Case Study: Customer Retention and Renewal Model for a Basketball Team

# 1.Discovery

The goal of this project is to build a comprehensive retention model to predict customer renewal behavior for a local basketball team. The objectives are threefold:

Churn Prediction: Estimate the likelihood of each customer renewing their season tickets.

Customer Segmentation: Group customers based on their behaviors and engagement levels.

Lifetime Value Prediction: Forecast the future value of each customer to prioritize retention efforts.

A multi-model approach was adopted to achieve these objectives:

**Churn Prediction:** Implemented classification algorithms such as Logistic Regression, Random Forest, and XGBoost.

**Customer Segmentation:** Used clustering techniques, including K-Means, Mini-Batch K-Means, Gaussian Mixture Models, and Agglomerative Clustering.

**Lifetime Value Prediction:** Regression models like Linear Regression, Random Forest Regressor, and XGBoost Regressor were employed.

The combination of these models provides a holistic view of customer behaviors and aids in crafting targeted retention strategies.

# 2. Data Preparation and Initial Data Analysis (IDA)

The provided datasets were cleaned and merged, ensuring a solid foundation for model building. Key steps included:

## Data Cleaning:

* + Merged Customer Details and Renewal Details on Customer\_ID.
  + Missing values in Zip\_Code\_Distance\_From\_Stadium were filled with the mean.
  + Dropped rows with missing Renewal\_or\_Non-renewal\_Date.
  + Fixed the wrong leap year date
* Feature Engineering:

Created new features that could enhance model accuracy, such as:

**Time\_Since\_Last\_Renewal:** The number of days since the customer last renewed.

**Game\_Attendance\_Ratio:** Proportion of games attended to total games.

**Engagement\_Score**: Sum of phone calls, live conversations, and emails sent in the 90 days before renewal.

**Revenue\_Per\_Game**: Calculated as the total revenue per game based on the average price per seat and total seats held.

**Games\_Missed\_Ratio**: Ratio of games missed to total games.

**Loyalty**: A binary feature indicating whether the customer has been a season ticket holder for more than 3 years.

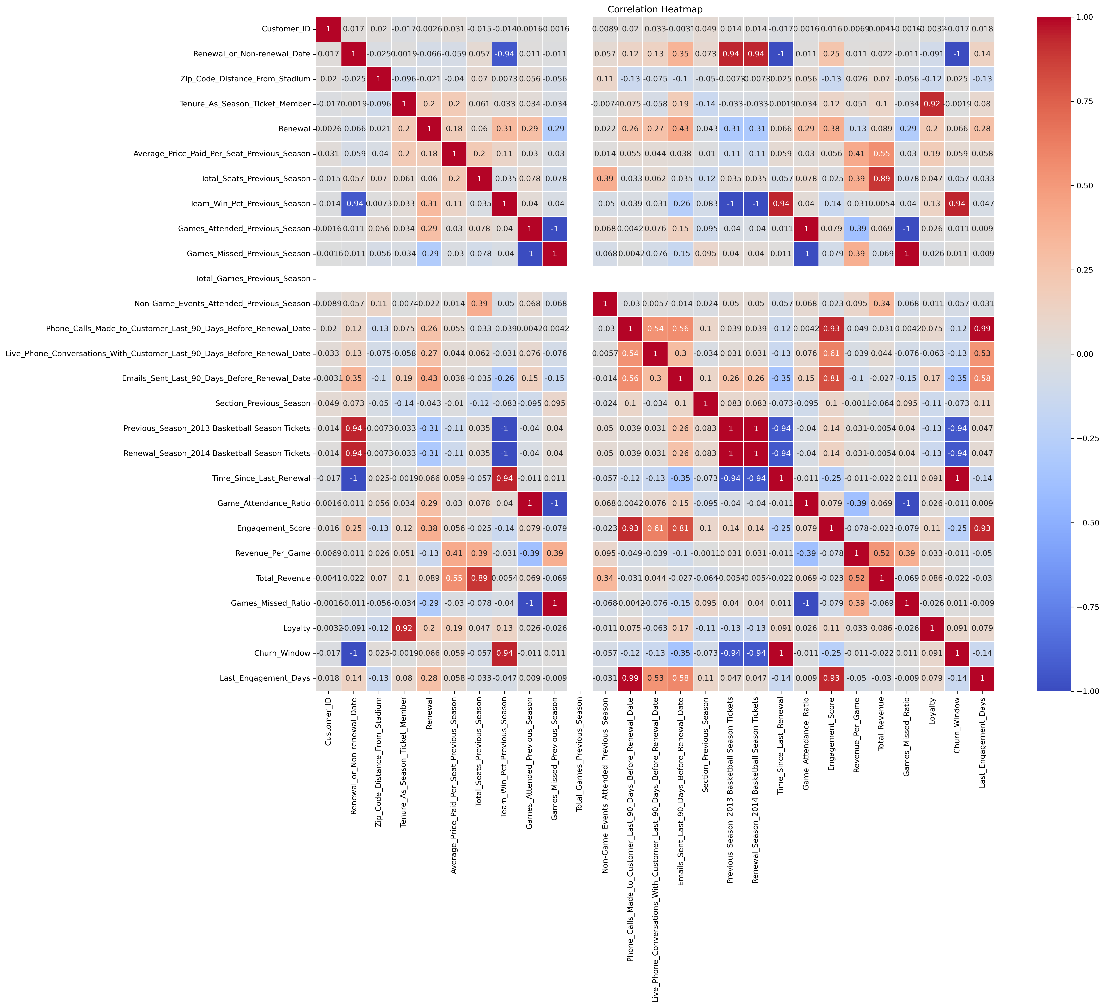
**Additional data that could improve predictions include:**

1. Customer satisfaction surveys.
2. Social media interaction data.
3. External factors such as relocation or income changes.
4. Detailed team performance metrics (e.g., star player acquisitions).

# 3. Model Planning and Exploratory Data Analysis (EDA)

**Visualizations and Analyses**:

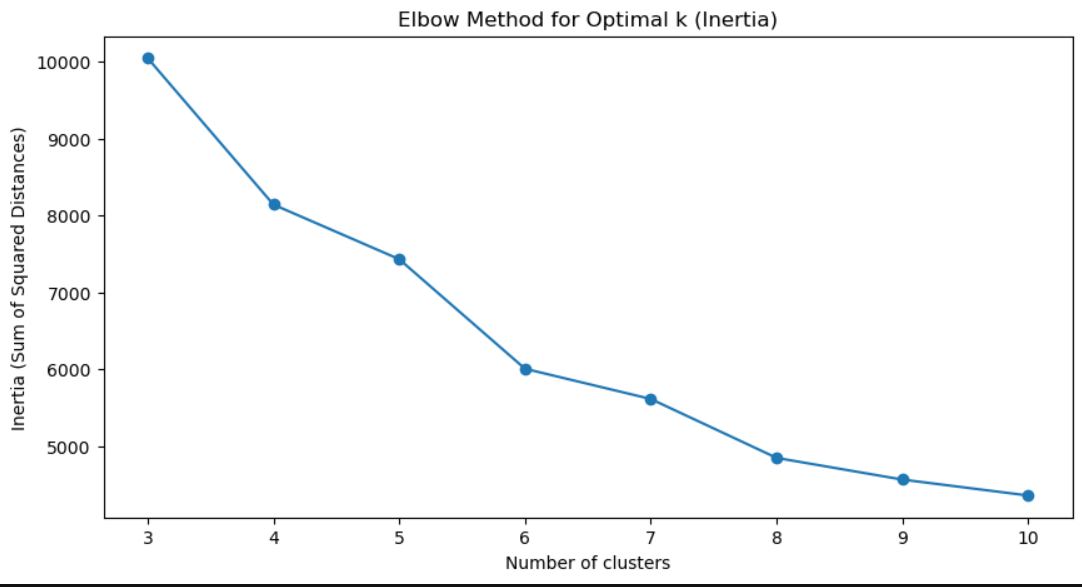
**Correlation heatmaps:** Used to identify relationships between variables and potential multicollinearity issues.



**Distribution plots:** To understand the spread and skewness of numerical variables.

**Box plots:** To identify outliers and understand the relationship between categorical and numerical variables.

**Silhouette score** **and elbow method plots**: Used to determine the optimal number of clusters for customer segmentation.



A graph with a line and numbers

Description automatically generated

These visualizations helped in:

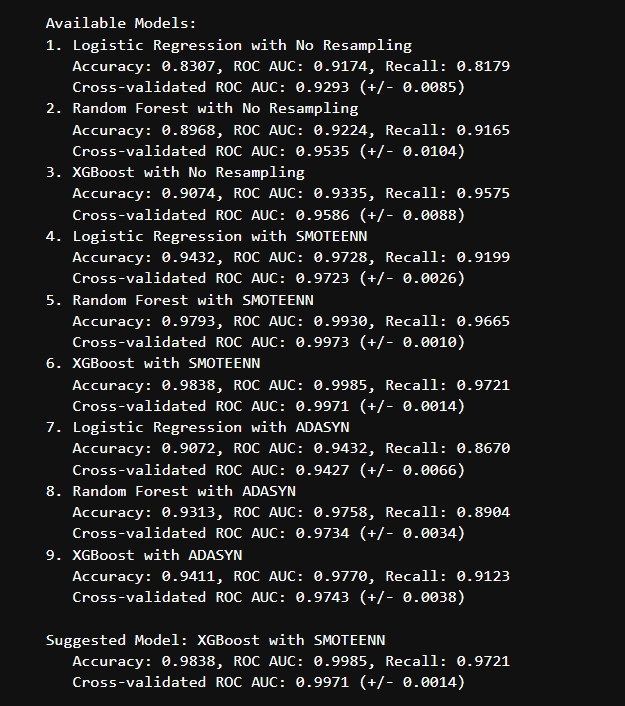
1. Identifying the most important features for each model
2. Detecting and handling outliers
3. Guiding feature engineering decisions (e.g., creating composite scores like Engagement\_Score)
4. Selecting the appropriate algorithms for each task

# 4. Model Building

We built three main models using Python:

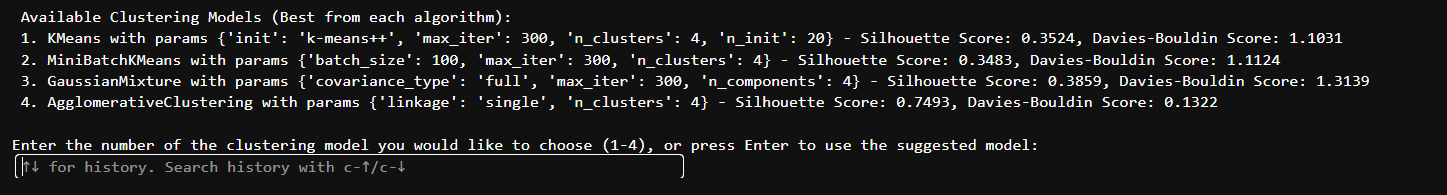
### Churn Prediction Model:

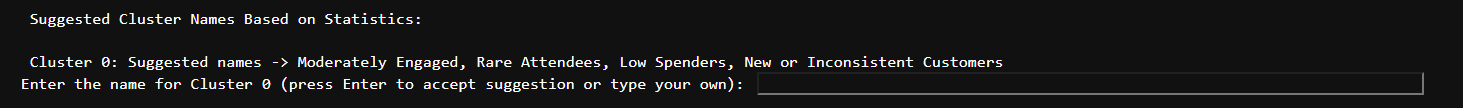
* **Algorithms Used:** Logistic Regression, Random Forest, and XGBoost were employed.
* **Data Splitting:** The dataset was split into training and testing sets using stratified sampling.
* **Feature Selection:** Ridge Regression and Variance Inflation Factor (VIF) were applied to select the best-performing features.
* **Hyperparameter Tuning:** GridSearchCV and RandomizedSearchCV were used to optimize hyperparameters.
* **Performance Metrics:** The models were evaluated using accuracy, ROC AUC, and recall metrics.



### Customer Segmentation Model:

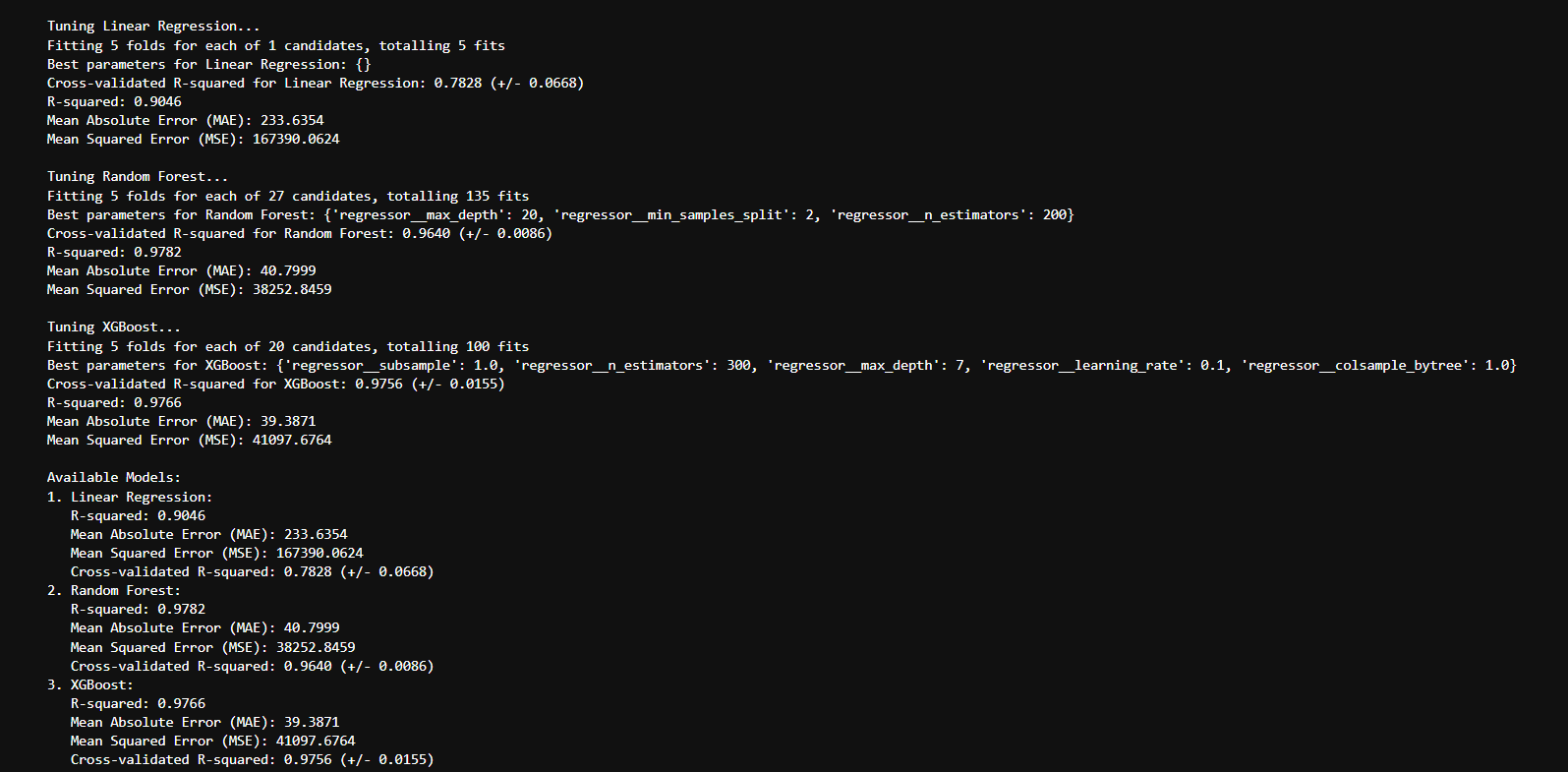
* **Clustering Algorithms:** K-Means, Mini-Batch K-Means, Gaussian Mixture Models, and Agglomerative Clustering were tested.
* **Evaluation:** The silhouette score and Davies-Bouldin index were used to evaluate cluster quality.
* **Optimal Clusters:** The elbow method and silhouette analysis were applied to determine the optimal number of clusters.
* **Cluster Naming:** Clusters were named based on their characteristics (e.g., "Highly Engaged" or "Frequent Attendees").





### Lifetime Value Prediction Model:

* **Algorithms Used:** Linear Regression, Random Forest Regressor, and XGBoost Regressor.
* **Evaluation Metrics:** R-squared, Mean Absolute Error (MAE), and Mean Squared Error (MSE) were used to assess model performance.
* **Cross-Validation:** Models were cross-validated to ensure robustness and prevent overfitting.



The code demonstrates a robust approach to model building, including feature selection, hyperparameter tuning, and model evaluation.

# 5. Communicate Results

#### A user-friendly Streamlit dashboard was developed to allow key stakeholders to interact with the models in real-time. The dashboard includes:

**Churn Prediction**: Users can input customer data and receive a churn likelihood prediction, with a detailed explanation of the key factors influencing the outcome.

**Customer Segmentation**: The dashboard displays which segment a customer belongs to and provides insights into the characteristics of each segment.

**Lifetime Value Prediction**: Provides an estimate of a customer's future revenue contribution, helping prioritize high-value customers.

#### Actionable Insights:

**Retention Strategies**: Use churn predictions to identify customers at high risk of leaving and implement retention campaigns.

**Targeted Engagement:** Tailor engagement strategies based on customer segmentation (e.g., offering incentives to low-engagement segments).

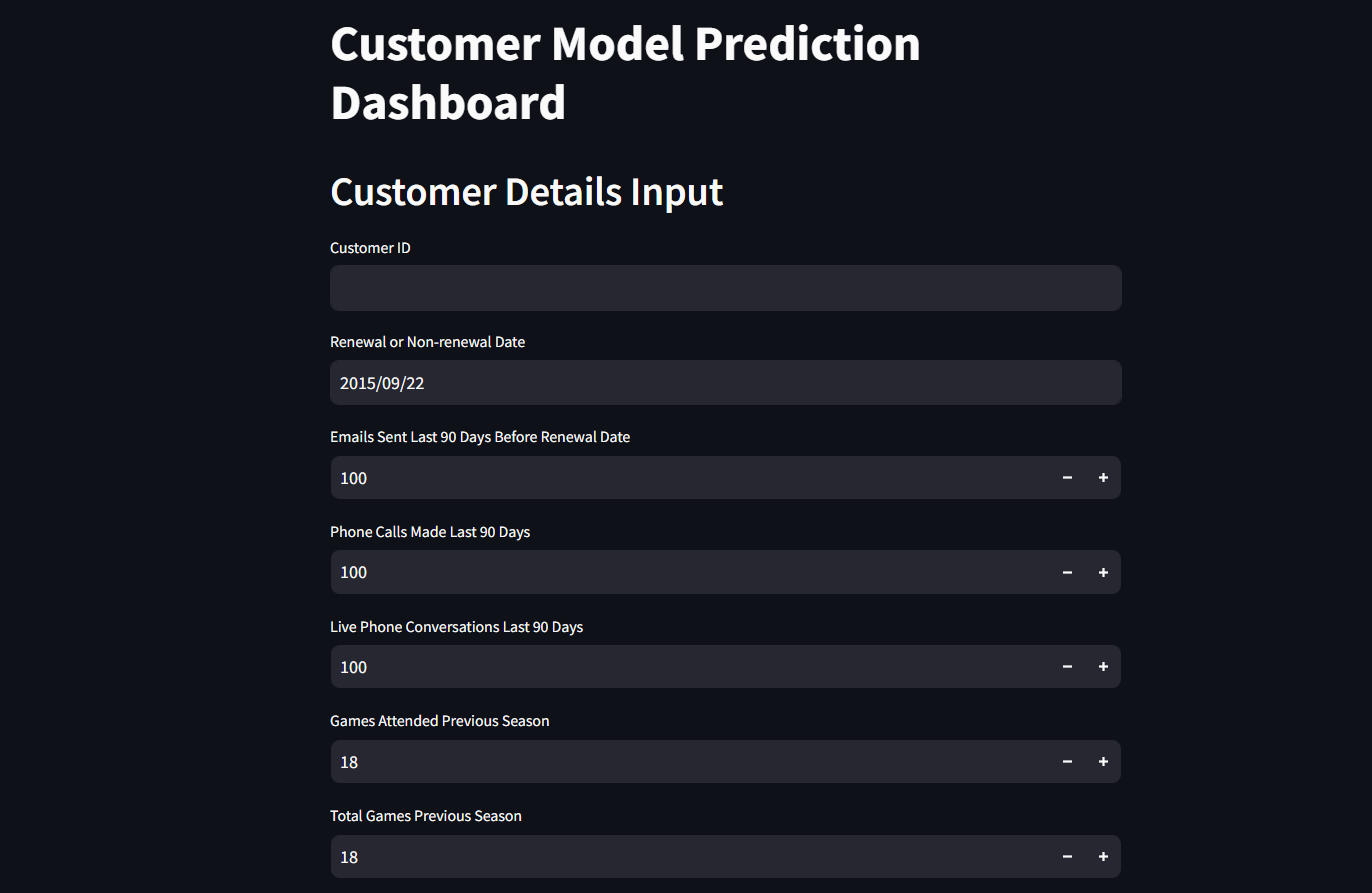
**Revenue Optimization:** Prioritize high-lifetime-value customers for personalized offers and loyalty programs.

#### Key Metrics:

Churn Model: Accuracy, ROC AUC, recall.

Segmentation Model: Silhouette score.

Lifetime Value Model: R-squared, MAE, MSE.



A screenshot of a computer

Description automatically generated

# 6. Operationalize

**Advanced Deployment Strategy:**

* **Containerization and Orchestration**: Models were containerized using Docker and managed with Kubernetes for automatic scaling, load balancing, and zero-downtime updates across different cloud environments.
* **Cloud Deployment & Hybrid Cloud**: The models were deployed on a hybrid cloud architecture using AWS SageMaker for machine learning workflows and Google Cloud AI for redundancy. This ensured scalability, reliability, and cost efficiency. The infrastructure also leveraged AWS Lambda for serverless model triggers and Fargate for seamless container management.
* **API Development & Microservices Architecture**: Deployed RESTful and GraphQL APIs using FastAPI and gRPC to allow efficient, real-time interaction between systems, improving communication speed. The microservices architecture enabled independent scaling and testing of individual components, increasing system resilience.
* **CI/CD Pipelines**: Fully automated deployment pipelines were built using GitLab CI/CD and Jenkins to ensure continuous integration, automated testing, and deployment with minimal downtime. ArgoCD was implemented for Kubernetes-native continuous delivery, allowing full automation of the deployment process.

**Data Pipeline Automation and Monitoring:**

* **Data Pipeline Automation**: Leveraged Prefect and Airflow for orchestrating complex machine learning workflows, including data ingestion, feature engineering, and model retraining. Feature Store platforms like Tecton were used to automate and version control features, ensuring consistency between training and inference.
* **Real-time Model Monitoring**: Integrated Datadog, Prometheus, and Grafana to monitor system health, model drift, and accuracy in real time. Custom dashboards tracked key metrics like latency, resource utilization, and model prediction confidence scores, allowing for rapid issue detection.
* **Advanced Logging and Auditing**: Used ElasticSearch combined with Fluentd for centralized logging, ensuring comprehensive monitoring, troubleshooting, and auditing. OpenTelemetry was implemented for distributed tracing, providing end-to-end visibility of requests across microservices.

**Model Lifecycle Management & Retraining:**

* **Model Lifecycle Management**: Used MLflow and Kubeflow for tracking experiment runs, models, and datasets. DVC (Data Version Control) was employed to manage data and model versions, ensuring reproducibility and traceability across different environments.
* **A/B and Multi-Armed Bandit Testing**: Deployed Optimizely for controlled A/B tests and Azure Machine Learning’s Bandit algorithms to automatically optimize model deployment by directing more traffic to the best-performing models.
* **Automated Model Retraining and Governance**: Kubeflow Pipelines or Tecton was set up to automatically retrain models based on model performance or data drift triggers. ZenML provided an additional layer of model governance, ensuring all models comply with regulatory standards and performance benchmarks.

**Security, Governance, and Compliance:**

* **Data Security**: Employed AWS KMS and Google Cloud Key Management for encryption of data at rest and in transit, while HashiCorp Vault was integrated for managing sensitive information like API keys and database credentials.
* **Compliance and Auditing**: Integrated AWS Audit Manager and SOC2-compliant frameworks to monitor and ensure compliance with data privacy laws such as GDPR and CCPA.
* **Access Control & Authentication**: Implemented OAuth2.0 and OpenID Connect for secure access control and authentication, ensuring only authorized users can interact with the models and systems.

**Performance Optimization and Cost Management:**

* **Auto-scaling and Cost Efficiency**: Implemented KEDA (Kubernetes Event-driven Autoscaler) to optimize the scaling of models based on event triggers, reducing idle resource costs. Spot instances and preemptible VMs were utilized to significantly reduce cloud infrastructure costs.
* **Hardware Optimization**: Models were fine-tuned for deployment on NVIDIA Triton Inference Server using TensorRT and CUDA for real-time, low-latency inference, especially for edge devices and large-scale applications.

**Collaboration and Model Interpretability:**

* **Model Interpretability:** Integrated SHAP and LIME into the deployment pipeline to provide interpretability for model predictions. This allowed stakeholders to understand the factors driving key business predictions, increasing trust in the models.
* **Collaboration Tools:** Leveraged Weights & Biases for experiment tracking and collaboration among data science teams, making the entire machine learning lifecycle visible and transparent to business stakeholders.

**7. Future-Proofing and Continuous Improvement:**

* **Edge Computing Integration:** Deployed lightweight models on Edge AI platforms using NVIDIA Jetson and AWS IoT Greengrass, enabling low-latency, on-device decision-making.
* AI **Governance & Fairness Auditing:** Fairlearn and Aequitas were used to regularly audit models for bias, ensuring fairness across different demographic groups.
* **Active Learning for Improved Efficiency:** Implemented active learning techniques to continuously improve models by selectively retraining on new, informative data points.