models, connecting the models with the database to store data, creating visualizations for the output data.

Methodology

Phase Two of the project was split into three parts – Research, Data collection and labeling, and model and dashboard development. The Research part largely focused on machine learning which would be a good fit for the problem/requirement which is at hand. It involved topics and data from various white papers, blogs, etc. The data collection and labeling was the process of collecting the conversational data from Open sources. Once the data was collected, the data needs to be custom labeled to train the machine learning model. This process is essential because the machine learning model predicts the needed results based on the data that is used for training.

Rationale

With technology advancing in all aspects of life, it should be advancing in the field of mental health too, which is one of the reasons for the birth of Project Propensity. Mental health issues vary in people of any age, from school kids to old people. Everyone might have experienced mental health-related issues at least once in their lifetime. Also, in the post-pandemic world, everyone around the world might have been through some mental stress: they might have lost a loved one, they would have lost their jobs, they might have been stuck wherever they are unable to reach home, etc. Everyone would

have gone through something which would have affected them mentally. The Project Propensity might be a one-point diagnosis solution for that, the Ariadne agent listens to our conversations, analysis the data, and provides a report which would help to know their wellbeing from time to time.

Requirements

- Find possible ways to improve and enhance the performance of the current working models.
- Create new datasets/monologues to train the model.
- Design and develop a database to save the input data to be consumed and the output data generated by the machine learning model.
- Research on machine learning models which would be the best fit for the Tense and Subject classification.
- Implement the finalized machine learning algorithm for the Tense and Subject classification.
- Develop interactive reports based on the data from the ML models.

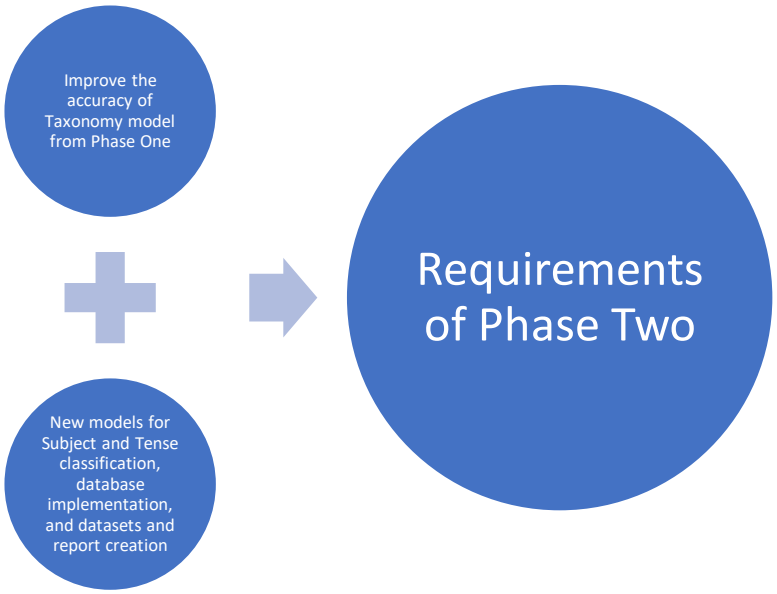

Figure 1:Requirements for phase 2

Research Findings

The following research questions are the key challenges in Phase Two and researching and analyzing those paves the way for the project.

- How to improve the accuracy of the Taxonomy model. Should we change the current ML model or find techniques to improve the accuracy.

The Taxonomy model was designed and built using the Naïve Bayes classifier algorithm in Phase One of the project. This model gave an accuracy of 25%. So there was a need to improve the accuracy of the model to make a realistic prediction of data. To increase the prediction accuracy of the model, there was a need to change the model. The Naïve Bayes classifier was replaced with the Logistic Regression model and this model gave a prediction accuracy of approximately 70%.

- What models would be the best fit for Subject and Tense classification?

The Subject and Tense classification model were the new ML models which were built from scratch in Phase Two. After extensive research, for the Subject model, the spacy library was used and for the Tense classification model, the Logistic Regression has been used. The former model is a pre-trained model which needed no training or labeled datasets to make predictions. Whereas the latter is a model which is a custom-trained one based on the training from the labeled datasets.

- How can we create/find data that matches or closely corresponds to real-life conversations?

The project is mostly based on the conversational data of everyday life as the user would be talking/texting our agent Ariadne. So finding the necessary data that accurately reflected the real-world data posed a challenge. After browsing and searching hundreds of sites and white papers, the MELD dataset which was used in a white paper provided to be a good dataset to train the models. This dataset is a

conversational dataset between Joey and Chandler from the American Sitcom "F.R.I.E.N.D.S".

- What database can be used to store the processed data? And how to design it?

With numerous databases available in the market, it was tough to find a good database to store the application data. After some extensive research, the MySql database was chosen to be an apt candidate for storing the data. MySql is an open-sourced relational database management system. Also, for the design part, the database was designed in such a way that it stores the raw data, that was required by the models for training purposes, and also the test data along with the prediction values.

- There are many visualization tools in the market. Which tool would be apt for our analysis visualization?

Once the data has been ingested and predictions have been made, the results must be shown using visuals to make them more understandable and appealing to the end-user. Also, the application requires a dashboard that summarizes the data and shows the necessary and needed insights to the user. For designing the dashboard, Microsoft Power BI was used. Power BI is one of the leading visualization tools and has many features and cloud infrastructure. This software has all the necessary components to design and show the necessary insights in a dashboard.

Outcomes and Benefits

With the completion of Phase Two, the core part of the project is almost over. The outcomes and benefits of completion of Phase Two are the following:

- Two new models were added in Phase Two of the project for Subject and Tense classification.

- Database has been implemented so that the team can view and analyze the past data.
- As the data is stored, the visuals developed can be across a wide time frame.
- The core part of the project is completed, so that the next phase of the project which might include building UI and developing app/wearables tech can proceed.
- A SharePoint site is created containing all the necessary research documents, analysis documents, workflow documents, source code, sample datasets, etc.
- Client has an idea of what has been completed so far and undertakes the next necessary steps to carry forward the project.

Deliverables

With the completion of Phase Two, the following components will be delivered to the clients.

1. Improved ML models for Taxonomy Analysis
2. Model for Tense classification
3. Model for Subject classification
4. Datasets used for training/testing the models
5. Database design architecture and scripts
6. Visualization of the analysis results
7. Project-related documents
8. Reference documents

Process

The process followed in Phase Two of the project is as follows:

1. Research

2. Design
3. Development
4. Testing
5. Validation
6. Result Analysis
7. Documentation

Process Visualization

Research:

This is the key stage of the project as it involves analyzing the various machine learning algorithms which would prove to be competent for the type of classification chosen. The advantages and disadvantages of the models are analyzed and their efficiencies are compared to choose the right model which would fit our classification problem.

Design:

The Design part of the model should be concentrated because only an efficient design would lead to an efficient model. The Design part involves how the data is handled and how the model is being fed with input data and what would be the expected output data. It also covers the design of the database.

Development:

The development process closely follows the design process and follows the steps as per the design process in building the model. All the logic and the functions intended to be followed will be developed with the help of the IDEs and the supporting python libraries.

Testing:

Once the development of the model is complete, the testing should be done to check whether the model is working as expected or is having any issues/errors. The datasets to train and test the model must be either created or curated from websites.

Validation:

After testing with the training data, the results of the model must be validated to check whether the model is generating the results as expected. If the efficiency of the model is not satisfactory, then ways to optimize the model's efficiency should be explored and implemented.

Result Analysis:

After all the above process is completed successfully and the expected results are obtained, various visualization software must be used to create reports, dashboards, charts, etc. Only with the proper analysis, it can be ensured that the model is created as per the requirements and the expectations.

Documentation:

This is a crucial process and must be carried out in all the stages of development. Documenting all the processes, requirements, steps, analysis, and results of the project provides an insight to the clients as well as helps anyone or any team to work on the project with ease. Also, the documents are a part of the deliverable, so they must be followed in each process.

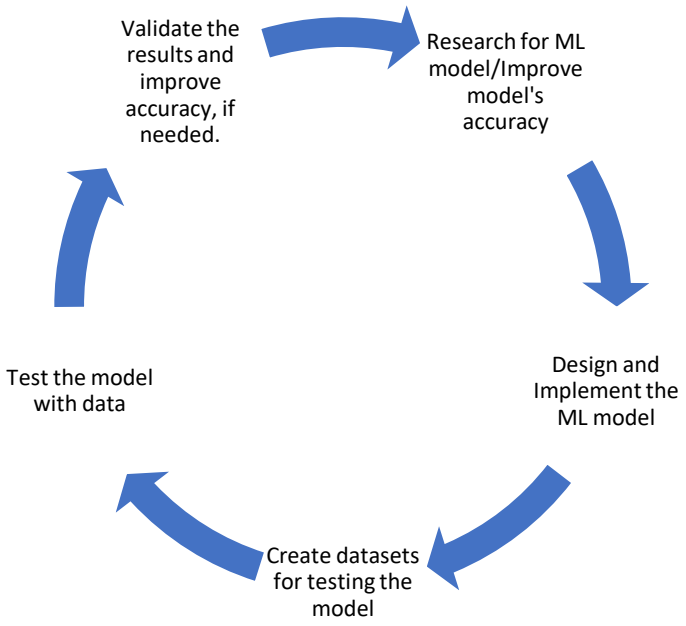

Figure 2: Cycle of our project

Tools and Software

The following tools have been used in the project and the same will be used for further development.

- Python 3. x
- Python Libraries: Numpy, Pandas, NLTK, Vader, Text blob, sci-kit learn, etc.
- Tableau 2021.2.4
- PyCharm
- Visual Studio Code
- MS-Office Suite
- Microsoft Power BI
- Google Collab Notebooks
- Microsoft SharePoint

Python 3. x: Python is an interpreted high-level general-purpose programming language. It is the most used programming language for building machine learning models, artificial intelligence, data science, and even web application development. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs, as well as its object-oriented approach, aim to help programmers write clear, logical code for small and large-scale projects

Python Libraries:

Various python libraries like Numpy, Pandas, Scikit-learn, NLTK, Vader, Text blob, etc., are the core libraries used for building the machine learning models. The major libraries used in the project are as follows:

Numpy: NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Pandas: Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

NLTK: The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language.

Scikit-learn: Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means, and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

VADER: VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments.

Text Blob: TextBlob is a Python library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, etc.

Tableau: Tableau is a visual analytics platform transforming the way we use data to solve problems—empowering people and organizations to make the most of their data.

PyCharm: PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains (formerly known as IntelliJ). It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda.

Visual Studio Code: Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux, and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

Microsoft Office Suite: Microsoft Office, or simply Office, is a family of client software, server software, and services developed by Microsoft. It consists of Microsoft Office, Microsoft Excel, Microsoft PowerPoint, etc., bundled together.

Microsoft Power BI: Power BI is an interactive data visualization software product developed by Microsoft with a primary focus on business intelligence. It is part of the Microsoft Power Platform. Power BI is a collection of software services, apps, and connectors that work together to turn unrelated sources of data into coherent, visually immersive, and interactive insights. Data may be input by reading directly from a database, webpage, or structured files such as spreadsheets, CSV, XML, and JSON.

Google Collab Notebooks: Google Colaboratory (also known as Colab) is a free Jupyter notebook environment that runs in the cloud and stores its notebooks on Google Drive. Colab, or "Colaboratory", allows you to write and execute Python in your browser.

Microsoft SharePoint: SharePoint is a web-based collaborative platform that integrates with Microsoft Office. Launched in 2001, SharePoint is primarily sold as a document management and storage system, but the product is highly configurable and its usage varies substantially among organizations.

Deliverables Detailing

Dataset Collection and Labelling

The project is based on analyzing the conversations between Ariadne and a user. So the data required for training the models must closely resemble the real-world conversations. Finding such data was a huge problem as most of the NLP datasets available online were not conversational.

After a lot of research and searching online, the MELD dataset was extracted from a white paper, which had conversations from the American Sitcom F.R.I.E.N.D.S.

Once the dataset was extracted it had to be labeled manually to train the models. The manual labeling of the data was time-consuming. After manually labeling the data, the data was stored in a MySql database using a python program that reads the CSV/excel files and then inserts those records into the input_table of the database.

Sentiment Model: As this is a pre-trained model, it doesn't require labeling.

Taxonomy/Mood Model: The labels for this model were anger, disgust, fear, joy, sadness, surprise, and neutral. Based on these labels the sentences were labeled for training the model.

Tense Model: The Tense model is custom-trained. So, the data was labeled for this model. The labels were past, present, and future.

Subject Model: This model is based on a pre-trained model; hence labeling was not necessary.

Database Design and Architecture

In Phase One of the project, all the input and output data were handled as files. This approach could work better in the short term but will have performance issues in the long run.

This was the reason why storing the data in the database is a better option. There are several advantages to this, such as,

- The implementation of this database would help to create a centralized repository for the raw data to be used for the models.
- Avoids the need to maintain multiple files for different models.
- Data from the database can be exported into multiple file formats if needed.
- Would be easy to search a particular data using SQL queries.
- The predicted output data can be maintained, and we can use them to compare performances if we are changing the ML model for prediction.
- Quicker processing of a large amount of data.

To store and retrieve data, the MySql database was used. MySql is an open-sourced database that has good support for python and Jupyter notebooks. The MySql Server and MySql client has been used to create databases, tables, and querying data.

The database has two tables. They are:

- **Input_data**: This table holds the raw labeled data to be used by the models.

- **Output_data**: This table holds the predicted output data from the different models.

Input_data:

The input data table has the following columns:

Column_name	Datatype	Description
sentence	varchar(300)	The sentence in a conversation
sen_sentiment	varchar(20)	The sentiment of the sentence
sen_taxonomy	varchar(20)	Taxonomy of the sentence
sen_subject	varchar(20)	The subject of the sentence
sen_tense	varchar(20)	Tense of the sentence
added_by	varchar(20)	Name of the person who worked on the labeling of the dataset
added_date	varchar(20)	Date when the record has been added to the database

Table 1: Raw input data table for models

Output_data:

The output_data table has the following columns:

Column_name	Datatype	Description
sentence	varchar(300)	The sentence used in the prediction
predicted_sentiment	varchar(20)	The predicted sentiment of the sentence
predicted_taxonomy	varchar(20)	Predicted taxonomy of the sentence
predicted _subject	varchar(20)	The predicted subject of the sentence
predicted _tense	varchar(20)	The predicted tense of the sentence
added_by	varchar(20)	Name of the model inserting the record
added_date	varchar(20)	Date when the record has been added to the database

Table 2: Raw output data table for models

Classification Models

The conversation data from users were analyzed for four different subjects, namely

- Sentiment
- Taxonomy/Mood
- Tense
- Subject

Sentiment Classification Model

Sentiment analysis (or opinion mining) is a natural language processing (NLP) technique used to determine whether data is positive, negative, or neutral. It is the process of detecting positive or negative sentiment in text. It's often used by businesses to detect sentiment in social data, gauge brand reputation, and understand customers.

Since humans express their thoughts and feelings more openly than ever before, sentiment analysis is fast becoming an essential tool to monitor and understand the sentiment in all types of data. Our application heavily relies on capturing the sentiment of a conversation as it is one of the key elements in judging a customer's behavior.

The Vader model is the backbone of the Sentiment classification model. VADER (Valence Aware Dictionary for Sentiment Reasoning) is a model used for text sentiment analysis that is sensitive to both polarity (positive/negative) and intensity (strength) of emotion. It is available in the NLTK package and can be applied directly to unlabeled text data.

VADER sentimental analysis relies on a dictionary that maps lexical features to emotion intensities known as sentiment scores. The sentiment score of a text can be obtained by summing up the intensity of each word in the text.

Process

- The data preprocessing stage involves removing stopwords and unwanted characters using regex from the sentences.
- After the sentences are cleaned, the words in the sentences are split and the lemmatization technique is applied to the words.
- Once it has been done, all the words are put up back together to form sentences.
- Since the VADER model is pre-trained, it does not require the labeling of data for making predictions.
- After the data preprocessing and cleaning, the data was fed to the machine learning model to make predictions.
- The output of the machine learning process is the predicted sentiment of the sentences.
- This predicted output can be saved as an excel file or stored in a database.

Taxonomy Classification model

The taxonomy classification model is more of a mood classification model. Here the emotion/mood from the sentences of a conversation will be captured. The sentences were labeled with different moods such as anger, disgust, fear, joy, sadness, surprise, and neutral.

For the Taxonomy model, the model built in Phase One was showing an accuracy of 20%. After a lot of research, different techniques were implemented to improve the accuracy of the model. But even after fine-tuning the model, it was not showing a considerable increase in accuracy. Hence, the taxonomy model was rebuilt from the scratch using a different ML algorithm.

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. It is another powerful supervised ML algorithm used for binary classification problems (when the target is categorical). It essentially uses a logistic

function to model a binary output variable. Hence, Logistic Regression seemed to be an ideal candidate for the Taxonomy model.

Process:

- Logistic Regression is a supervised learning algorithm, so the dataset needs to be labeled.
- The sentences are manually labeled with the seven different labels marking the emotion/mood of each sentence.
- After labeling, the data is then cleaned for stopwords and unwanted characters are removed using the regex.
- Then the sentences are split into words and lemmatization is applied to those words.
- The lemmatized words are joined back to a sentence and stored in a list.
- The LabelEncoder is then used on the target variable and vectorization is carried out on the dataset.
- Then the dataset is split into a training dataset and a test dataset.
- The model is trained with the training dataset and after that, the model is fed with the testing dataset to make predictions.
- The model gives the prediction output for the test dataset.
- The prediction results and actual results are used to find the accuracy of the model.

Tense Classification model

Analyzing the conversations for Sentiment or Mood is not enough to understand the mental well-being of an individual. The Sentiment and the Mood of the individual may vary with past experiences, present circumstances, or future decisions. So, it is really

important to extract the tense of the conversations and use them along with Sentiment and Mood to enhance our findings on the mental wellbeing of an individual.

The Tense model is a custom-built supervised learning model. The logistic regression algorithm is used for building the model. This is the same algorithm used in the Taxonomy/Mood model.

Process:

- The process for building the Tense classification model is the same as the Taxonomy model.
- Logistic Regression is a supervised learning algorithm, so the dataset needs to be labeled.
- The sentences are manually labeled with the three different labels marking the tense of each sentence.
- After labeling, the data is then cleaned for stopwords and unwanted characters are removed using the regex.
- Then the sentences are split into words and lemmatization is applied to those words.
- The lemmatized words are joined back to a sentence and stored in a list.
- The LabelEncoder is then used on the target variable and vectorization is carried out on the dataset.
- Then the dataset is split into a training dataset and a test dataset.
- The model is trained with the training dataset and after that, the model is fed with the testing dataset to make predictions.
- The model gives the prediction output for the test dataset.
- The prediction results and actual results are used to find the accuracy of the model.

Subject Classification Model:

Whenever there is a conversation between two people it will be based on something, that's what is called the subject. Similarly, when users talk to Ariadne they talk about their interests, their experiences, their families and friends, etc. Users mostly have meaningful conversations with Ariadne. Finding out the subject of the conversations may help us analyze what is the mood or sentiment attached to that particular subject and how it can affect the mental health of the individual in the future.

For the Subject classification model, the pre-trained model from the Spacey library has been used. This ML algorithm generates different tags based on the sentences it reads. From these tags, the subject-related tag is filtered out to identify the subject of the sentence.

Process:

- For the Subject classification model, the pre-trained model from the Spacey library has been used.
- This ML algorithm generates different tags based on the sentences it reads.
- From these tags, the subject-related tag is filtered out to identify the subject of the sentence.
- Since the Spacey model is pre-trained, it does not require the labeling of data for making predictions.
- The data was fed to the machine learning model to make predictions.
- The output of the machine learning process is the subject of the sentences.
- This output can be saved as an excel file or stored in a database.

Analysis Dashboard

The results from all the models will be either saved as a file or stored in a database. With those data, the insights of the conversations between the user and Ariadne can be populated into a dashboard. The following explains the various parts of our user dashboard.

Home Page

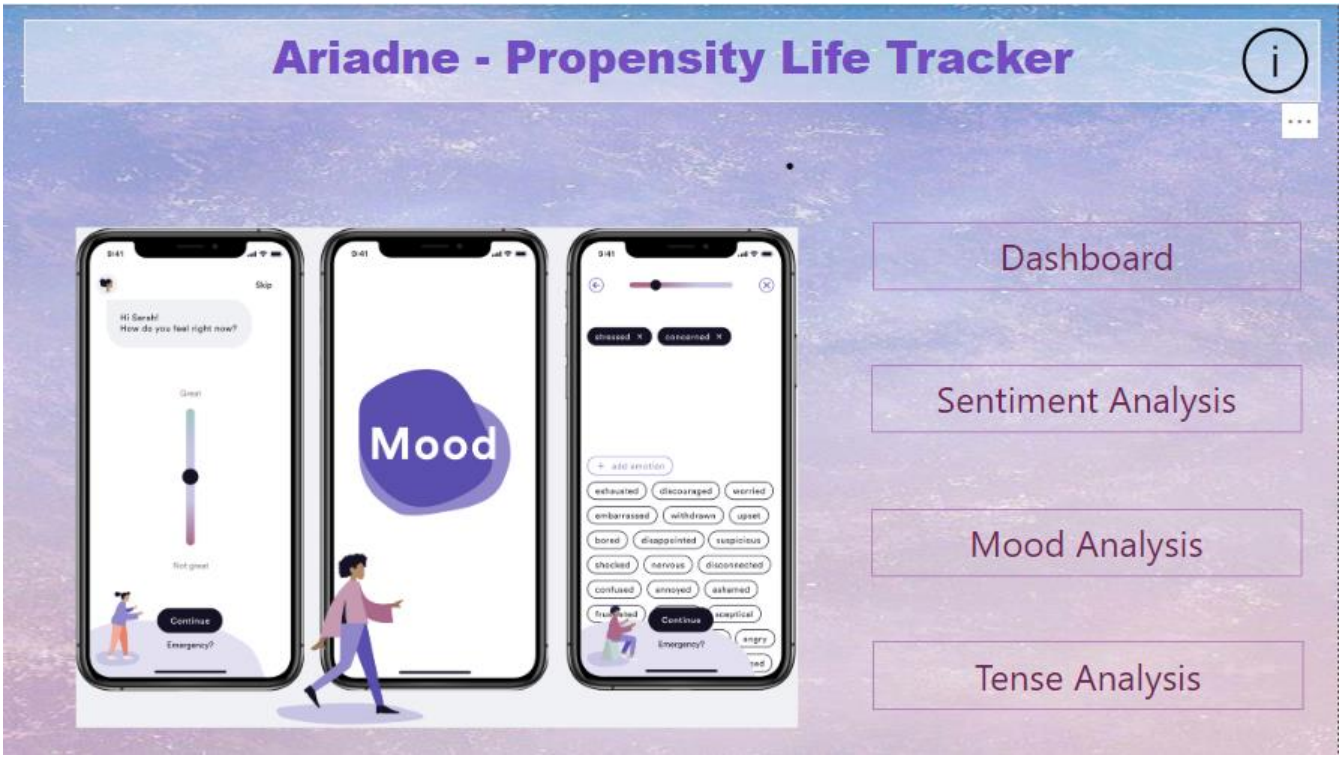

Figure 3:Home page of Propensity dashboard

Background portrait from https://in.pinterest.com/pin/657948445475079 87/

Image from https://bootcamp.uxdesign.cc/ux-ui-case-study-a-mood-tracking-app-that-identifies-your-stress-triggers-efa3467e791

The Home Page is the first page that the user sees. This is the homepage of our Propensity website or the mobile app. This page has links to the four dashboards and an information link to know more about the Propensity application.

Information Page

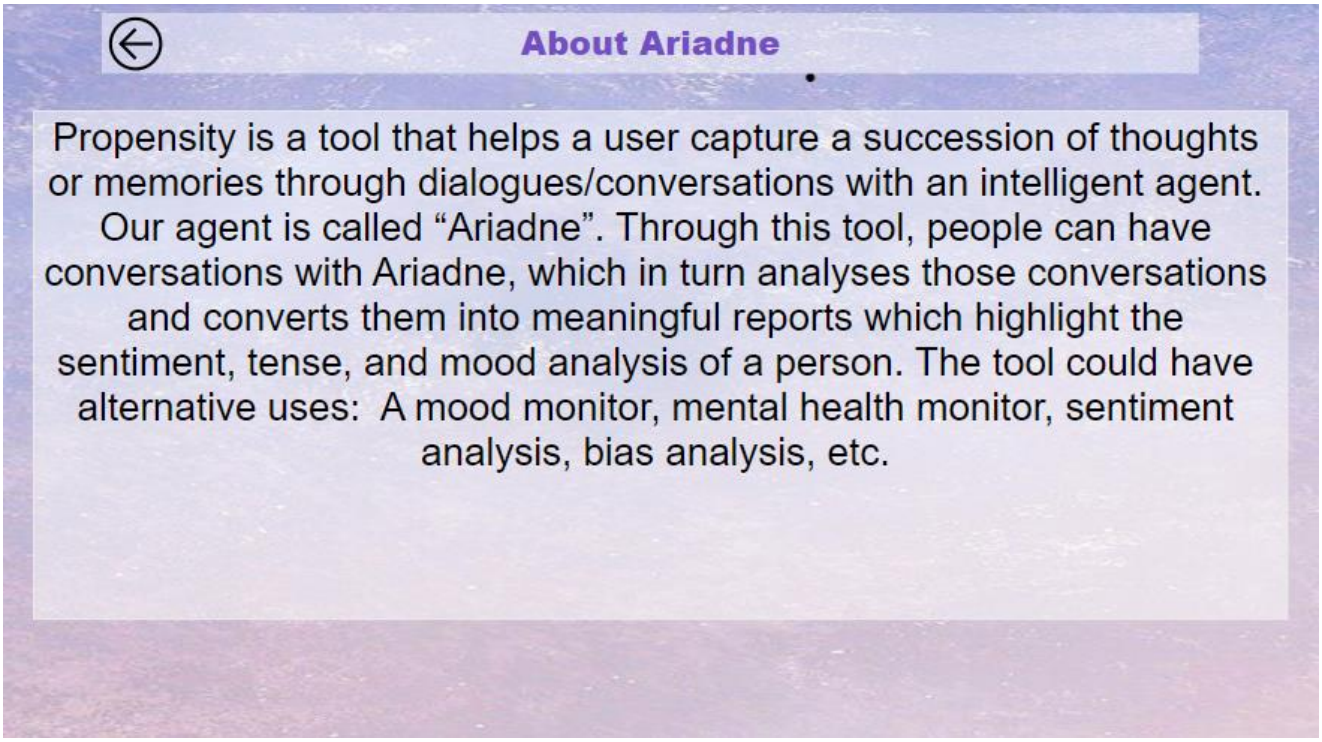

Figure 4: Information page of propensity dashboard

The information page displays information about the Propensity application and the Ariadne agent. There is a back button to navigate back to the Home Page.

Analytics Dashboard

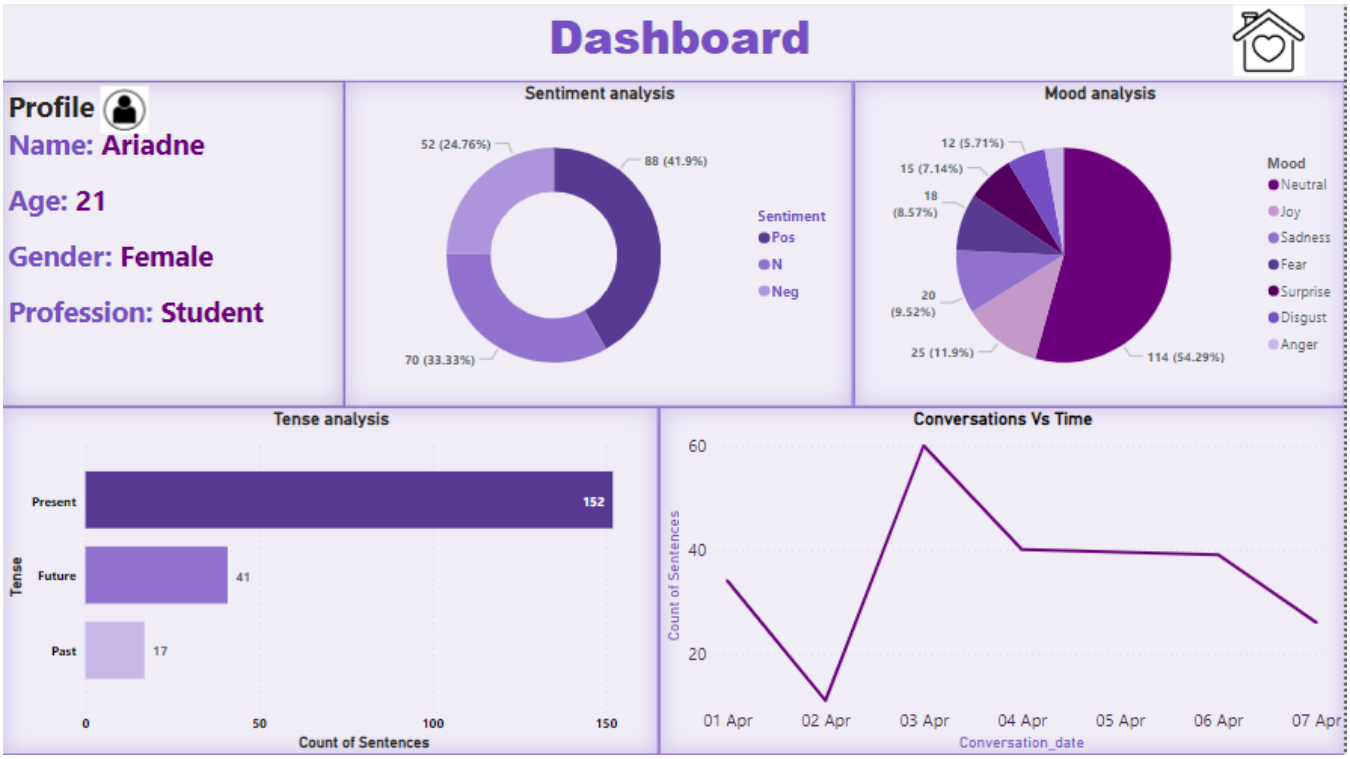

Figure 5: Propensity dashboard

The Analytics Dashboard is the overall dashboard that summarizes the data and the insights obtained from the output of all the models. This dashboard contains the following parts

- Text Box: The text box component is used to hold the user details, such as name, age, gender, and profession, in place.
- Donut Chart: The output of the Sentiment analysis model is summarized here with all the three sentiment types into a donut chart.
- Pie Chart: This visual holds the data from the Taxonomy summarized to show their percentages.
- Horizontal Bar Graph: The output of the Tense classification model is shown here which denotes what the user is mostly talking about either the past, present, or the future.

- Line Chart: The line chart is used to show the user how often he/she is conversing with Ariadne. This visual shows the number of conversations a user is having over some time.
- Home Button: A button is provided for the convenience of the user to navigate back to the Home Page.

Sentiment Analysis Dashboard

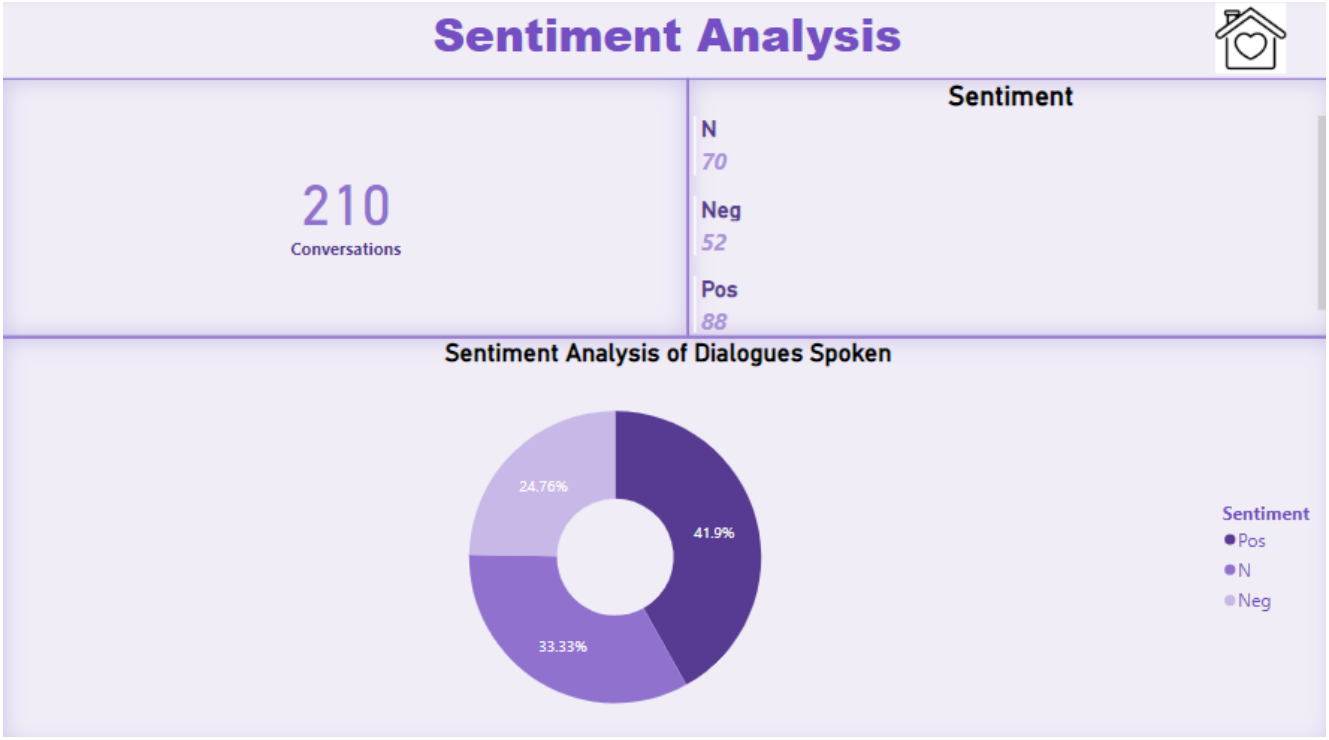

Figure 6: Sentiment analysis dashboard

The individual component dashboards are provided to users if they want to have a deeper understanding of any particular data. Here the Sentiment model data is shown to the user where the user can have more insights from the dashboard.

Mood Analysis Dashboard

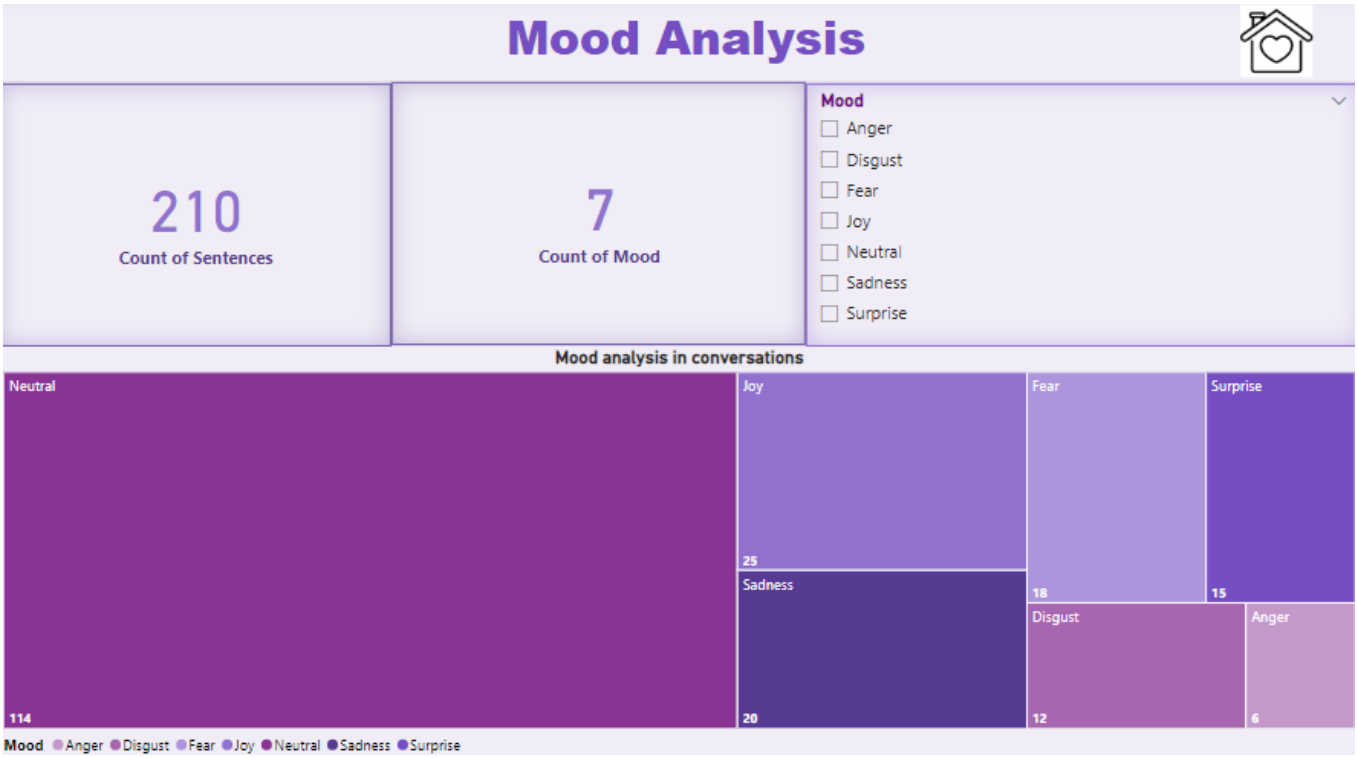

Figure 7: Mood analysis dashboard

In the Mood Analysis dashboard, further insights related to the Taxonomy/mood data are shown to the user. Users can view the number of conversations they had with Ariadne and can filter the moods to have an exact idea about how their mood was on a particular day, week, or month.

Tense Analysis Dashboard

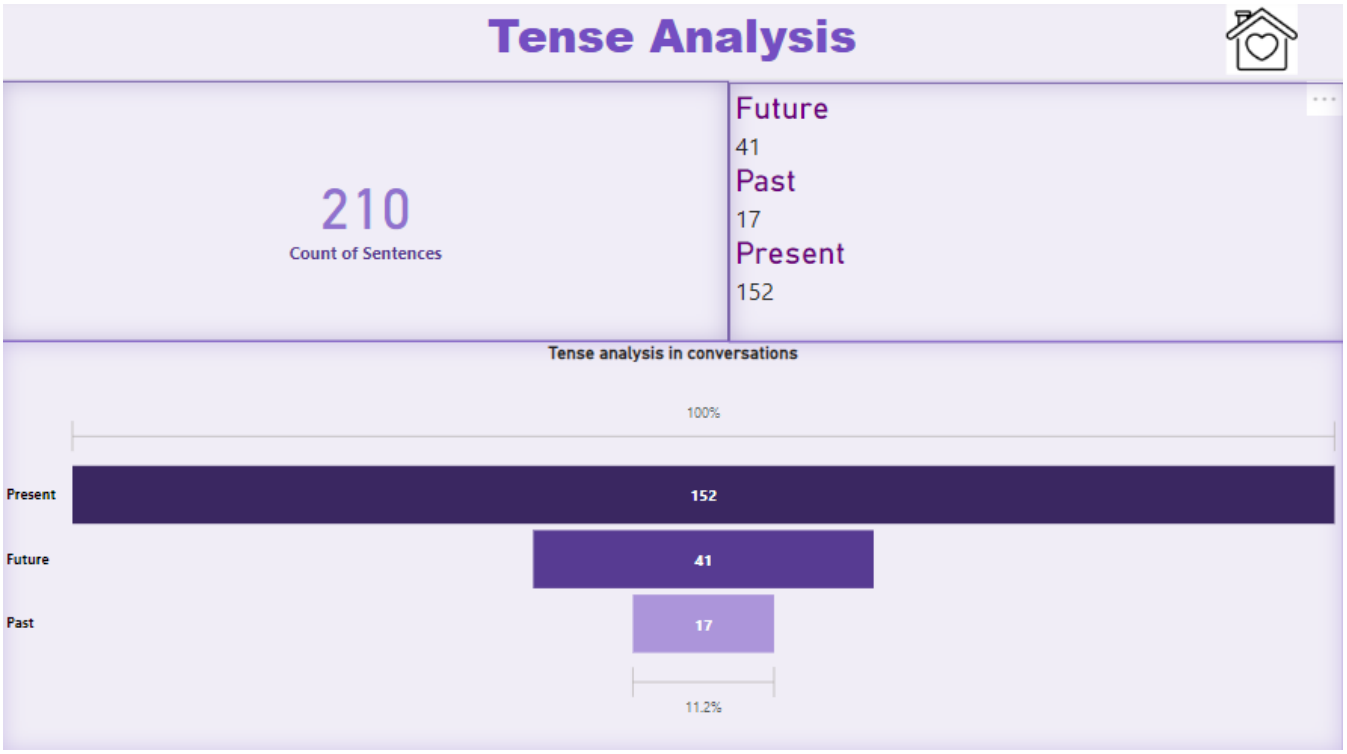

Figure 8: Tense analysis dashboard

The Tense Analysis dashboard shows further insights into the data regarding the Tense part of the conversation. Users can view whether they have discussed past, present, or future events with Ariadne.

Conclusion

With all the four models working and the prototype being built, the base required for the Ariadne project has been set up. Further developments might include integrating the dashboard with the code base, moving the codebase to the cloud, saving the data required in the database, and building a mobile app.

References

- https://georgiancollege.sharepoint.com/sites/ProPensity-LifeTracker/2021F
- https://www.sciencedirect.com/topics/computer-science/logistic-regression
- https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664
- https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/
- https://spacy.io/usage/spacy-101
- https://github.com/declare-lab/MELD
- https://paperswithcode.com/dataset/dailydialog
- https://docs.microsoft.com/en-us/power-bi/create-reports/service-dashboards-design-tips