**Applied AI Internship Task – Swiggy**

**Project Title:** AI-Powered Restaurant Review Summarization & Risk Prediction  
**Author:** Manya Chugh  
**Date:** 02 Sept, 2025

**1. Introduction**

For this project, the goal was to help restaurant partners get meaningful insights from customer reviews. I was provided a dataset containing reviews and restaurant information, and my task was to summarize feedback, identify high-risk restaurants, and create a simple web app that shows these insights in real-time.

So, I developed a solution that:

1. Summarizes customer reviews into short meaningful insights.
2. Predicts which restaurants are at high risks.
3. Provides explainability and factors which drive the risk prediction.
4. Deploys the solution in a simple web app.

This combines NLP for summarization, machine learning for risk prediction, and SHAP for interpretability, offering a complete end-to-end solution.

**2. Dataset & Preprocessing**

The dataset included customer reviews, ratings, monthly sales, staff turnover, complaints count, and a risk label indicating whether a restaurant was high risk.

**Data Cleaning:**

* Filled missing reviews with "no review" to ensure completeness.
* Converted the date column into a proper datetime format for consistency.
* Lowercased all review text to maintain uniformity.

**For Visualization:**

* Risk label distribution highlighted the proportion of high-risk vs low-risk restaurants.
* Complaints by risk label revealed a clear link between complaints and risk.
* Correlation heatmap provided insights into how features like sales, ratings, and complaints relate to each other.

**Feature Engineering:**

* **Sentiment Score:** Calculated using TextBlob to capture whether reviews were generally positive or negative.
* **Review length:** Counted words in each.
* Combined these with features : monthly\_sales, complaints\_count, avg\_rating, staff\_turnover.
* Features were standardized using StandardScaler to improve model performance.

**Data Preparation:**

* The final feature matrix X and target vector y were split into 80% training and 20% testing, ensuring both high-risk and low-risk restaurants were represented.

**3. Summarizing Reviews (NLP)**

The goal was to condense all reviews into a meaning 2-3 sentence summary for quick and valuable insights.

**Model Choice:**

* I used BART-large-CNN, a transformer model that was trained on CNN daily mail dataset for summarization tasks.
* As it is suitable for abstractive summarization. It understands context across multiple reviews and generates coherent summaries.

**Evaluation:**

* Used ROUGE scores for word overlap and cosine similarity for semantic similarity.

| **Review** | **ROUGE-1 F1** | **ROUGE-2 F1** | **ROUGE-L F1** | **Cosine Similarity** |
| --- | --- | --- | --- | --- |
| 1 | 0.720 | 0.696 | 0.720 | 0.763 |
| 2 | 0.643 | 0.615 | 0.643 | 0.665 |
| 3 | 0.485 | 0.452 | 0.485 | 0.563 |
| 4 | 0.000 | 0.000 | 0.000 | 0.000 |
| 5 | 0.839 | 0.828 | 0.839 | 0.810 |

**Observations & Challenges:**

During the implementation of BART, the summarizer did not generate meaningful outputs for the restaurant reviews. It was found that many reviews in the dataset contained random or incomplete text. Since BART is a generative model trained on coherent and structured text, it struggled to condense such noisy input into meaningful summaries. As a result, the summaries produced did not accurately reflect the content of the reviews. But on real words and reviews, it would work just fine.

**4. Predicting High-Risk Restaurants (ML)**

Using restaurant metrics and review-derived features, I trained three models: **Logistic Regression, XGBoost, and SVM**.

| **Model** | **Precision** | **Recall** | **F1-Score** | **ROC-AUC** |
| --- | --- | --- | --- | --- |
| Logistic Regression | 0.879 | 0.717 | 0.790 | 0.981 |
| XGBoost | 1.000 | 1.000 | 1.000 | 1.000 |
| SVM | 0.948 | 0.972 | 0.960 | 0.999 |

**Insights and conclusion:**

It can be clearly seen that Xgboost outperformed and achieved a perfect score with all metrics equal to 1.0, reason most likely being its ability to capture non-linear interactions and ability to handle class imbalance(which can be seen in our dataset{safe restaurants heavily outnumber risky restaurants } Although perfect metrics on a test set may indicate overfitting specially if the dataset is small or not very diverse, But for now it can be concluded that XGBoost performed excellent on this dataset, real world validation is recommended.)SVM also performed really well with an f1-score of 0.96 and ROC-AUC of 0.99. Logistic Regression performed decently but had a lower recall of 0.72, thus, it may have missed some high-risk restaurants. Thus based on these results, XGBoost is the best choice for deployment and further explainability analysis.

**5. Explainability (SHAP Analysis)**

For explainability, I used SHAP to understand feature contributions for XGBoost.

**Key Features Influencing Risk:**

1. Average Rating (avg\_rating) – Lower ratings increase risk.
2. Complaints Count (complaints\_count) – More complaints push risk higher.
3. Monthly Sales (monthly\_sales) – Lower sales contribute to high-risk predictions.

**Case Studies:**

1. **Restaurant ID: 65** – Predicted High Risk
   * SHAP values for the first review:
     + avg\_rating = -1.44 (pushing toward high risk)
     + complaints\_count = 1.72 (pushing toward high risk)
   * Low ratings and moderate complaints drive this restaurant into the high-risk category.
2. **Restaurant ID: 6** – Predicted Low Risk
   * SHAP values for the first review:
     + avg\_rating = 1.26 (pushing toward low risk)
     + complaints\_count = -1.49 (pushing toward low risk)
   * Strong ratings and fewer complaints keep this restaurant in the low-risk category.

**Insights:**

* Restaurants can clearly see which factors influence risk and address them proactively.

**6. Deployment & Drift Detection :**

The solution was deployed as a Streamlit web app:

* Select a restaurant ID to see summarized reviews.
* Check High Risk / Low Risk predictions.
* View top 3 features driving the prediction.

**Drift Detection (Bonus):**

* The app checks for new or unusual words in incoming reviews.
* For now, reviews are consistent with the training data. But the system is ready to flag unusual patterns in the future, ensuring long-term reliability.

**Demo video link:**

* [**LINK**](https://drive.google.com/file/d/1XlcrxRwfYMZUyz4f2EcJ24mXm1RYdZ51/view?usp=sharing)https://drive.google.com/file/d/1XlcrxRwfYMZUyz4f2EcJ24mXm1RYdZ51/view?usp=sharing

**7. Challenges & Learnings**

* Review texts in dataset were extremely short or random which made summarization harder and time taking to understand how well it worked.
* XGBoost achieved perfect metrics, which may indicate overfitting.
* I found SHAP explainability particularly interesting because it adds transparency and showing exactly why the model made a prediction and making the results actionable in real-world scenarios.

**Potential Improvements:**

* Fine-tune BART for better handling of short or sparse reviews.
* To deal with class imbalance which was really high in our dataset, Oversampling could be done or even new/ more features could improve results.
* Streamlit app can be enhanced with dashboards for trend monitoring.

**8. Conclusion**

This project provides a **complete, end-to-end AI solution** for restaurant partners:

* **Summarizes reviews** into concise insights.
* **Predicts high-risk restaurants** accurately using XGBoost.
* Provides **transparent explanations** with SHAP.
* Offers a **user-friendly web interface** and drift detection for long-term monitoring.

It demonstrates **practical AI skills** in NLP, machine learning, explainability, and deployment—ready for real-world application in the food-tech domain.