MLOps Group No: 92

Group Members Names:

1. PEYALA SAMARASIMHA REDDY - 2023AA05072
2. PEGALLAPATI SAI MAHARSHI - 2023AA05924
3. ANIRUDDHA DILIP MADURWAR - 2023AA05982
4. K VAMSIKRISHNA - 2023AA05209

M5: Final Deliverables

In this assignment, we worked on a structured MLOps workflow across four key tasks. The goal was to ensure a robust, interpretable, and well-monitored machine learning pipeline. For easy understanding and showcase our results, we implemented each task separately but the same flow is maintained as the MLOps pipeline. We explored many tools in all tasks.

M1: Exploratory Data Analysis (EDA)

Work completed is, to automate EDA, we used **YData Profiling** and **Sweetviz**. These tools helped generate detailed reports, visualizing key insights like class distribution, missing values, feature correlations, and potential outliers. **As fashion MNIST is image data it only shows features.**

Justification for Tool Selection

- **YData Profiling**: Provides comprehensive dataset summaries automatically.
- **Sweetviz**: Offers interactive visualizations, making it easier to compare training and test distributions.

MLOps Best Practices

Automating EDA ensures consistency and saves time. It helps in identifying potential data quality issues early, making the pipeline more reliable and reproducible.

M2: Feature Engineering & Explainability

Work completed is, feature engineering included **PCA for dimensionality reduction** and **Explainable AI methods (SHAP, LIME, InterpretML)** to understand feature importance. **As fashion MNIST is image data it only shows features,** we only analysed those visualisations, based on relation between each features and distributions etc and refined feature engineering code based on observations.

- **PCA** was applied to reduce redundant dimensions.
- **SHAP** helped identify the most influential features in predictions.
- **LIME** provided local explainability for model behavior.

- **InterpretML**'s Explainable Boosting Machine (EBM) guided feature selection.

Justification for Tool Selection

- **SHAP**: Provides a global and local understanding of feature importance.
- **LIME**: Helps in explaining individual predictions, which is crucial for trust in AI models.
- **InterpretML (EBM)**: Allows for inherently interpretable modeling.

MLOps Best Practices

Explainability ensures transparency, helping stakeholders trust the model. Feature selection based on interpretability leads to more robust, fair, and generalizable models.

M3: Model Selection & Hyperparameter Optimization

Work completed is, we used **Auto-sklearn** for automated model selection and hyperparameter tuning. The best-performing model was a **Support Vector Classifier (SVC) with RBF kernel**, optimized using Bayesian search. The final model achieved 0.849 accuracy on the test set with C = 113.11, Gamma = 0.000117 and Tol = 0.00846

Justification for Tool Selection

- **Auto-sklearn**: Efficient AutoML framework that automates model selection and hyperparameter tuning.
- **Bayesian Optimization**: Ensures an optimal search strategy for tuning parameters.

MLOps Best Practices

AutoML reduces manual effort in model selection, improving reproducibility. Automated hyperparameter tuning ensures optimal performance without manual guesswork.

M4: Model Monitoring & Performance Tracking

Work completed is, we implemented model tracking and monitoring using **MLflow**, **Prometheus** and **FastAPI** for real-time performance tracking, we passed input data for 10 different inputs to end point url and it predicts the fashion mnist data class number. **Kolmogorov-Smirnov (KS) test** was used for drift detection. We generated the synthetic data to show drift as the original data not have drift.

Justification for Tool Selection

- **MLflow**: Provides versioning and experiment tracking.
- **Prometheus**: Enables real-time performance monitoring.
- **FastAPI**: Allows easy API deployment for serving models.
- **KS Test**: Detects distribution shifts, indicating when retraining is needed.

MLOps Best Practices

Model monitoring ensures long-term reliability. Drift detection helps identify when models need retraining, preventing degradation in real-world performance.