

# Understanding the Factors with SHAP (after model creation)

---

- **nzz**: This feature has a high impact on the model's predictions, with both high and low values (red and blue dots) influencing the SHAP values significantly (long shape).
- **num\_read\_articles**: Higher values (red dots) of this feature tend to move the SHAP values to the left, indicating that users who read more articles are less likely to buy. Conversely, lower values (blue dots) have a neutral or slightly positive impact on the prediction, contributing to a higher likelihood of buying.
- **number\_of\_newsletters**: This feature also has a significant impact, with higher values increasing the probability of buying.
- **contentTypeArticles**: The impact of this feature varies, but higher values tend to have a positive contribution.
- **time\_from\_last\_session**: Lower values (blue dots) tend to have a positive impact on the prediction, suggesting that users who have recently visited are likely to buy.
- **days\_since\_registration**: Higher values have a positive impact, meaning users who registered a long time ago are likely to buy.
- **other**: This feature shows a mixed impact with both high (red) and low (blue) values affecting the prediction differently.
- **mostLikedCategories\_NotAvailable**: This categorical feature shows that if the preferred category is not available, it negatively affects the likelihood of buying.
- **num\_display\_articles**: The number of display articles also has a variable impact, with higher values contributing both positively and negatively in different contexts.

## Model Improvements

---

### Feature Engineering

- Create new features by combining existing features. For example, interactions between `num_read_articles` and `time_from_last_session`.
- Extract time-based features such as day of the week, month, or season if applicable.

### Hyperparameter Tuning

- Perform extensive hyperparameter tuning using techniques like Grid Search or Random Search with cross-validation.
- Use other machine learning algorithms like Random Forest, Gradient Boosting Machine (GBM) like LightGBM or CatBoost.

### Feature Selection

- Remove non-significant variables.
- Principal Component Analysis Non-Linear Dimensionality Reduction.

# Deployment Steps

---

## Final Model Training

- Select Deployment Environment (Heroku, Shiny Server, GCP).
- Create Virtual Environment and install python.
- Containerization with Docker.
- Create a Web Service Flask/Shiny.
- Deployment of the branch on the server.
- Configure CI/CD Pipeline GitHub Actions.

## Collaboration Strategies

---

- Schedule initial meetings to understand the business goals, key performance indicators (KPIs), and specific problems the machine learning solution should address.
- Hold regular meetings (e.g., weekly or bi-weekly) to update the business team on progress, challenges, and next steps.
- Use visual aids such as dashboards, charts, and presentations to make technical concepts accessible.
- Use Agile methodologies, such as Scrum or Kanban, to ensure iterative development and continuous feedback.
- Create early prototypes or minimum viable products (MVPs) to demonstrate the potential of the solution.
- Conduct UAT sessions where the business team can interact with the solution, provide feedback, and identify any issues before full deployment.