Natural Language Processing

Classification Models in Technical Support

TABLE OF CONTENTS

TABLE OF CONTENTS

Project Goal and Background

- The focus of this project is to use Natural Language Processing techniques in developing machine learning models.
- The setting is in Technical Support for an automotive company.
- Cases represent all business functions' applications (e.g. Supply Chain; Finance)
- The goal is to identify best-fit models for hypothetical deployment.

TABLE OF CONTENTS

List of Tools

- Data download via GUI as CSV files.
- **Excel** for initial cleanup and line-by-line coding.
- **Anaconda** + Jupyter Notebook + Python + **NLTK Library** for Natural Language Processing operations
- **Knime** for no-code modeling.
- Jupyter Notebook + **TensorFlow** library for code-based models

TABLE OF CONTENTS

Data Wrangling

USER-CREATED TICKETS

Advantages
- (mostly) Consistent Titles.
- Automation can add keywords and text.

Disadvantages
- Users can leave description blank
- Users may not know the actual product; or use incorrect form.

HELP DESK-CREATED TICKETS

Advantages
- Knowledge advantage in determining issues.

Disadvantages
- Agents can fill multiple issues under only one ticket.
- Inconsistent affected service.
- On repetitive tasks, agents can leave non-descriptive information.

Data Clean-Up

- Removed non-English entries
- Removed entries with non-descriptive information.
- Combined similar terms.
  - E.g. “Microsoft Office” and “MS Office”
- Combined free-text forms as one.
- Transformed all text to lowercase
- Corrected Affected Service

Data : New Features

FEATURES IN DATA

Free-Form Fields
- Title
- Description
- Resolution

Other
- Affected Service
- Category Issue
- Ticket ID

FEATURES CREATED

Function
- Class under which an affected service falls.

Global Category
- Global Function Category.
- Functions fall under one Global Category

Functions

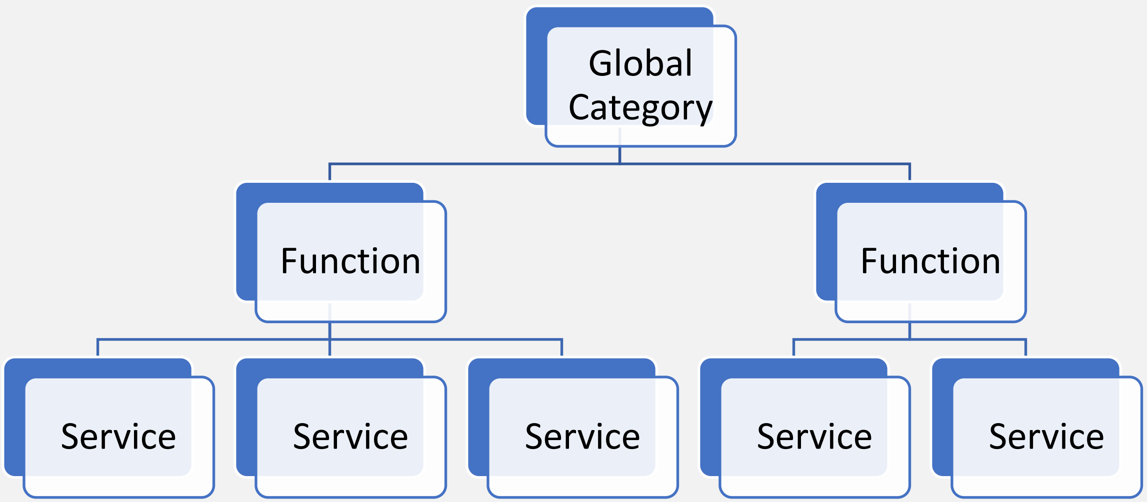
- **Function** categories are based off business function
- Business structure influenced how the functions were grouped**.**

- When a single class or product accounted for a high percentage within a Function, it was broken off into its own category.
    - e.g. Microsoft Office was a large portion of non-engineering software and was coded as its own Function.

Global Categories

- Grouped **Business Functions** into their corresponding general **categories.**
- For example:
  - All hardware-related issues are grouped into **Hardware.**
  - Applications relating to product development, design, production and quality control fall under **Product Life Cycle**.

Data Hierarchy



- Individual services fall under a Function
- In turn all Functions fall under a Global Category

Data Summary

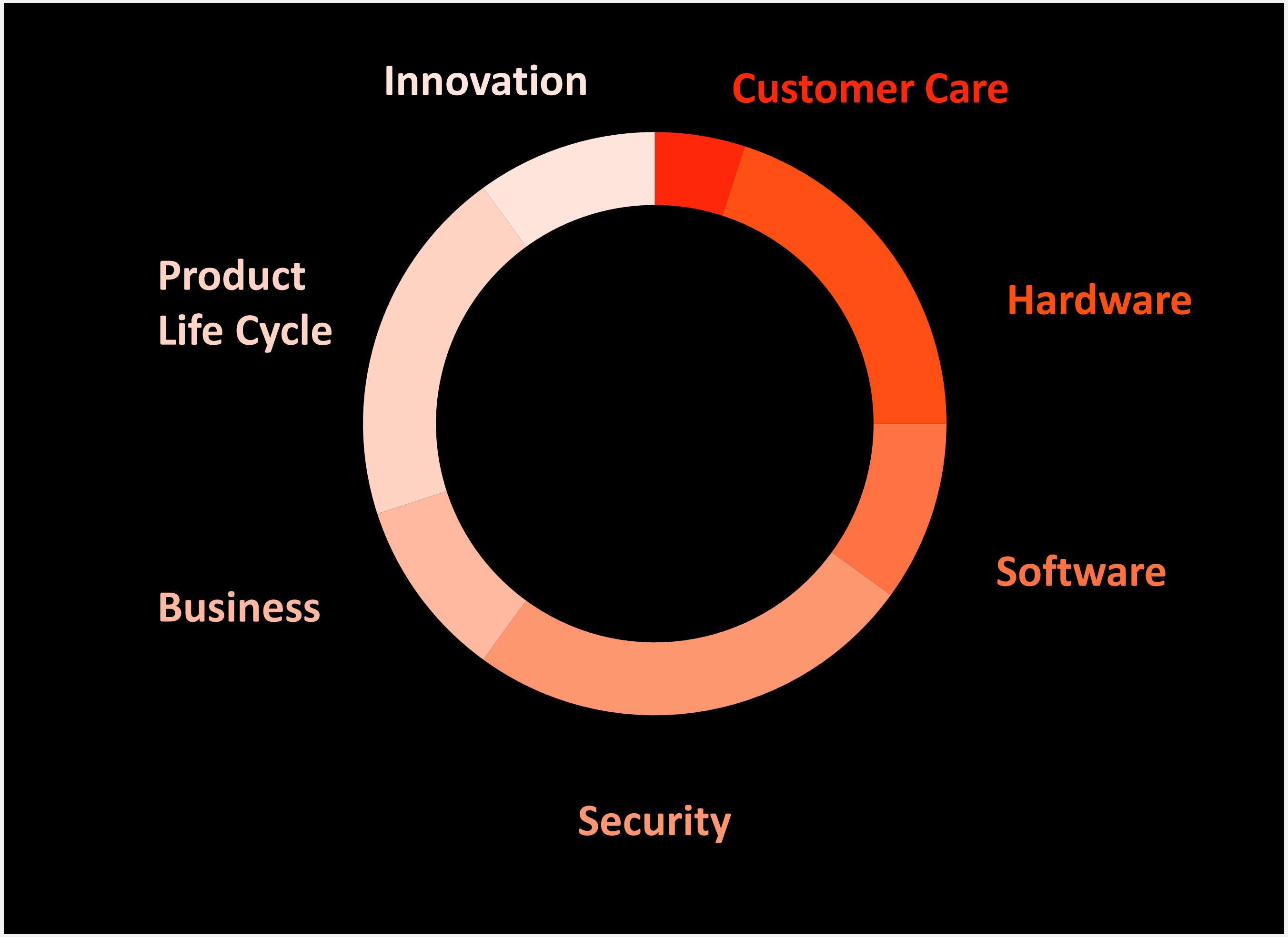
Data Size

- Data set consisted of 28,173 entries

Data Properties:

- 966 Affected Services
- 19 Function Categories
- 7 Global Categories

Global Categories



Distribution of Data by Function

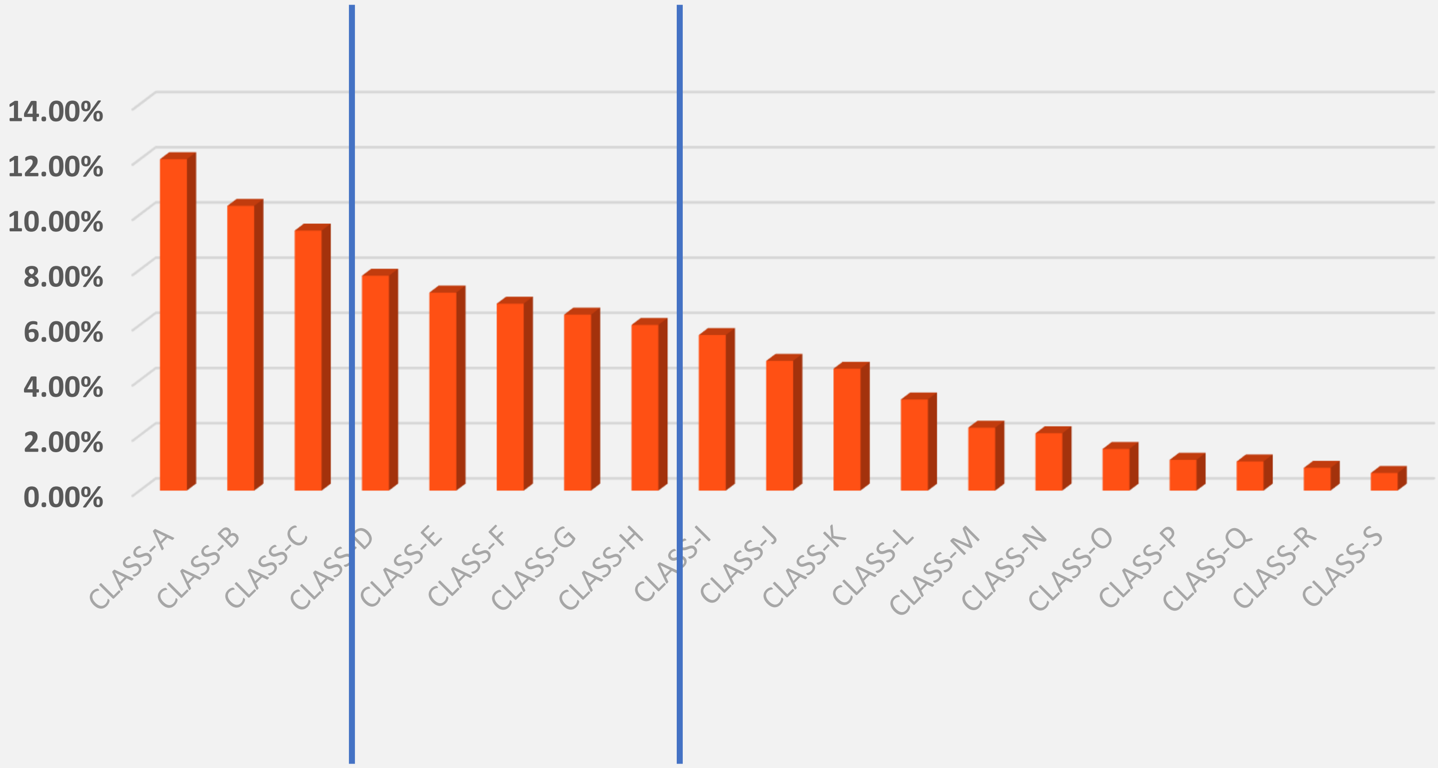


TABLE OF CONTENTS

NLP Operations

- NLP Operations were calculated on the '**description**' feature in the dataset.

STOP WORDS

Stop Words are words that add no significant value to the meaning of the sentence.

Stop words are generally articles, transitive words (e.g. "for", "in", "at")

First step in data preparation.

STEMMING + LEMMATIZATION

Both of these are techniques that reduce words to their root or base unit.

Lemmatization uses sentence context.

Only **Stemming** was used in this project.

N-GRAMS

N-grams combine words into tuples of *n* length.

I explored bigrams (2-grams) and trigrams (3-grams)

No significant changes or improvements noticed during exploration phage.
Did not use n-grams.

Keywords

- There are two main sets of Keywords identified:
  - **Global:** Most common words in the entire dataset.
  - **Function**:
    - Most common words when looking at each individual **Function**.
- A computed aggregate bag was created by assigning most frequent words from both keyword sets.
  - 50% of words came from Global
  - 50% of words came from Function

Feature Encoding

- For each word, a single feature is calculated to indicate its presence in the description.
  - (1 = yes, 0 = no)
- The models developed did not receive the description text as input.

- Models took as input one of five **Keywords Groups:**
  - **Global Keywords**. Up to 457 features.
  - **Function Keywords.** Up to 203.
  - **Combination Keywords.** Up to 190
  - Two **Component Keyword** groups.

Feature Engineering: Components

- In an effort to simplify the complexity of models, I sought to combine features into groups, or **components**.
- These calculations were only performed on **Global Keywords.**

3-KEYWORD COMPONENT

Combines every three common words into one component.

152 Components (456 keywords)

5-KEYWORD COMPONENT

Combines every five keywords into one component.

91 Components (455 keywords)

ROW ID	W1	W2	W3	W4	W5	W6	W7	W8	W9
1001	1	0	1	0	1	0	0	1	0
1002	0	0	0	0	1	0	1	1	1
1003	0	1	1	0	0	0	1	1	1

ROW ID	C1	C2	C3
1001	2	1	1
1002	0	1	3
1003	2	0	3

TABLE OF CONTENTS

Model Development

- I tested four types of machine learning algorithms:
    - K-Nearest Means
    - Decision Trees
    - Gradient Boosted Trees
    - Random Forests. (TensorFlow and Knime)

Model-Building Components

Input
- **Keywords**: Global, Function, Combination
- **Components:** 3-Keyword, 5-Keyword

Word Bank
- Dependent on keywords.
- Up to 457

ML Algorithm
- Decision Tree
- Gradient Boosted Tree
- Random Forest
- K-Nearest Neighbor

Target Variable
- Function
- Global Category

592 Trained Models

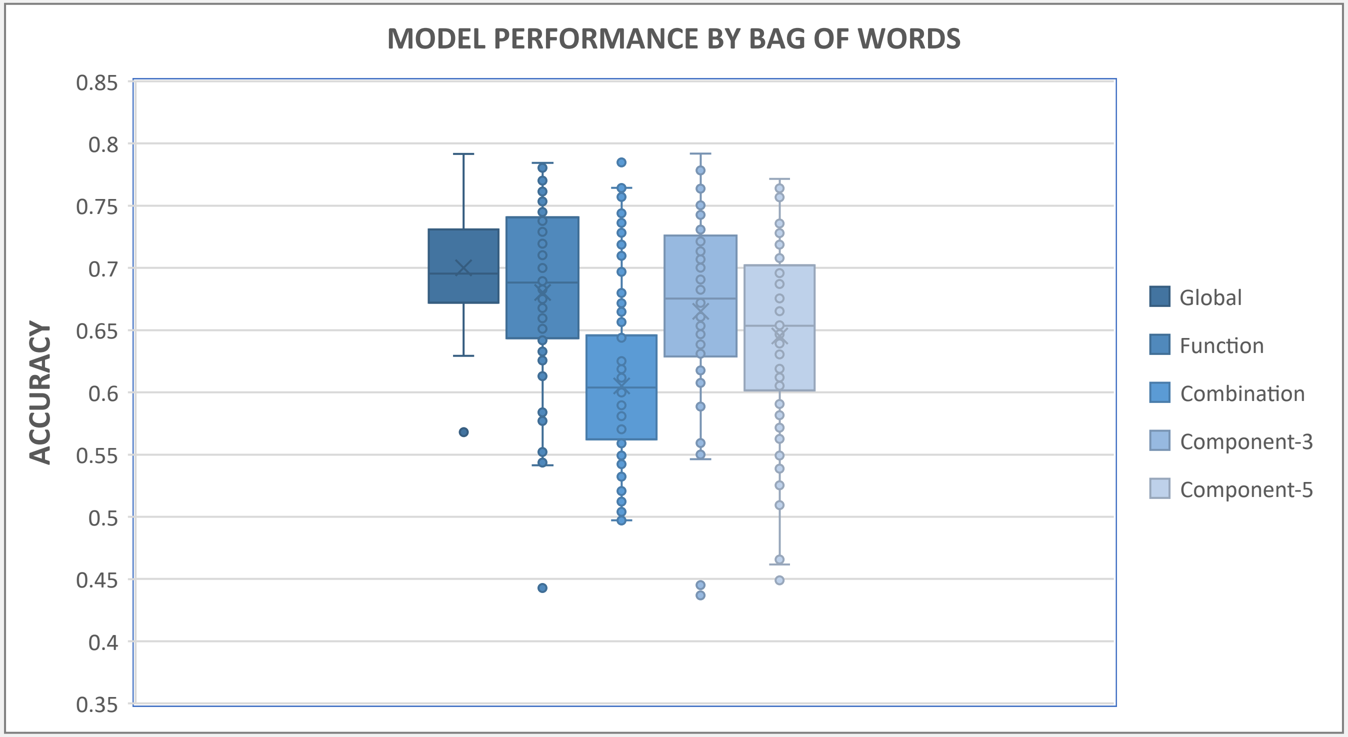
TABLE OF CONTENTS

Performance and Results

- **Evaluation**
  - Performance was analyzed as accuracy of model.
  - A model that works well on overall performance is selected because real-world scenario is interested in handling volume.
  - I decided that there is no need to focus on any individual target or try to balance performance across targets.

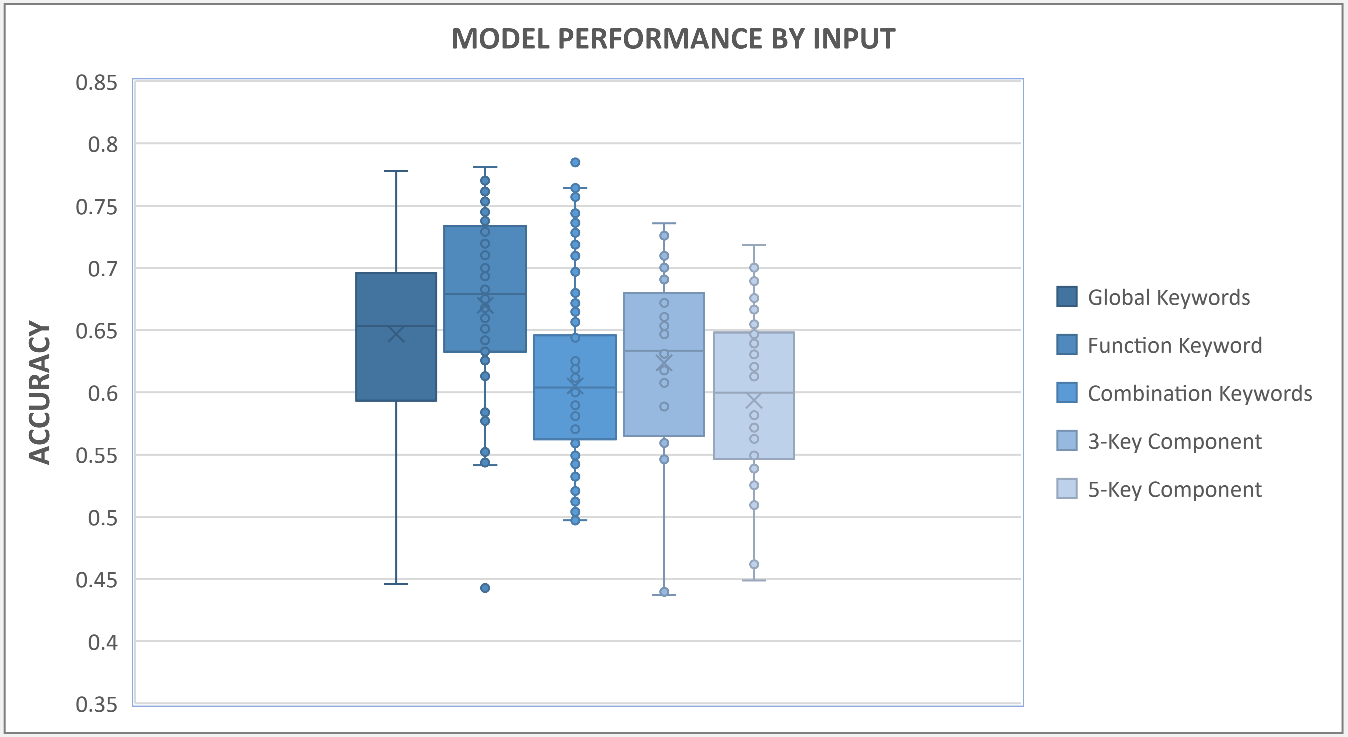
Performance and Results

- **Input (Keyword Type)**
  - On average **Global Keywords** outperform others.
  - Combination Keywords weakest performer.



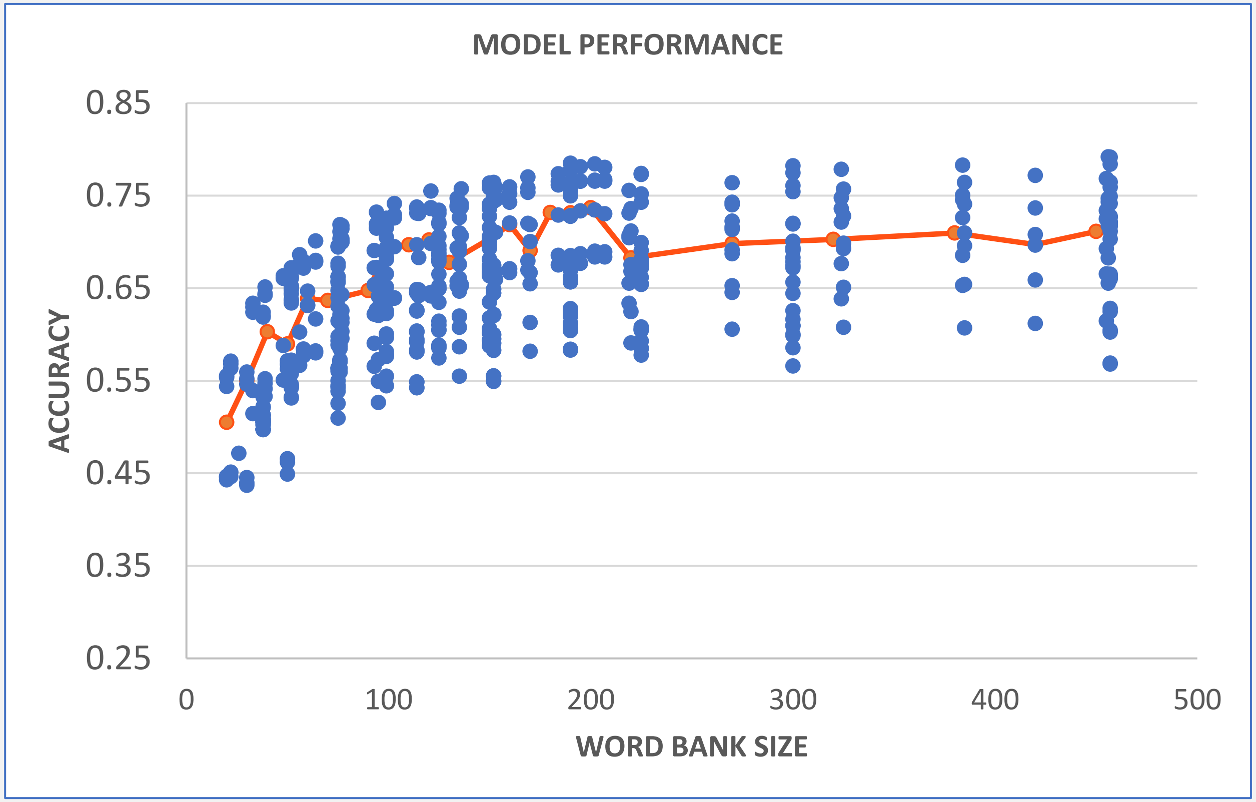
Performance and Results

- ***But…*** Global Keywords models can use up to 457 words.
- Combination Keyword models up to 190.
- Limiting performance analysis to only models with 200 or fewer words:



Performance and Results

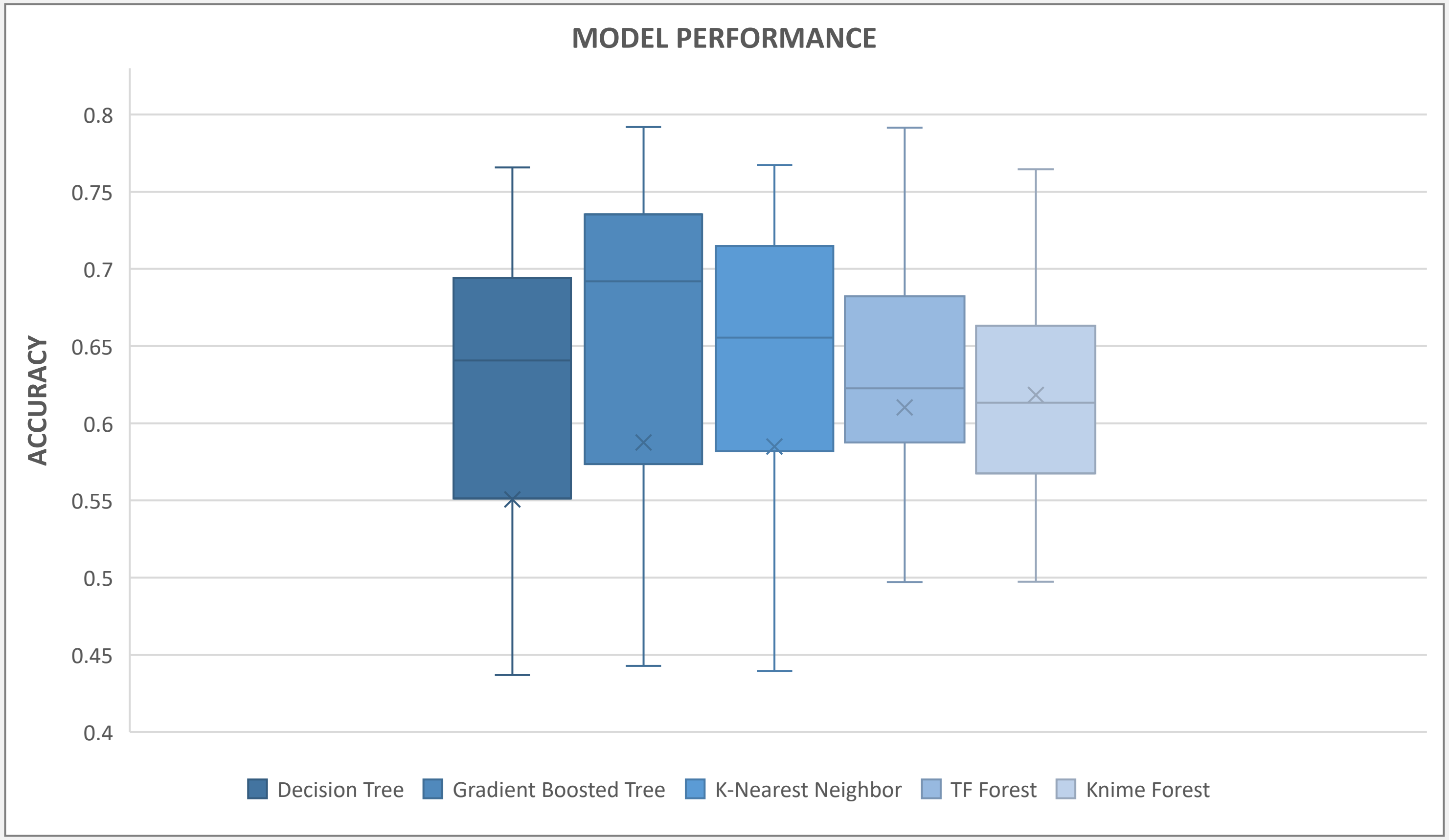
- **Word Bank (Number of keywords)**
  - As the number of keywords increases, so does performance.
  - However, performance flattens around 200 keywords.



Orange line shows average performance for all models using word bank of that size.

Performance and Results

- **Machine Learning Algorithm**
  - Gradient Boosted Trees are best performers overall.
  - Knime Random Forests are weaker.



Performance – Top Models

TARGET	ALGORITHM	INPUT	WORD BANK	ACCURACY
Global Category	Gradient Boosted Tree	Component-3	456	0.7919
	Random Forests. Depth: 50. Trees: 50	Global Keywords	457	0.7915
	Gradient Boosted Tree	Combination	190	0.7848
	Gradient Boosted Tree	Function Keywords	202	0.7843
	Random Forests. Depth: 50. Trees: 25	Global Keywords	457	0.7837
Function	Gradient Boosted Tree	Component-3	384	0.7454
	Gradient Boosted Tree	Component-3	456	0.7432
	Gradient Boosted Tree	Component-3	324	0.7357
	Gradient Boosted Tree	Function Keyword	202	0.7344
	Gradient Boosted Tree	Function Keyword	195	0.7333

Natural Language Processing

Classification Models in Technical Support