maltepe university
istanbul
www.maltepe.edu.tr

Faculty of Engineering and Natural Sciences

Turkish Joke Generator

Prepared by

Furkan Aksoy ~ 210706029

Emre Sarı ~ 220706304

Mehmet Güzel ~ 210706030

Tamay Yazgan ~ 210706022

Ömer Faruk Özer ~ 210706028

Yaren Yıldız ~ 200706040

Table Of Contents

Risk Document

Turkish Joke Generator

Project Duration: February 27, 2025 – April 24, 2025

Team Members: Furkan Aksoy

- Emre Sarı
- Tamay Yazgan
- Ömer Faruk Özer (Scrum Master)
- Mehmet Güzel
- Yaren Yıldız

  Advisor: Prof. Ensar Gül

1. Introduction

The purpose of this Risk Document is to identify, assess, and establish mitigation strategies for potential risks throughout the lifecycle of the Turkish Joke Generator project. This document serves as a living record that will be updated throughout the project as new risks emerge and existing risks are addressed. Effective risk management is crucial to ensure the project meets its technical, operational, and timeline objectives.

2. Risk Identification

Our team has identified the following potential risks that may impact the project:

- **Inadequate Model Accuracy:** The GPT-2 model may not produce culturally coherent or humorous outputs.
- **Data Quality and Cleaning Issues:** Inaccurate or poorly processed data could lead to suboptimal model training.

- **Integration Challenges:** Difficulties in integrating the trained model with the Flask-based web application might result in performance issues.
- **Scheduling and Resource Delays:** Delays in task completion or resource availability could jeopardize the project timeline.
- **Communication Breakdowns:** Ineffective communication among team members may lead to misunderstandings and workflow disruptions.
- **Technological Infrastructure Risks:** Issues related to Google Colab, PyCharm, or hardware limitations (e.g., GPU availability) could affect model training and application performance.

3. Risk Analysis and Evaluation

Each identified risk is analyzed based on its potential impact on the project and its probability of occurrence. The risk register below summarizes these aspects:

Risk	Impact	Probability	Mitigation Strategy	Owner
Inadequate Model Accuracy	Reduced user satisfaction and failure to meet project objectives.	Medium	Improve data quality, perform extensive hyperparameter tuning, and consider alternative generation methods.	Data & Model Team (Mehmet, Yaren)

Risk	Impact	Probability	Mitigation Strategy	Owner
Data Quality and Cleaning Issues	Faulty outputs due to improper training data.	Medium	Implement rigorous data pre-processing, conduct manual reviews, and utilize automated cleaning scripts.	Furkan, Emre, Tamay
Integration Challenges (Model to Web)	Delays or performance bottlenecks in the web application.	High	Develop early prototypes, maintain continuous integration, and adhere strictly to API design standards.	Furkan, Emre, Tamay
Scheduling and Resource Delays	Project timeline overruns and missed deadlines.	High	Use agile sprint planning, monitor progress closely using Trello, and hold regular stand-up meetings.	Ömer Faruk & Entire Team
Communication Breakdowns	Misalignment on project tasks and delays in issue resolution.	Medium	Establish clear communication channels via WhatsApp and scheduled	Ömer Faruk & Entire Team

Risk	Impact	Probability	Mitigation Strategy	Owner
			meetings, and document all decisions in Trello.	
Technological Infrastructure Risks	Interruptions in model training or deployment due to tool limitations.	Low	Plan backup strategies, monitor resource usage on Google Colab, and keep alternative development tools ready.	Entire Team

4. Risk Mitigation Strategies

For each risk, the team has developed specific mitigation strategies:

- **Inadequate Model Accuracy:**
  - Enhance data quality by rechecking the dataset.
  - Fine-tune hyperparameters and run multiple training iterations.
  - Explore fallback options if the model underperforms.
- **Data Quality and Cleaning Issues:**
  - Implement both automated and manual data validation processes.
  - Schedule periodic reviews to ensure data consistency.
- **Integration Challenges:**
  - Build and test a prototype early in the project.
  - Maintain a continuous integration process to catch issues early.

    - Use modular development practices for easier debugging.
- **Scheduling and Resource Delays:**
    - Utilize agile methodologies with clear sprint planning.
    - Monitor progress daily through stand-up meetings.
    - Have contingency plans and buffer times built into the schedule.
- **Communication Breakdowns:**
    - Use centralized communication tools (WhatsApp, Trello) to keep all team members aligned.
    - Document all key decisions and action items.
    - Hold regular review meetings to address any emerging issues.
- **Technological Infrastructure Risks:**
    - Monitor usage on Google Colab and prepare alternative resources if needed.
    - Stay updated with tool documentation and community forums.
    - Ensure team members are ready to switch to backup tools if necessary.

5. Risk Monitoring and Reporting

Risk monitoring is an ongoing process throughout the project lifecycle. The team will:

- **Conduct Regular Reviews:** Evaluate existing risks and identify new risks during weekly meetings.
- **Update the Risk Register:** Revise the risk register to reflect the current status and any changes in risk severity.

- **Report to Stakeholders:** Provide regular updates on risk status during sprint reviews and project demos.
- **Implement a Feedback Loop:** Use feedback from testing and user input to adjust risk management strategies as necessary.

6. Conclusion

This Risk Document provides a clear framework for identifying, analyzing, and mitigating risks associated with the Turkish Joke Generator project. By proactively addressing these risks, the team aims to ensure a smooth development process and successful project delivery within the established timeline. Continuous monitoring and adaptive strategies will help minimize the impact of potential disruptions, enabling the team to focus on achieving technical and operational excellence.

7. References

- **Schwalbe, K. (2015).** *Information Technology Project Management.* Cengage Learning.
  Provides foundational concepts on project management practices and risk mitigation strategies.
- **Schwaber, K., & Sutherland, J. (2020).** *The Scrum Guide.*
  Offers comprehensive guidelines on implementing Scrum methodology effectively for agile project management.
- **Flask Official Documentation:**
  Flask Documentation
  Reference for developing robust web applications using Flask.
- **GitHub Guides:**
  GitHub Guides

Best practices for version control and collaborative software development.

- **Google Colab Documentation:**
  Google Colab
  Supports cloud-based model training and collaborative coding practices.