# Project Design - Phase 2

Customer requirements

- Accurate Prediction: The customer requires a predictive model that can accurately classify passengers as satisfied or dissatisfied based on their flying experience.

- Real-time Insights: The customer needs a web-based application that provides real-time insights on passenger satisfaction.

- Usability: The application should be user-friendly and intuitive, allowing the customer to easily access and interpret the generated insights.

- Customizability: The customer wants the flexibility to customize the application according to their specific needs and preferences.

- Scalability: The solution should be scalable to handle a large volume of data and accommodate future growth.

- Performance Metrics: The customer requires performance metrics such as accuracy, precision, recall, and F1 score to evaluate the model's performance.

Requirements analysis

- Data Collection: The system should be able to collect data from various sources, including online surveys and social media platforms, to capture passenger feedback effectively.

- Data Preprocessing: The solution needs to handle missing values, outliers, and inconsistencies in the collected data to ensure data quality.

- Feature Selection: The system should identify relevant features that significantly impact passenger satisfaction and consider feature engineering techniques to create informative features.

- Model Building: The solution should include the ability to build and evaluate multiple machine learning models, considering algorithms such as logistic regression, decision trees, random forests, and neural networks.

- Model Evaluation: The system needs to assess model performance using appropriate metrics such as accuracy, precision, recall, and F1 score.

- Real-time Insights: The application should provide real-time insights on passenger satisfaction, allowing the customer to identify areas of improvement promptly.

- Customizability: The solution should be easily customizable, allowing the customer to tailor it to their specific needs and preferences.

- Scalability: The system should be able to handle a large volume of data efficiently and accommodate future growth in data collection.

- User Interface: The application should have a user-friendly and intuitive interface, enabling easy access to generated insights.

Technical architecture
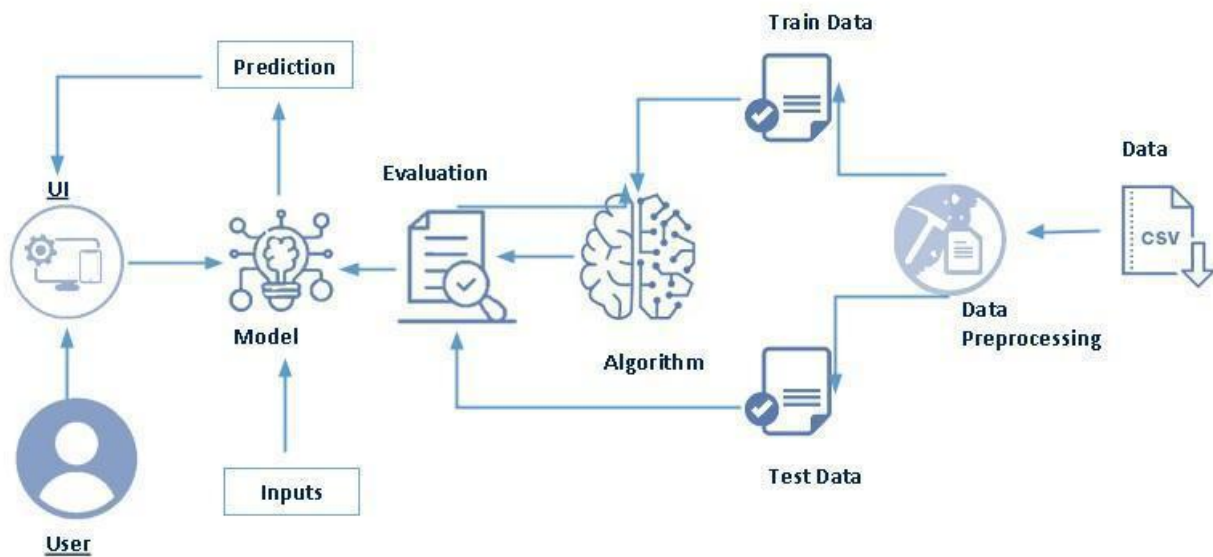
Open Source frameworks

- Data Collection: Utilize tools like BeautifulSoup and Scrapy for web scraping and data collection from online sources.

- Data Preprocessing: Use libraries such as Pandas and NumPy for data cleaning, transformation, and feature engineering.

- Model Building: Implement machine learning models using scikit-learn or TensorFlow, depending on the algorithm chosen.

- Web Application: Consider using frameworks like Flask or Django for web application development, providing a robust and scalable solution.

- Visualization: Utilize libraries like Matplotlib or Plotly for data visualization and generating insightful plots and charts.