Project Proposal

- **Title**: Customer support ticket classification
- **Group Size**: 3 people
- **Timeline**: 2 Weeks
- **Tentative Start Date: 23rd Aug 2021**
- **Tentative End Date: 4th / 5th Sept 2021 (Final review by this date).**

Expected Member Profile

Must-have

- Relevant programming background (e.g., Java/Python/R)
- Skilled in manipulating and processing data using libraries such as Scikit-learn, Pandas, and NumPy
- Knowledge of supervised and unsupervised machine learning algorithms including classification, clustering, and regression.
- Knowledge of the basics of NLP like TF-IDF, word embeddings (word 2 vec, glove)
- Can commit the required time (12-15 hours per week) and doesn't miss any sessions/sections.
- Good communication skills and a Team player.

Good to have

- **Math:** Basic math including algebra, statistics, and probability (permutation and combinations).

Recommended Preparation and Study Material

- The Python Workshop

  Sections 1, 2, 3, 4, 7, 9, 10
- The Data Visualization Workshop

  Sections 1, 2, 3, 4
- The Data Science workshop

  Sections 3 -12

- The NLP workshop

  Sections 1, 2, 3, 4, 8

Technology Stack

- Python 3x and Anaconda distribution
- Jupyter IDE (jupyter notebook or jupyter lab)
- Python stack: flask or streamlit
- Plotting/visualization libraries: Matplotlib and Seaborn
- Machine learning libraries: Sklearn
- NLP libraries: spaCy, gensim

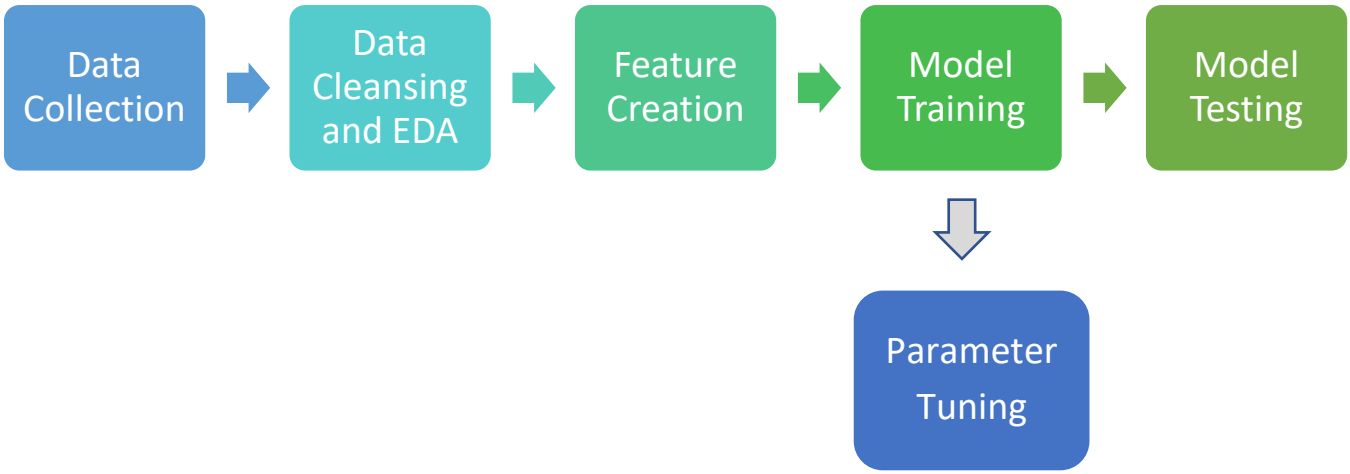
Datasets

The dataset is an outcome of a collaboration between Microsoft and Endava. You can find the dataset here. It has the following columns of interest:

- Title – a short description of the issue (type: string)
- Body – a detailed description of the issue (type: string)
- Ticket type – either L0 or L1 (type: int)
- Category – a ticket can belong to either one of the 12 categories (type: int)
- Subcategory – a ticket can belong to either one of the 58 sub-categories (type: int)
- Urgency – how quickly should this issue be resolved (type: int)
- Impact – what is the business impact of this issue (type: int)

In our case, the target column is ticket type.

Project Architecture



GitHub Repo Link

Will be shared once the draft is finalized.

Recommended System Setup

Hardware Requirements – Laptop/Desktop with at least 8GB RAM.

Software Requirements

- Operating system: Windows 10.
- Python 3.5+, Anaconda version (supports Python3.x) – available for Mac, Windows, and Linux (Debian/Ubuntu).
- Spreadsheet (Excel/LibreOffice).

Libraries – pandas, numpy, scikit-learn, streamlit, lightgbm, gensim, embedding-as-a-service, spacy, matplotlib or seaborn

Problem Statement

Company XYZ is well-known in the space of eCommerce and has been serving its customers happily for the past 30 years. They follow their motto, “Customer is King”, seriously, and ensure that customers' issues are resolved in time. However, it is a time taking task. For each ticket, a team of SMEs read the ticket to understand the issue and assigns it to the appropriate support level. This means, many times, for a simple issue, a customer might need to wait for hours, maybe days before it is resolved, which is not good. An unhappy customer could mean a loss of millions of dollars in revenue.

They have come to you with this problem, and see if you can help. What they want is a system that can automatically classify the tickets to either one of the levels (L0 or L1) based on its content.

User Stories

1. As part of management, I would be interested to see how does the new ticket classification system impact the bottom line of the business.
2. As a product owner, I would be interested to see the main areas of concern for our customer (can be a category, sub-category etc). This will help in

understanding ‘what customers want’.

3. As a marketing lead, I would be interested to see the customer feedback based on which I can lead product, pricing and marketing discussions.
4. As a data science lead, my focus will be to ensure that the customer tickets are classified correctly for a timely resolution. I will iteratively improve the underwriting algorithm by using suitable means and having checks of data quality.

Expected Solution

The solution will focus on developing a classification system that is capable of identifying and assigning the appropriate customer-level support to resolve the customer queries in time. A simple UI interface will be developed where a user query is given as input, and the outcome is the assigned level (L0 or L1) along with a probability score.

Project Timeline

Work Package

Well-documented analysis with basic exploration, and actionable insight for the business. The objective is to find the solution to a business problem using data by collaborating with other team members where an individual’s contribution to the overall solution is imperative and is evaluated throughout the course.

Milestone 1: Data understanding (EDA), Data Cleaning and Feature Creation
User stories: 2, 3

Week 1

Induction and project overview:

- Team induction and explanation of the project by the team mentor
- A detailed plan for the team to work on in the week, including key objectives and various steps.

Individual task:

- Setup the software environment required, and access the data set.
- Explore and review the features available in the data.

- Perform extensive data exploration using bar charts, pie charts, histograms, etc, and come up with your observations.
- Collaborate with others for a summary.
- Perform data cleaning. This includes but is not limited to removing URLs, numbers, UTF-BOM encodings, etc.
- Transform the text into vectors using TF-IDF

Group task:

- Group to discuss and consolidate the data summary.
- Organize all the findings and summarize them in the Jupyter notebook.

Milestone 2: Model building and Hyperparameter tuning

User stories: 1, 4

Week 2

Individual task:

- Build a baseline logistic regression model using the TF-IDF vectors.
- Evaluate the baseline model on the test set.
- Transform text to vectors using word-2-vec.
- Re-train the logistic regression model using the word vectors.
- Evaluate its performance on the test set.
- Experiment with other models like the random forest, extra tree classifier, light gbm.
- Tune their hyperparameters, and evaluate on the test set.
- Compare three models: baseline, word-2-vec, and tuned model, and summarize your findings.

Group task:

- Discuss among the group to train different models and variations of the model.
- Summarize various models in a single notebook and key features.
- Tune the model after discussion with others for different settings which can be split among others in the group.
- Summarize impact to the business with the model.
- Present the findings to the panel with business applications and solutions to

the users.
- List the caveats and other shortcomings in the note.

Project Outcome & Deliverables

- A simple UI developed using either Flask or Streamlit
- Presentation Deck explaining the project
- Jupyter Notebooks with source code and models
- Final Project Report

Assessment Criteria

- Data Summary and Exploratory Data Analysis (EDA):
    - Organization of data summary
    - Quality and reusability of the code
    - Presentation of the exploratory data analysis and highlights of the findings.
- Model performance:
    - Overall efficacy of the analysis and model.
- Business solution:
    - # of actionable insights for the business.