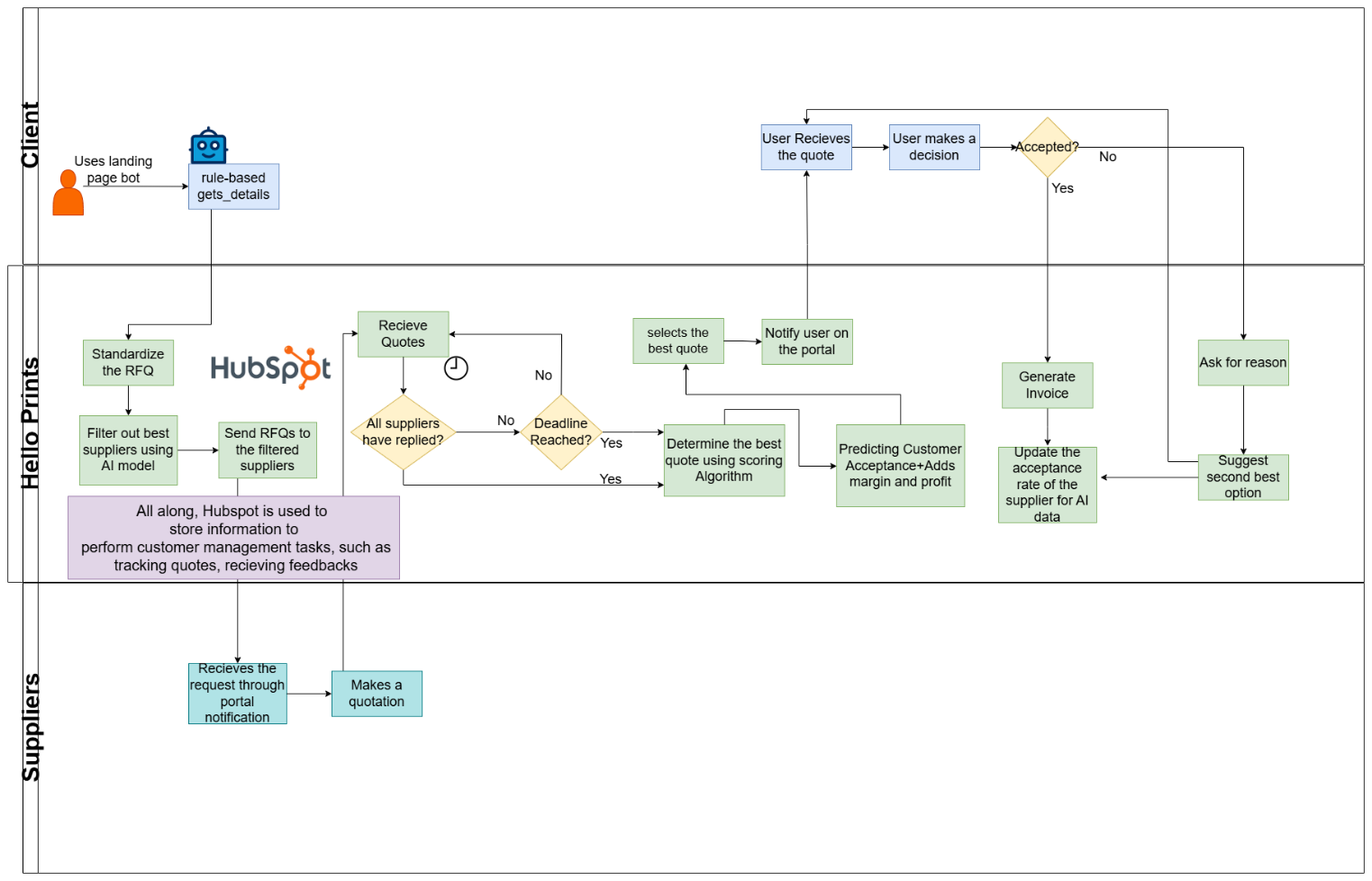
****

**Components Description**

1. **Customer Inquiry and Data Capture**

For this objective, we have designed a rule-based chatbot that will be hosted on the landing page of Hello prints, integrated by Hubspot. The idea is, the chatbot will act as a primary interaction point where the user can place orders, receive final quotes, and inquire about products and services, all without human intervention. The bot uses a rule-based approach which means it will have a set of predefined questions that the customer will answer. It ensures a **structured and guided conversation**, making it easier to get product details. Since the bot follows a predefined decision tree, it minimizes ambiguity and ensures that no critical information is missed. There are three alternatives to this approach

1. **NLP-based Chatbot (Logistic Regression + TF-IDF):** By training a classification model on sample data, we can allow users to express their needs in a more natural language format. TF-IDF vectorization would help the model understand text patterns and classify customer intent accordingly.
2. **Named Entity Recognition using SpaCy**: This approach can extract relevant information from texts without the need for predefined questions.
3. **GPT-based Conversational AI:** Another option is using GPT, which can handle complex queries and understand context better than traditional NLP methods. However, this would require additional resources for fine-tuning and maintaining conversational accuracy. It is also not cost effective for small use-cases where we can use other methods to get same results

Ultimately, the **rule-based chatbot** was selected due to its **efficiency, and ease of deployment**. Future improvements may involve **hybrid approaches stated above**

1. **Supplier Communication**

When we receive the product details, we standardize them using a structured Request for Quotation (RFQ) format that includes all essential information. This ensures consistency and clarity in supplier communication. The standardized RFQ, along with any additional notes, is then sent to the most suitable suppliers.

To determine the top suppliers that are capable of fulfilling the order, we utilize a Linear Regression model, which evaluates suppliers based on criteria such as cost, delivery time, and reliability. Only the highest-ranked suppliers receive the RFQ. Suppliers access these RFQs through their dedicated portal, where they can review requests and submit their responses. All supplier responses are logged in a centralized database which allows seamless tracking and analysis. We wait for either all suppliers to respond or until the deadline expires. We will use Cron Job for this Once the response window closes, our algorithm evaluates the received quotations to select the most optimal supplier.

1. **Data Aggregation and Analysis**

The system aggregates supplier responses, including important details like price, delivery time, reliability, cost, and product type. These responses are collected in a structured format and are represented as records in a DataFrame (as of now), where each supplier’s quote includes key attributes such as price, delivery time, and reliability.

* 1. Algorithm for Scoring Supplier Quotes:

To evaluate supplier quotes, an algorithm is implemented that scores each quote based on multiple criteria. The criteria include:

* Cost: The price quoted by the supplier is evaluated. The lower the price (compared to the customer’s budget), the higher the score.
* Delivery Time: The delivery time required by the supplier is compared to the customer’s desired delivery time. A faster delivery time results in a higher score.
* Reliability: Supplier reliability is taken into account, where a higher reliability score results in a better evaluation.

The scoring is done by calculating a closeness score between each supplier’s quote and the customer’s preferences. The score takes into account the price difference, delivery time difference, and reliability mismatch, penalizing suppliers whose quotes do not meet the customer’s needs.In addition to the manual scoring algorithm, a final score is calculated by considering the supplier’s acceptance probability and margin. This final score ranks suppliers, and the supplier with the highest score is selected.

1. **Quote Optimization**
   1. **Machine Learning for Customer Acceptance Prediction:**

A Random Forest Classifier model is trained using past customer data to predict the likelihood of customer acceptance of a supplier’s quote. The model uses features like price, delivery time, and reliability to make predictions. The target variable is whether the customer accepted or rejected the quote. The trained model generates a predicted probability of acceptance for each supplier’s quote. This probability is then incorporated into the final score calculation, ensuring that the system selects the supplier whose quote has the highest likelihood of being accepted by the customer.

* 1. **Machine Learning for Margin Viability:**

A Random Forest Regressor model is used to predict the profit margin for each supplier’s quote based on the supplier’s price, delivery time, and reliability. This model helps assess whether the quote is financially viable for the company.The regressor predicts the profit margin based on historical supplier data that includes the supplier’s cost, delivery time, and reliability.The predicted margin is used in the final scoring system, where the system gives preference to suppliers offering better financial margins, ensuring profitability.

* 1. **Supplier Ranking and Final Selection:**

Once the individual scores are calculated for each supplier, they are combined into a final score that incorporates: Predicted Customer Acceptance Probability, Closeness to Customer Demand, Profit Margin. The supplier with the highest final score is selected as the optimal supplier for the customer’s needs. This ensures that the quote chosen is the one that balances customer preferences, profitability, and likelihood of acceptance.

1. **Customer Presentation and Feedback Loop**

The end result will be displayed to the customer on the portal where he placed the order. From there, a customer can accept or reject an offer. The supplier responses and customer’s feedback are logged in a centralized data store (e.g., a JSON file as of now). Which will then be used in future machine learning. If the offer is rejected we might offer the second best in line for approval while updating the acceptance of the previous one.

**Technology Stack**

For this supplier response aggregation system, I have chosen a set of technologies that are well-suited for handling data processing, machine learning, and integration with other systems. Below is the proposed technology stack:

**1. Backend:**

* **Python**: Python is a versatile language that is well-suited for handling data analytics and machine learning tasks.
* **Flask or Django (for API Development)**: Flask or Django can be used to create an API for interacting with external systems, enabling the system to receive supplier responses and push recommendations or updates back to the client interface.
* **pandas & NumPy**: These libraries are used for data manipulation and analysis. pandas is especially useful for managing supplier data and customer preferences in a structured format (DataFrame).

**2. Data Processing & Machine Learning:**

* **scikit-learn**: This machine learning library is used for implementing the **Random Forest Classifier** and **Random Forest Regressor** models, which are at the heart of predicting customer acceptance and margin viability. The library provides simple and efficient tools for model training, prediction, and evaluation.
* **Joblib**: Can be used when we are training actual models. (Since in my prototype I am not saving or loading any model, I haven’t used it) It is very useful for saving and loading the trained models efficiently, especially when dealing with large datasets or complex models.

**3. Data Storage:**

* **HubSpot CRM**: Integrated to manage customer interactions, track quotes, and store customer preferences and feedback. This ensures seamless communication and follow-ups with customers.
* **JSON Files**: In my Prototype, I am using JSON files for demonstration purposes

**4. Frontend (for User Interaction,):**

* **React.js**: For more interactivity and a modern, scalable frontend, React.js can be used to build dynamic user interfaces that allow for real-time updates of supplier quotes, feedback, and rankings.

**5. Data Synchronization:**

* **REST API**: To facilitate communication between the system and any external data sources (e.g., incoming supplier responses or customer feedback), REST APIs or WebSockets can be used for real-time data exchange.
* **Celery (Crontab)**: Crontab will be used for running the scoring algorithm. After the deadline has passed (lets say 4 hours), we will run the scoring algorithm on the suppliers that have replied. This will ensure that we reply to the customer on time.

**Challenges & Solutions**

Below are some potential challenges that may occur in implementing a system like this and their proposed solutions:

**1. Real-time Data Synchronization:**

* **Challenge**: Real-time data synchronization is essential when handling customer feedback and supplier responses. Supplier responses may not always be received in real-time. Some suppliers may delay their responses, leading to delays in the decision-making process. Or some customers may not get the quote on time that may affect customer satisfaction
* **Solution**:
  + Use an asynchronous data processing approach with tools like Celery or task queues to process supplier responses as they arrive.
  + Implement a timeout mechanism that allows the system to move forward with available supplier responses while still attempting to retrieve delayed responses in the background.
  + With HubSpot integrated, real-time data about customer feedback and decisions will be automatically logged. As soon as the customer accepts or rejects a quote, HubSpot workflows can trigger automated updates in the system. For example, if the customer approves the quote, HubSpot can send a notification to the relevant teams and update the customer profile accordingly, while triggering AI-driven adjustments to future supplier recommendations.
  + Instant Updates for Customers: Through the integration of real-time notifications via HubSpot, customers will be instantly informed about key updates related to their quotes. Customers can receive timely alerts (via email and within their HubSpot portal) as soon as new supplier data is processed or customer feedback is incorporated. This minimizes lag time and ensures customers always have the latest information on time.

**2. Handling Supplier Response Delays:**

* **Challenge**: Some suppliers may submit quotes late, or their data may be incomplete or incorrect. This could lead to missed opportunities or skewed decisions.
* **Solution**:
  + Implement **validation mechanisms** that check for the completeness and accuracy of supplier data before it's processed.
  + Incorporate **penalties in the scoring system** for suppliers with frequent delays or incomplete data, making their quotes less favorable in the decision-making process.

**3. Scalability:**

* **Challenge**: As the number of suppliers and customer interactions grows, the system may struggle to handle increasing data volumes and real-time predictions.
* **Solution**:
  + Use horizontal scaling by distributing tasks across multiple servers or cloud infrastructure to handle more data.
  + Optimize machine learning models for performance. For instance, use model compression techniques (like quantization) or distributed learning frameworks to train models on larger datasets.
  + Implement database sharding to distribute supplier data across multiple databases and optimize query performance.

**4. Data Quality and Consistency:**

* **Challenge**: Supplier data may vary in format, and differences between different supplier responses can result in incorrect analysis.
* **Solution**:
  + Use data cleaning algorithms to detect anomalies and to ensure that supplier data is consistent and comparable.
  + Data validations to ensure all required fields are properly populated before processing. It will help ensuring accuracy and consistency.

**Business Impact Discussion:**

1. **Operational Efficiency:**

Our solution directly addresses the inefficiencies in HelloPrint's current manual quotation process by automating several key steps.

* The system uses AI algorithms to automatically evaluate and rank suppliers based on multiple criteria. This human error and the time spent.
* As supplier quotes come in, the system immediately processes the data and updates the customer-facing interface. This allows instantly access the most competitive offers without having to wait.
* With HubSpot integrated, the entire process, including supplier responses, customer feedback, and quote generation, is seamlessly captured in the CRM. This leads to faster follow-ups and better tracking of each quotation.
* By automating data aggregation, quote scoring, and response handling, the system dramatically improves the speed and efficiency of the quotation process.

2. **Increased Win Rates:**

Our approach is specifically designed to increase the likelihood of customer acceptance by focusing on providing highly competitive and personalized quotes:

* Using historical data from past customer interactions, the system predicts the most likely suppliers to win a contract based on past acceptance rates.
* Customers receive tailored quotes with clear, competitive pricing, delivery times, and reliability scores. By using AI to optimize quotes, we ensure that customers receive the best possible offers that meet their needs while increasing the likelihood of acceptance.
* With ongoing feedback from customers and suppliers, the system continuously refines its predictions and recommendations. If a customer rejects a quote, the system learns from the feedback to improve future quote recommendations, ensuring better win rates over time.

3. **Profitability:**

Our system is designed to ensure that every accepted quote maintains or improves HelloPrint's profit margins:

* The system incorporates a machine learning model that predicts the profitability of each quote. This allows HelloPrint to assess the margin viability of each offer before it is sent to the customer, ensuring that only profitable quotes are finalized.
* Over time, the AI model tracks supplier performance, identifying suppliers that consistently deliver high-quality products at competitive prices. This enables HelloPrint to prioritize suppliers that align with profitability goals, ultimately reducing costs and increasing margin stability.

**4. Alignment with Business Objectives:**

Our solution aligns perfectly with HelloPrint’s overall business objectives of increasing revenue, improving efficiency, and maintaining profitability:

* By significantly reducing the time it takes to generate and send quotes, HelloPrint can respond to customer inquiries more quickly, leading to higher conversion rates and increased revenue opportunities. The AI-driven approach ensures that HelloPrint consistently offers competitive prices, which improves the chances of securing business.
* The system improves internal operations and reduces human error with real-time quote generation, AI-backed supplier selection, and integrated workflows with HubSpot.
* The system’s ability to evaluate margin viability and supplier performance ensures that HelloPrint maintains healthy profit margins even when offering competitive prices.
* The constant feedback loop allows the AI to refine future supplier selections, ensuring long-term profitability.

You can find the prototype here