Hackathon Report

Team 27

1 Introduction

The hackathon was an intensive event that brought together a diverse team of data scientists, engineers, and domain experts with the goal of developing innovative data processing and analysis solutions. Over the course of the event, we explored various approaches to organizing, analyzing, and modeling our dataset. Below is a detailed breakdown of our work, including successes, challenges, and lessons learned.

2 Organizing Data for Efficient Processing

The first step of our project involved gathering and structuring our dataset to ensure efficient processing and analysis. Given the volume and complexity of the data, we faced challenges related to missing values, inconsistent formatting, and high computational costs.

2.1 Data Cleaning and Preprocessing

- **Handling Missing Values:** We implemented imputation techniques, including mean and median replacement, as well as advanced interpolation methods for time-series data.
- **Standardization & Normalization:** We applied Min-Max scaling and Z-score normalization to ensure numerical stability in our models.
- **Encoding Categorical Data:** Used one-hot encoding for nominal variables and ordinal encoding where relationships between categories existed.
- **Outlier Detection & Removal:** Applied methods such as Tukey's fences and z-score filtering to identify and remove outliers that could skew our models.

2.2 Database Optimization

- **Indexing Strategies:** Created indexes on frequently queried columns to speed up data retrieval.
- **Partitioning Large Datasets:** Applied horizontal partitioning based on key attributes to optimize query performance.
- **Efficient Data Storage:** Converted raw CSV files into optimized formats like Parquet, which significantly reduced storage requirements and improved I/O operations.

3 A Failed Experiment: Inverse Chain Method

One of the novel approaches we attempted was the Inverse Chain Method (ICM), which we hypothesized could reveal hidden relationships within sequential data points. However, despite our efforts, the method did not yield useful results.

3.1 Hypothesis and Methodology

- The Inverse Chain Method aimed to reverse-engineer a sequence of events, identifying key precursors to specific outcomes.
- The process involved constructing dependency chains by backpropagating through time-series data and identifying key influencing factors.
- We implemented a recursive model that traced back significant changes in data points to their root causes.

3.2 Challenges and Failures

- **Computational Complexity:** The recursive nature of ICM required substantial memory and processing power, making it infeasible for large datasets.
- **Lack of Predictive Power:** The method failed to generalize well, as the extracted chains did not provide meaningful predictive insights.
- **Overfitting Issues:** Due to the excessive number of variables considered, the model tended to overfit to specific cases, reducing its practical applicability.

3.3 Lessons Learned

Despite its failure, the Inverse Chain Method provided valuable insights into sequential data processing and highlighted the need for more robust approaches. We decided to pivot towards alternative modeling techniques.

4 Exploring Alternative Methods: Generalized Additive Models (GAM)

Following the failure of ICM, we explored Generalized Additive Models (GAM) as a more interpretable and flexible modeling approach.

4.1 Why GAM?

- Unlike traditional linear models, GAM allows for non-linear relationships between predictors and outcomes.

- It provides interpretability through smooth functions, making it easier to understand complex relationships in the data.

4.2 Implementation and Findings

- We used penalized spline regression to model non-linear dependencies between key features.
- Applied cross-validation to optimize the number of basis functions and smoothness penalties.
- Compared GAM's performance with linear regression and random forests. Results showed that GAM provided better interpretability while maintaining competitive accuracy.

4.3 Key Takeaways

- GAM successfully captured non-linear relationships while avoiding overfitting.
- The interpretability of GAM helped us identify key drivers in our dataset, making it a valuable tool for future analyses.
- While GAM performed well, it struggled with high-dimensional data, leading us to explore more probabilistic approaches.

5 Bayesian Method for Future Cost Estimation

The final phase of our hackathon project involved implementing a Bayesian approach for future cost estimation. Bayesian inference provided a robust framework for dealing with uncertainty and incorporating prior knowledge into our predictions.

5.1 Bayesian Modeling Process

1. **Defining Prior Distributions:** Incorporated domain knowledge to set informative priors for key parameters such as cost fluctuations and risk factors.
2. **Likelihood Estimation:** Used historical data to define likelihood functions representing the probability of observed costs given underlying parameters.
3. **Posterior Inference:** Applied Markov Chain Monte Carlo (MCMC) methods to sample from posterior distributions.
4. **Prediction and Uncertainty Quantification:** Generated probabilistic forecasts with credible intervals, providing a more nuanced understanding of cost variations.

5.2 Results and Insights

- The Bayesian model produced probability distributions rather than single-point estimates, allowing for better risk assessment.
- Compared to traditional regression approaches, Bayesian inference provided better uncertainty quantification and adaptability to new data.
- The model successfully captured long-term trends and was more resilient to data shifts.

6 Conclusion

Our hackathon experience was both challenging and rewarding. We explored various modeling techniques, learned valuable lessons from failures, and ultimately implemented a robust Bayesian forecasting model.