17-634: Model Deployment & MLOPS Concerns

Xuchao Zhou

Model Deployment Strategy

To deploy the K-Nearest Neighbors (KNN) model for predicting airline fares, an API-based system will be developed, which will allow users to input relevant travel details (e.g. year, quarter, route, and distance) and receive a real-time fare prediction. The model will be hosted on a cloud platform such as AWS or Google Cloud, which ensures scalability and low-latency inference (https://cloud.google.com/vertex-ai/docs). Given that KNN is a lazy learner and does not require retraining, the nearest neighbors will be pre-computed for efficiency, significantly reducing lookup time at inference (https://umap-learn.readthedocs.io/en/latest/precomputed_k-nn.html). The deployment pipeline will consist of the following steps:

1. **Preprocessing Pipeline** – Standardizes numerical features (e.g., nsmiles, passengers_log) and applies encoding to categorical variables (quarter, route).
2. **Model Inference API** – An endpoint that accepts input features, retrieves the k-nearest neighbors, and returns the predicted fare.
3. **Database Integration** – Store historical queries to enable performance monitoring and potential retraining.
4. **Web Interface / Mobile App** – A user-friendly front-end for travelers and airline staff to retrieve fare predictions.

MLOps Considerations: Model Monitoring & Maintenance

Since KNN relies on historical data, continuous monitoring is required to ensure the model remains relevant. The following MLOps practices will be implemented:

1. **Performance Monitoring:** Regularly evaluate RMSE and R-squared values using new query data. If model performance deteriorates, potential reasons (e.g., market shifts, increased outliers) will be investigated.
2. **Data Drift Detection:** If new data deviates significantly from the training dataset (e.g., sudden fare fluctuations due to external events), an alert system will notify data engineers. A periodic re-evaluation of the dataset will determine if reprocessing or feature updates are required (https://www.evidentlyai.com/ml-in-production/data-drift#:~:text=Data%20drift%20refers%20to%20changes,system%20operates%20under%20familiar%20conditions.).
3. **Automated Model Updates:** While KNN does not require retraining, model efficiency can be improved by optimizing n_neighbors dynamically based on performance metrics. Additionally, periodic updates to the dataset (e.g., quarterly) will ensure the model reflects recent fare trends.

4. **Latency Optimization:** Precompute nearest neighbors and cache frequently requested predictions to reduce response time, ensuring real-time performance for end users.

Security Considerations & Mitigation Strategies

Deploying a machine learning model introduces security risks that must be addressed. The primary concerns and mitigation strategies include:

1. **Data Privacy & Protection:** Since the model relies on historical fare data, personal information should not be included in the dataset. Any user queries should be anonymized, and data stored in compliance with GDPR and CCPA regulations (https://www.ama.org/pages/california-consumer-privacy-protection-act-what-you-need-to-know/).
2. **Model Integrity Attacks:** To prevent adversarial inputs or data poisoning attacks, input validation will be implemented, ensuring only valid values for features (e.g., nsmiles, passengers_log) are accepted. Rate limiting will be enforced to avoid automated abuse (https://ubiops.com/as-mlops-hits-maturity-its-time-to-consider-cybersecurity/).
3. **API Security:** The inference API will require authentication to prevent unauthorized access. Security best practices such as token-based authentication (OAuth 2.0, https://oauth.net/2/) and HTTPS encryption will be implemented.
4. **Bias & Fairness:** Since fare pricing can be influenced by various socioeconomic factors, continuous fairness evaluations will be conducted to ensure no unintentional bias is embedded in the predictions. Regular audits will be conducted to check for disproportionate errors across different routes or travel periods.

Conclusion

By implementing these deployment, monitoring, and security strategies, the KNN model can provide accurate and efficient fare predictions while maintaining security, reliability, and fairness. Future enhancements may include integrating additional pricing factors (e.g., airline promotions, booking class) and experimenting with more complex models (e.g., Convolutional Neural Network) for performance comparison.