

**Group  
09**

# **CODE QUALITY CHECK**

## **BY SEMANTIC SEARCH**

# OUR TEAM



**E-18-276 Jayathu**

✉ [e18276@eng.pdn.ac.lk](mailto:e18276@eng.pdn.ac.lk)



**E-18-382 Denuwan**

✉ [e18382@eng.pdn.ac.lk](mailto:e18382@eng.pdn.ac.lk)



**E-18-412 Sanjaya**

✉ [e18412@eng.pdn.ac.lk](mailto:e18412@eng.pdn.ac.lk)

# PRODUCT OWNER

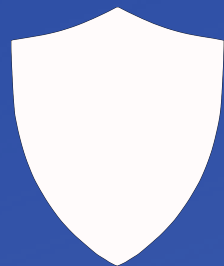


**Dr. Shamane Siriwardhana**  
(Phd.)  
University of Auckland, New  
Zealand

# WHY DO WE HAVE TO CHECK OUR CODE?



Before using a code in some system and deployment, first we need to check and review our code.



Mainly to reduce problems in collaboration and teamwork, reliability, ease of maintenance, readability, extendability etc.



# PROBLEMS WITH CODE CHECKING

Traditional way of checking – manual review



**Time Consuming**



**Difficult and  
subjective**



**Prone to human  
errors**

# PROBLEMS WITH EXISTING CHECKERS

## Limited Contextual Understanding

Since most systems rely on rule-based or static analysis, they lack a deep understanding of context and semantics

## Difficulty in handling code complexity

Difficulty in checking complex code structures, dependencies and interactions

## Limited detection of design issues and language dependent

Only focus on issues like syntax errors, naming conventions etc. Overlooks maintainability, extensibility and overall software quality.

## Lack of Human-Like Understanding

Works in a more robotic way, controlled by some pre-programmed rules



# MOTIVATION

Developing a system to check code quality with semantic search using machine learning



Avoiding language dependency and having a contextual understanding



Avoiding limitations in manual code reviewing



Ensuring consistency, readability, maintainability and overall quality



Scaling and efficiency



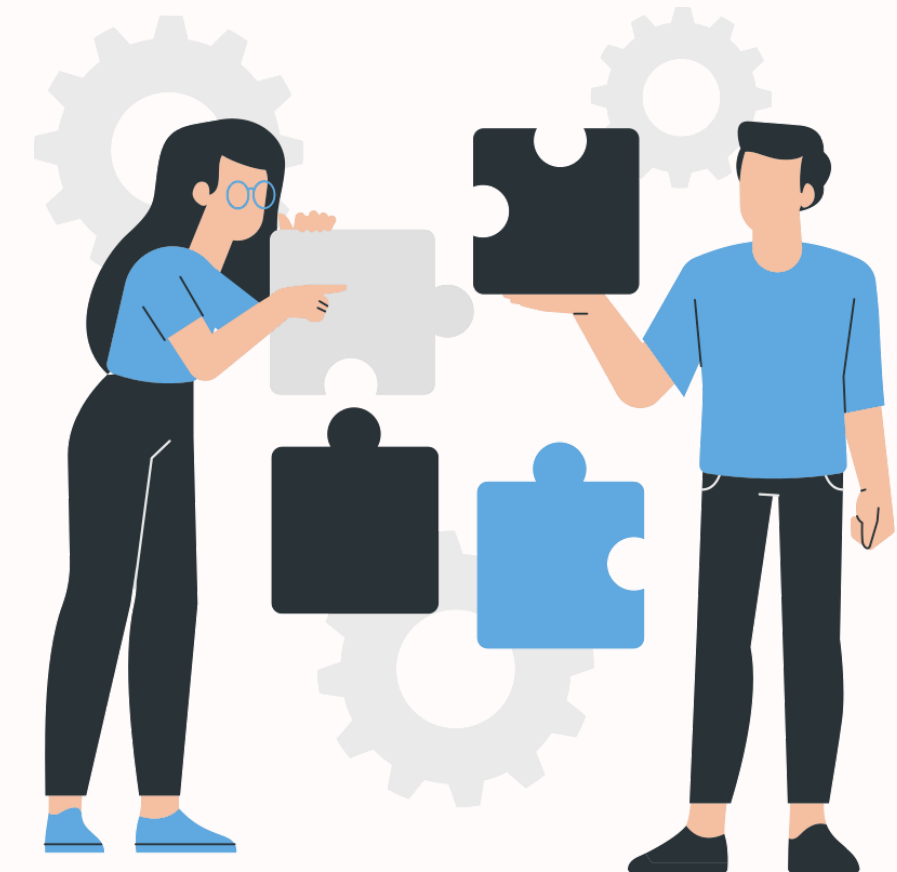
# Proposed Solution

To develop a deep neural network that can evaluate the quality of code and classify input code as good or bad using semantic analysis

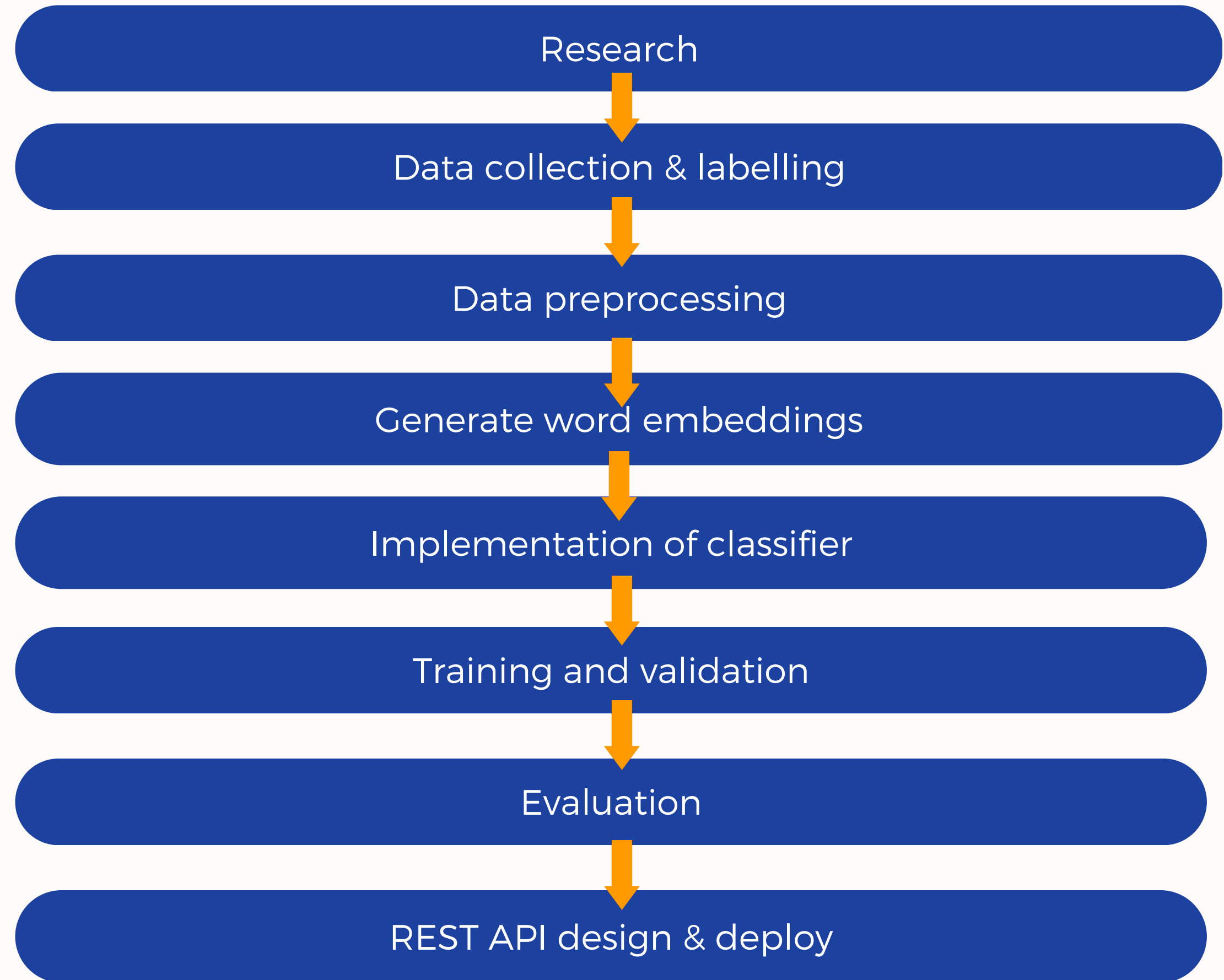
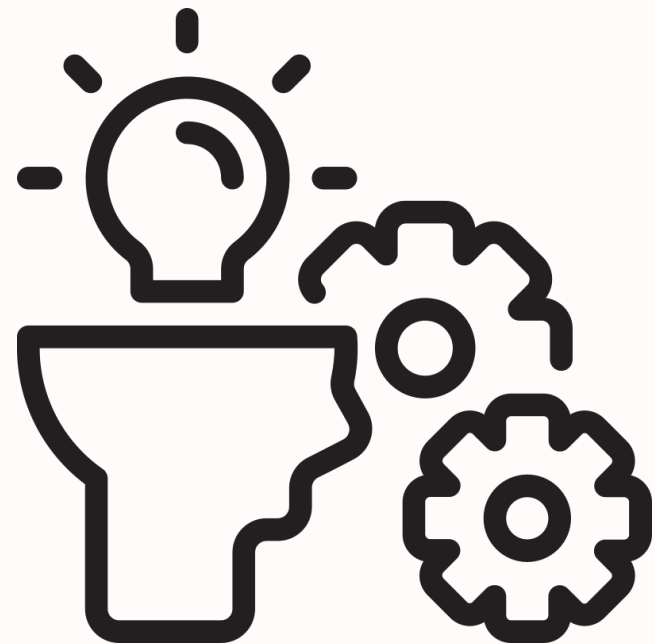


# HOW DOES THIS SOLVE THE PROBLEM AT HAND?

- Automated Code Analysis
- Semantic analysis
- Different programming languages
- Processing of large code base
- Code suggestions based on semantic search



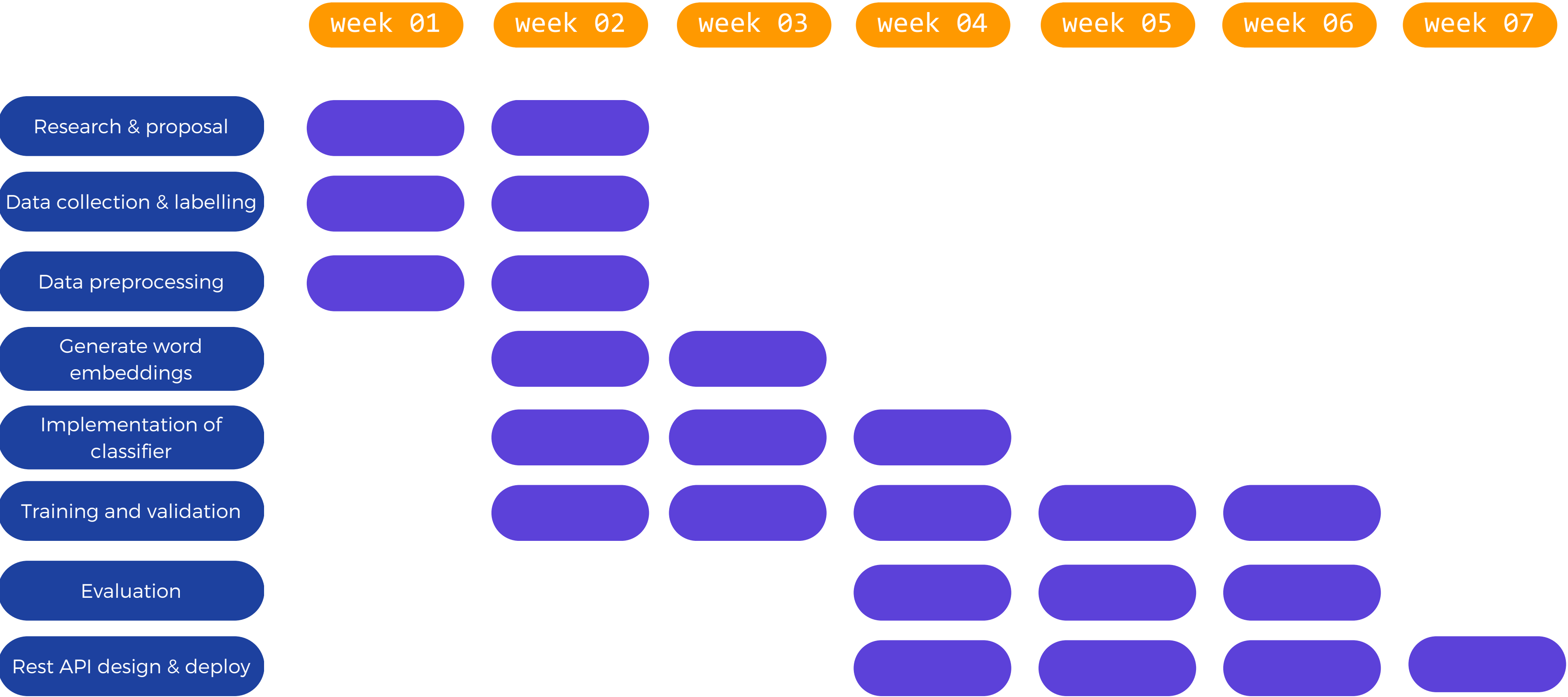
# PLAN OF WORK



# HIGH LEVEL SYSTEM DIAGRAM



# PROJECT TIME LINE



# TEAM STRENGTHS



- Knowledge in version control
- Basic machine learning knowledge
- Communication and collaboration
- Experience in AJILE methodology
- Rest API design



# EXTENDABILITY

This project can be extended to following areas



Visual Code plugin



Web application



Suggestions based on semantic similarities



