XNLP
Explainable NLP App
SEBA Lab

Mürüvet Doganay
& Didem Yaniktepe
& Julian Heiss
& Fabian Leibinger

February 19, 2024

Contents

1 Introduction

XNLP empowers users to delve deeper into the world of NLP text classification models. Imagine it as a tool that unlocks the "black box" often associated with these models, offering exploration and understanding through cutting-edge explainability methods.

What does XNLP offer?

- **Interactive exploration:** Dive into the inner workings of different NLP models to see how they process and classify text.
- **Insights into performance:** Get a research-based perspective on the strengths and weaknesses of various explanation methods through user feedback.
- **Insights into performance:** Get a research-based perspective on the strengths and weaknesses of various explanation methods through user feedback.
- **Model debugging:** Identify and pinpoint potential issues within your NLP models using XNLP's explainability methods, aiding in their optimization and performance improvement.

Key features:

- **State-of-the-art explainability methods:** Access a collection of cutting-edge techniques like **SHAP, LIME, and Counterfactual Explanations** to unravel the logic behind text classifications.
- **Interactive visualizations:** Gain intuitive understanding through clear and engaging visual representations of data and model behavior.
- **Data-driven feedback loop:** Contribute to the continuous improvement of XNLP by providing your feedback on the provided explanations.

Why is XNLP valuable?

- **Demystifies complex models:** Makes advanced NLP technology accessible to a wider audience by providing clear explanations.
- **Empowers informed decision-making:** Helps users understand the rationale behind model outputs and make better-informed choices.
- **Drives research and development:** User feedback contributes to advancing explainability methods in NLP.

2 Identified Personas, User Roles, Expectations

We have two types of Users. The NLP Analyst and the NLP Explorer. NLP Explorers are users who have a limited understanding of NLP. They may be business users, consumers, or other stakeholders who use NLP-powered applications but do not need to understand the underlying technology.

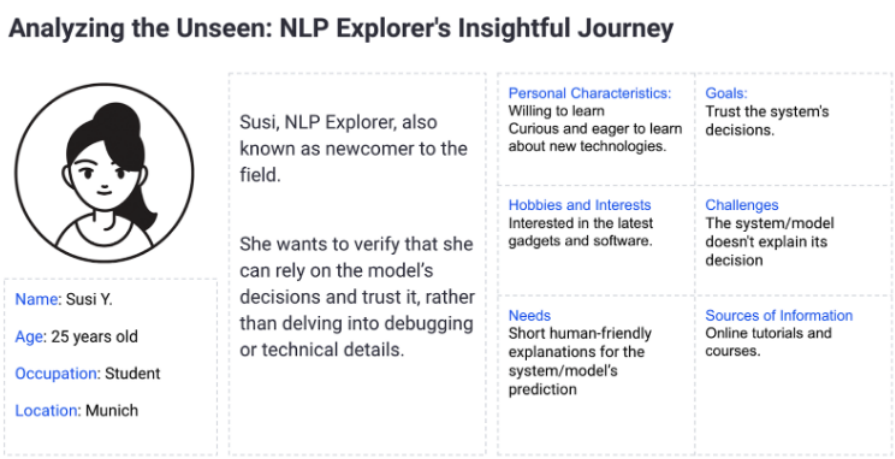

Figure 1: NLP Explorer

NLP Analysts are users who have a deep understanding of NLP and its applications. They may be researchers, developers, or practitioners

who use NLP tools and techniques regularly.

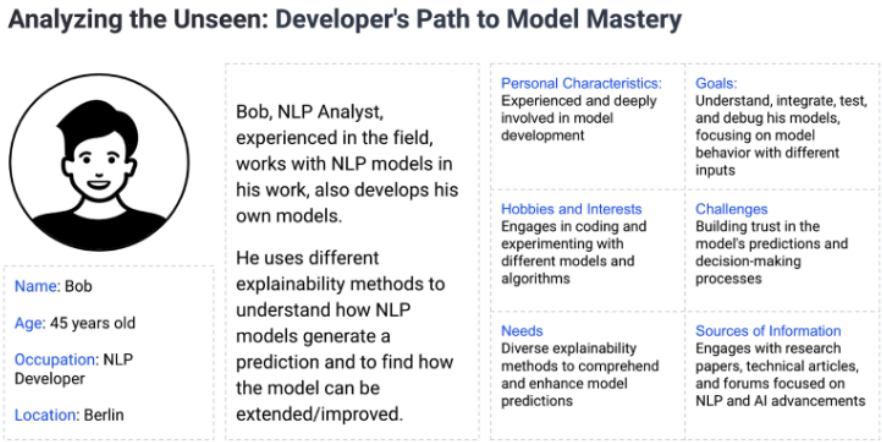

Figure 2: NLP Explorer

From a research perspective, collecting the user's purpose for using the application is really interesting. According to this, explanation methods can be rated from different usage standpoints.
We've classified the "usage expectations" into 4 classes:

- **Build trust:** understand how a text classification model comes to its prediction
- **Compliance:** test a specific text classification model in a given context
- **Debugging:** use explanation methods to test and debug text classification models
- **Research:** test models and explanation methods

3 User Guide

This gives an overview of the application and should help the user to understand how to use the web page.

3.1 Landing Page

Figure 3: Landing Page

At the beginning, the user is on our landing page. Here, the user gets short information about the tool.

3.2 Role Selection

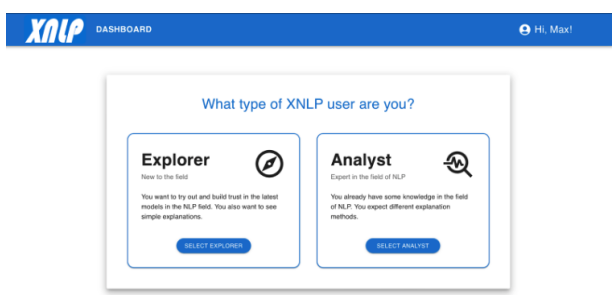

Figure 4: Role Selection Page

After the registration, the user is asked about his experience. Depending on that, the user has to choose Explorer or Analyst.

3.3 User Expectations

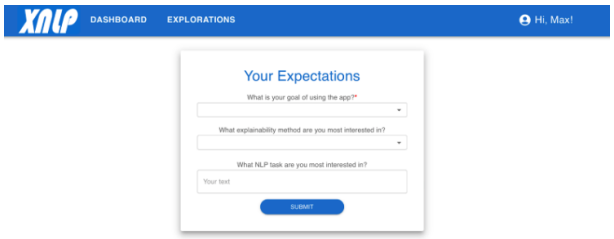

Figure 5: User Expectations Page

Afterward, the user is asked to enter expectations with the dropdown menu.

3.4 User Login

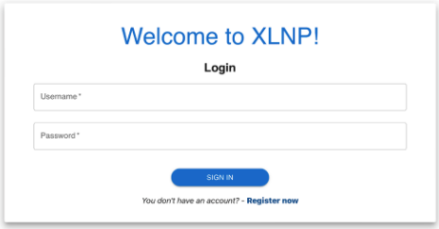

Figure 6: User Login Page

The User can log in to the user's account if user has already created one.

3.5 Dashboard

3.5.1 Empty Dashboard

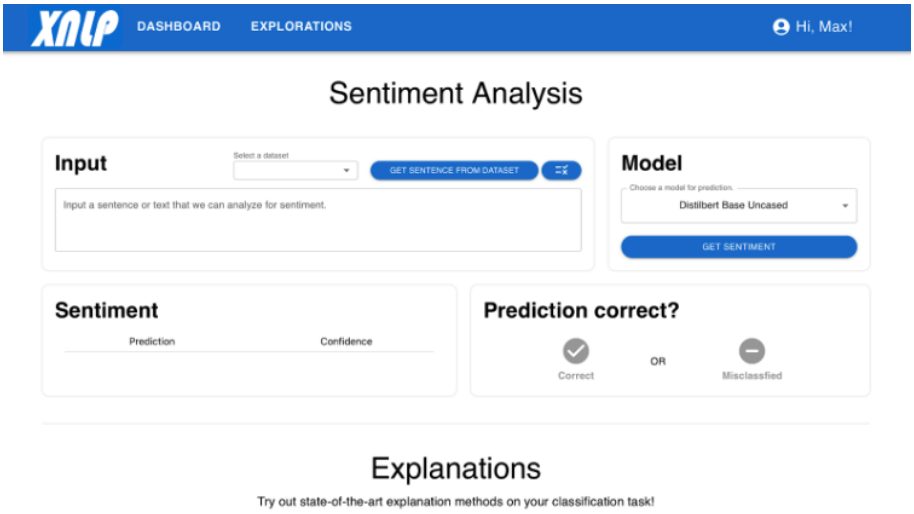

Figure 7: Empty Dashboard

This is the Dashboard of the XNLP tool when the user hasn't entered any input. The user can type or choose a random sentence from a dataset. There is also the option to load misclassified examples. The user can choose between different models. Later, the user can decide if the prediction is correct or incorrect.

3.5.2 Explorer Dashboard

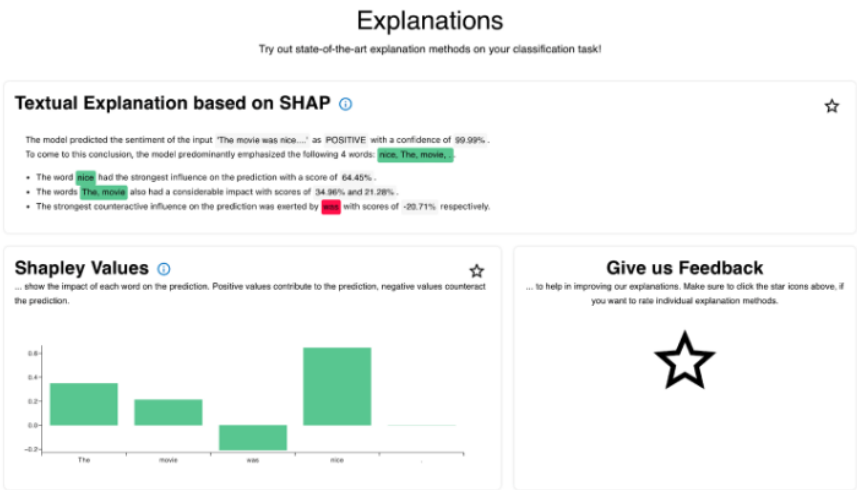

Figure 8: Explorer's Dashboard

The explorer has the explanation methods Textual Explanation based on SHAP and the Shapley Values visualized as Bar Chart. The User can give some feedback by clicking on the star.

3.5.3 Analyst Dashboard

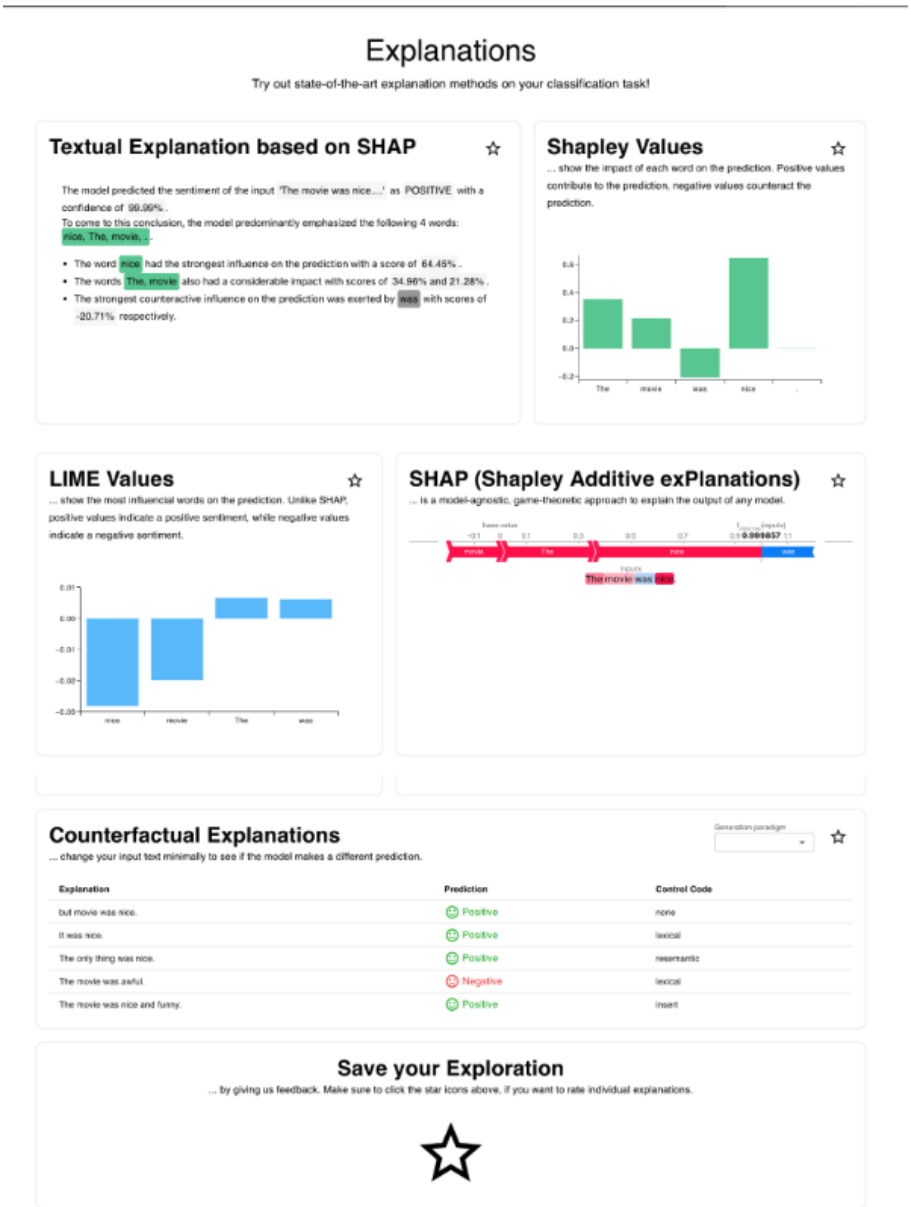

Figure 9: Analyst Dashboard

The analyst has the following methods: Textual Explanation based on SHAP, the Shapley Values visualized as Bar charts, Lime Values, SHAP

(Shapley Additive explanations), and Counterfactuals. The analyst can give feedback and save his explanation outputs by clicking the star icon.

3.6 Exploration Page

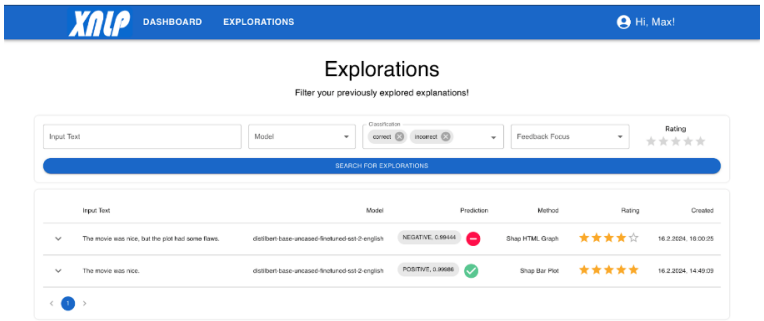

Figure 10: Exploration Page

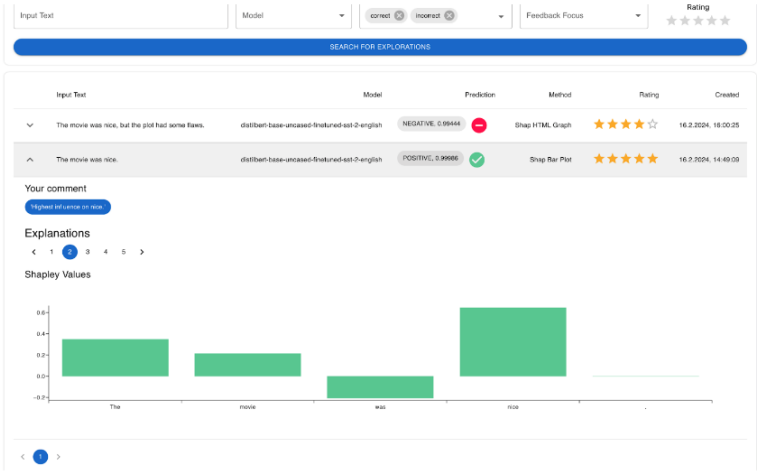

Figure 11: Exploration Page

The Exploration Page is a listing of all his saved explorations. Each saved exploration can be expanded by clicking the arrow on the left side.

3.7 Personal Profile

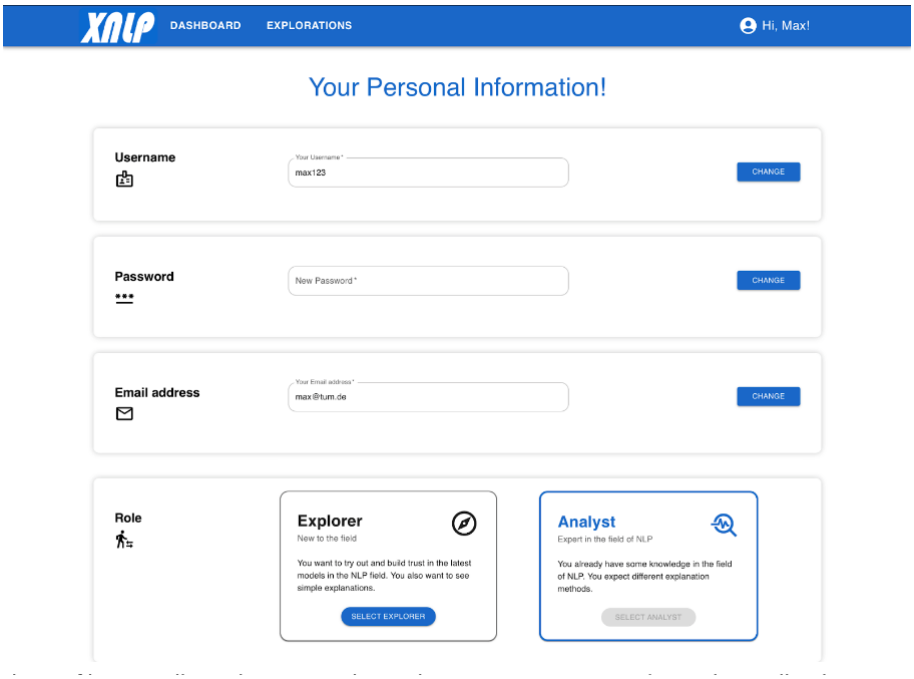

Figure 12: Personal Profile

The Profile page allows the user to change his username, password, email, and selected role.

3.8 Statistics for Admin

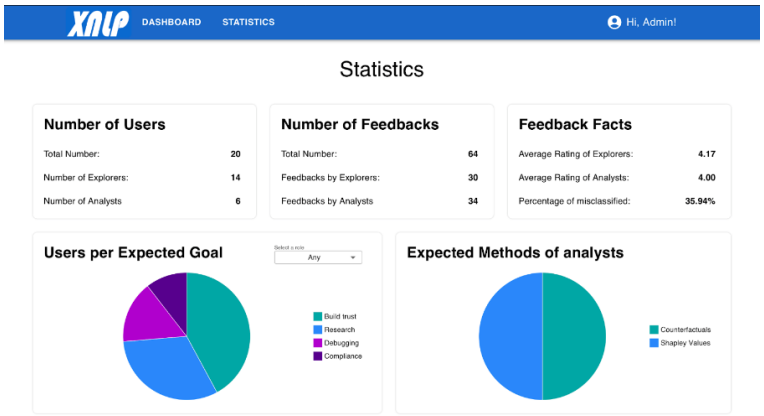

Figure 13: Statistics Page

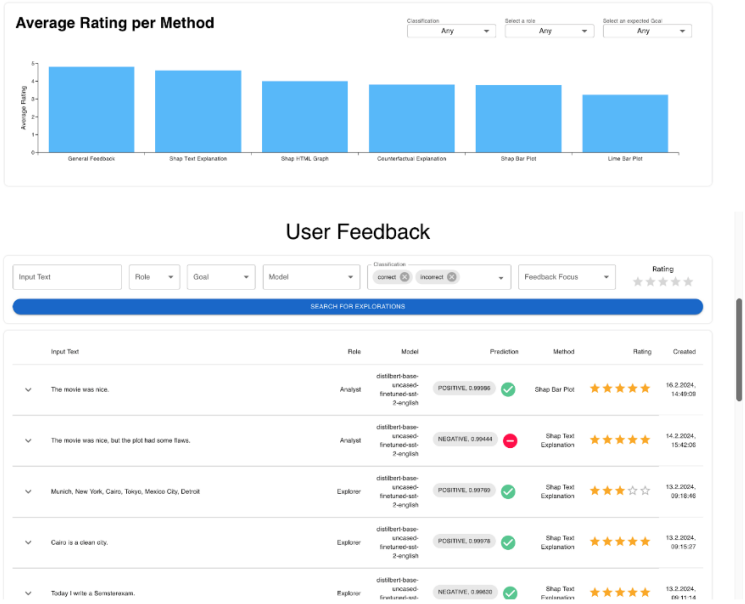

Figure 14: User's Feedbacks Page

The Statistics Page is only visible to Admins. This role is just for administrators of the website. The Statistics allow the admin to analyze

the given feedback. Several feedback options exist to see the feedback facts for particular user groups or specific contexts.

4 Architecture & Technology stack

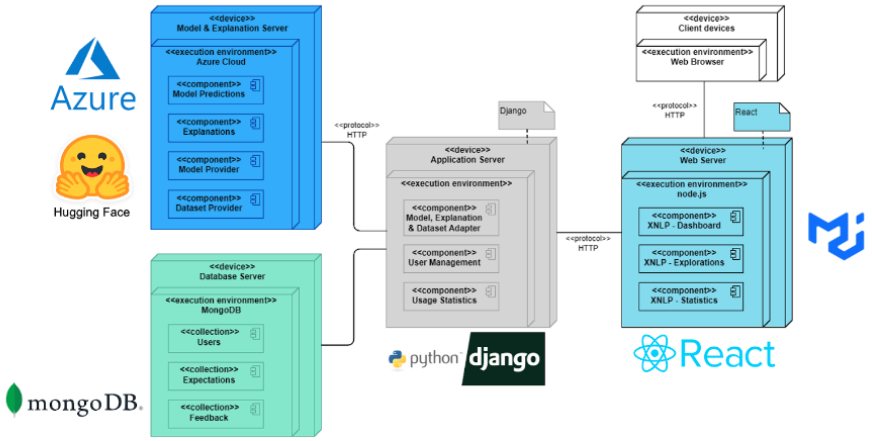

Figure 15: Architecture

This deployment diagram encompasses the solution architecture of our application. The application is separated into two main parts: backend and frontend. The frontend is implemented in JavaScript using React and MaterialUI. The frontend presents the dashboard, explorations, and statistics page to user client devices and can be accessed through a web browser. Model inference, explanations, database operations, etc., are all forwarded to the backend server (Django) via HTTP, which handles user management, statistics, models, datasets, and explanations. User management and usage statistics serve as a single component, which can be easily separated from the model, explanation, and dataset components. The user management and usage statistics component stores its data (user, expectation & feedback) on a MongoDB atlas database. The model & explanation component handles all explanation methods, loads and saves HuggingFace text classification models and datasets, and performs model predictions. This component can be deployed easily using GPU resources to improve

computation performance. In the current configuration of this GitLab project, these two components still reside on the same server. It can be deployed automatically using the Azure cloud (see readme for a detailed description). The frontend can be deployed separately. Use the existing deployment workflow for this.

4.1 Backend Side

Django was preferred on the backend side because our team has Django experience, and this framework can easily integrate with technologies such as huggingface. Django's ready-made components and security features simplify user management, statistics tracking, and model operations. In addition, the RESTful APIs provided using the Django REST framework provide an ideal solution for secure and efficient communication with the frontend. This allows our team to focus on backend development quickly and efficiently.

4.2 Frontend Side

On the frontend side, React and Material-UI were preferred using JavaScript. Thanks to React's flexible and component-based structure, this choice allows us to create quickly interactive user interfaces. Conversely, Material-UI offers a consistent user interface experience with its pre-built and customizable components. Using these technologies allows us to develop a user-friendly and visually impressive frontend. In addition, the frontend has a separate distribution, allowing it to scale and develop independently from the backend. This way, we can create a more flexible and efficient development process as a team.

4.3 Database

We've decided to use a MongoDB document-based approach for data persistence. MongoDB provides us with enough flexibility to handle all types of different data objects, e.g. also unstructured objects, JSON format etc., which can come in handy in an NLP use case. Furthermore, Atlas is free to use and easy to integrate. It can handle user data, HTML format, JSON objects, and fits, saving individual explanations. b

5 Key Takeaways

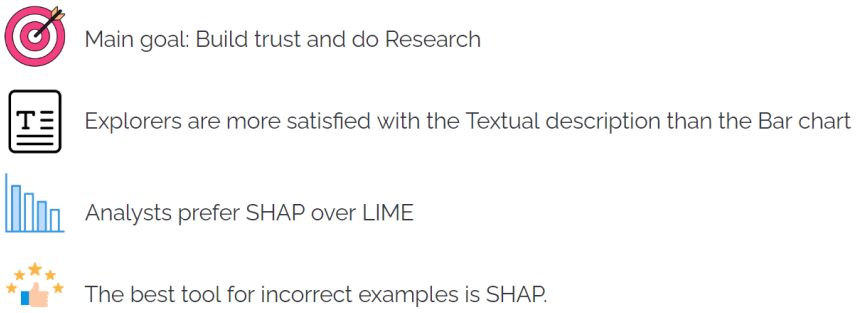

Figure 16: Key Takeaways from the initial deployment

After the initial deployment, we were able to collect 63 feedback objects on the provided explanations from a total of 19 users. The feedback objects were evenly distributed between analyst and explorer users. Since this amount of feedback data is not yet statistically significant, we won't go into further detail besides the vague key takeaways we've taken at the start of this chapter. This chapter leaves some room for further improvement. The app should be tested in a broader context and deployed with more resources for a more extended period to generate helpful feedback data.