Praktikum - Machine Learning for Information Systems Students

Machine Learning Project Proposal

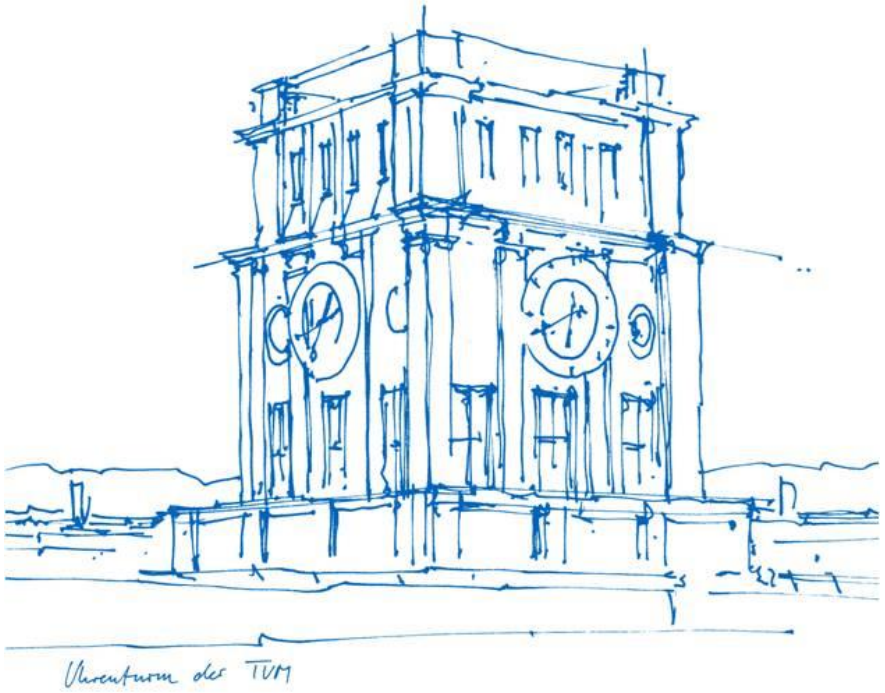
Group 3

Felix Böhm

Josep Quintana

Lucas Sanz Gorostiaga

Markus Linnenberg



Lehrstuhl I17 – Forschungsgruppe Wittges – SAP University Competence Center (UCC)

Technical University of Munich - SS2021

Table of Contents

Project Background

Status Quo

- The UCC uses a SAP ticket system to handle incoming customer requests
- Idea: Make this support system more efficient with the help of machine learning

Desired Outcome

- Improve user experience: Reduce the average ticket **resolution time**
- Improve efficiency: Reduce the number of **forwards per ticket** (could indicate wrong classification/ticket assignment)
- These metrics should be tracked in a live deployment to evaluate the ultimate business impact

Potential approaches

- Classify tickets into support level 1 or 2
- Classify category/sub-category
- Predict the correct support team for a ticket
- Clustering

Our suggested approach

Predict the correct **support team** for a given ticket. The prediction should be based on the ticket text as well as ticket metadata.

Why predict the support team and not alternatives?

- Tickets are ultimately assigned to a support team
- Categories and support level are in that sense only supportive information to help with assignment to the support team
- From a business perspective, correct support team assignment is key – Wrong ticket forwarding (or none) leads to higher resolution time and lower efficiency
- Trying to classify categories/subcategories may be challenging depending on the available data (number of categories/sub-categories)

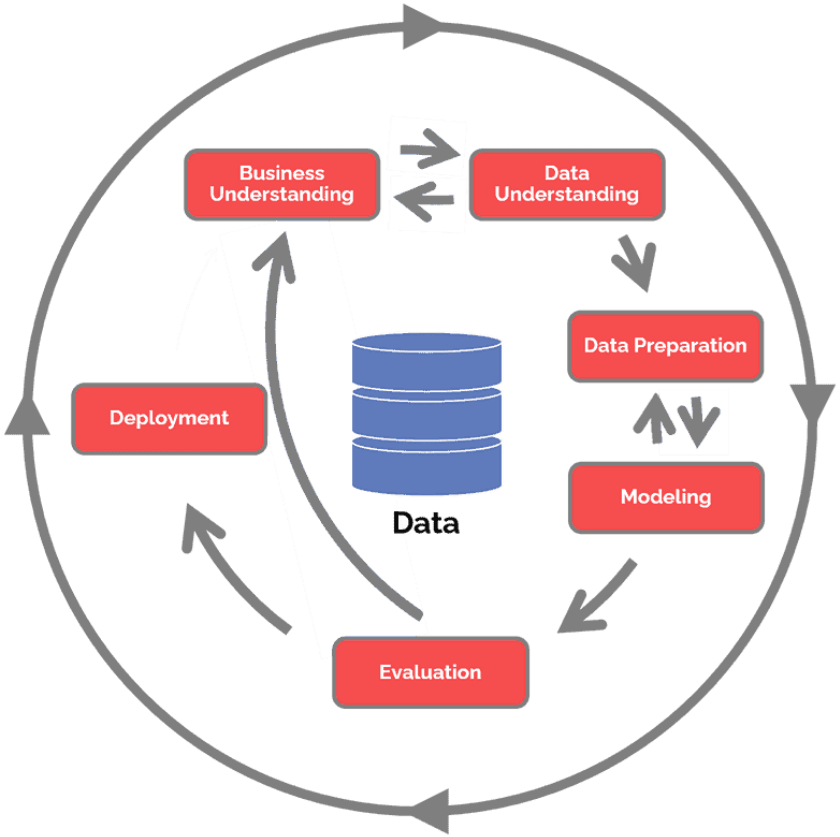
Tasks and task distribution

Task	Subtask	Responsible
Data Preparation	1. Combine ticket text with metadata (status log, ticket overview) 2. Extract relevant text used to make predictions	LSG, FB, JQ, ML
Setup and Organization	Git, project management, weekly *jour fixe*	LSG, FB, JQ, ML
Feature Engineering	1. Keyword extraction libraries 2. Term Frequency – Inverse Document Frequency 3. Manually define keywords	TBD
Exploratory Data Analysis	Detect anomalies in ticket text (empty or incomplete)	
Data Pre-Processing and Building the Pipeline	1. Data normalization 2. Pipeline design	
Model Design and Evaluation		
Create Dashboard in Streamlit / Prepare demo		
Project Write-Up / Potential future work		

LSG: Lucas FB: Felix JQ: Josep ML: Markus

Tasks

Follow the CRISP-DM model



Machine Learning Approach

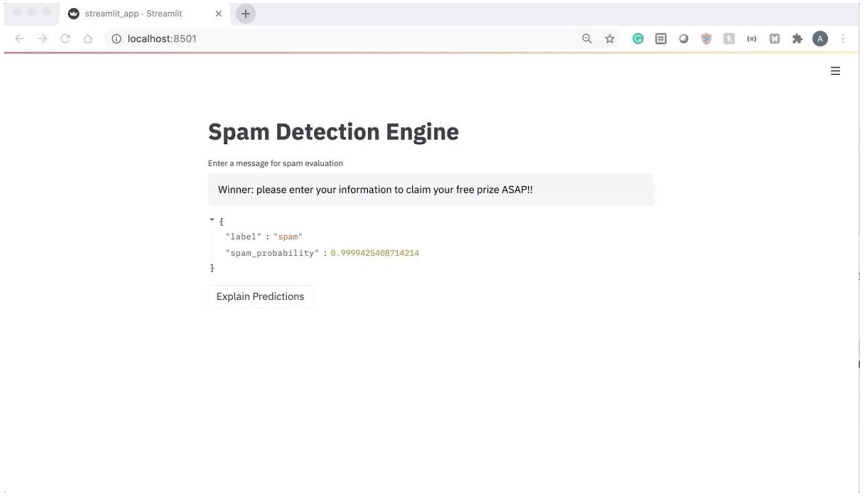
1. This is a **supervised learning problem** as we have the correct (human assigned) support team as a column in the “tickets.csv” file
2. Choice of **classification algo**: Naive Bayes, Random Forest, Decision tree, KNN, AdaBoost
3. **Packages/Libraries:**
    1. Sklearn: Machine Learning
    2. Pandas: Data Analysis
    3. Seaborn: Visualisation (e.g. confusion matrices)
    4. Nltk: Natural language toolkit
    5. Regex: Regular expressions
    6. Spacy: Natural Language
    7. Streamlit: Simple WebApp
4. **Parameters/techniques:** Stemming
5. Important would be that the model can take in very sparse input and do a **classification onto relatively few classes**

Evaluation

1. Precision / Recall, f1, accuracy
2. For evaluation when the system is deployed: mean number of forwards per tickets
3. What accuracy do we aim for: 100% on the test set of course 😛
   (or at least 75%)

Demo/Prototype

1. Streamlit Dashboard[1)]



Sample Streamlit WebApp for Spam Classification

2. Live Demo with Jupyter Notebook

1) Source: https://towardsdatascience.com/how-you-can-quickly-build-ml-web-apps-with-streamlit-62f423503305

Thank you for your time