Asok Kalidass Kalisamy

https://www.linkedin.com/in/asokkalidass/

Email : asok.kalidass@gmail.com
Mobile : +1-902-818-2603

Education

- **Dalhousie University** Halifax, NS
  Master of Science in Applied Computer Science; GPA: 3.84 *2017 – 2018*
  (NLP, Data Analytics & Data Warehousing)
- **Anna University** Coimbatore, India
  Bachelor of Engineering in Mechatronics; CGPA: (8.88/10.0) *2007 – 2011*
  (Robotics, Embedded Systems & Virtual Instrumentation)

Programming Skills

- **Languages**: Python 3x, JavaScript, TypeScript, HTML, SCSS, SQL, N0SQL, C#, Core Java.
- **Technologies**: Sklearn, Tensorflow 2, Keras, ONNX, Optuna, MLflow, Feast, AirFlow, Azure, AWS, Docker, Kubernetes, GraphQL, MongoDB, PyTest, Typer, Seaborn, DVC, Great Expectations, Flake8, Black, Locust, Alibi-detect, Iter8, Pydantic, Prometheus, Grafana, Mkdocs, GitHub Actions, Git, KFServing, Anaconda.
- **Web Framework**: FastAPI, Flask, Streamlit, React, Angular JS, ASP.Net MVC.

Experience

- **Provincial Health Services Authority** Vancouver, BC
  Senior Programmer *May 2020 - Present*
  - Created various Data-Sync services (strategy pattern) for data-integration using Hangfire.
  - Created the PO header module for the Centralized Back Order Management app.
  - Enhanced Complex data intensive master item management app using CQRS pattern.
  - Created TPL based Restful APIs with code first EF (Generic Repo Pattern) at the backend (multi-tiered pattern)
  - Designed Expression-based generic search extension function. (.Net Reflection and Expression Trees)
  - Created various Power BI Reports, SQL Scripting and wrote logging and authentication .net core pipelines.
  - Designed the react boilerplate application that has a generic API layer, Reduce & multi-document architecture.
  - Developed multiple generic React hooks components (Grid, Layouts, Lists, Dynamic Forms, and Controls)
  - Written Unit test for UI components (Jest & Enzyme) and Business Layer (MOQ and XUnit)
  - Performed root cause analysis for defects and provided resolution.
  - Actively participated in design discussion and peer code review.
- **Snipp Interactive** Vancouver, BC
  Full Stack Developer *Sep 2018 - Sep 2019*
  - Created a rewards dashboard using MVC, repository pattern (generic), and UnitOfWork in an N-tiered pattern.
  - Leveraged parallel programming for file processing and MMS queue using producer & consumer pattern that reduces the job execution time by 60%.
  - Developed various request pipelines. (Logging and JWT Auth)
  - Created an MMS testing tool for the QA team that eliminates the use of a paid Perfecto subscription.
  - Worked with the business for requirement clarifications.
- **Cognizant** Chennai, IN
  Associate *Aug 2011 and Dec 2016*
  - Implemented AGILE development methodology and provided estimates, and participated in weekly grooming meetings, daily & weekly status reports to the client on the progress of the tasks.
  - Written a load balancing logic to call the web service to request for criminal history.
  - Participated in various Client meetings with the team and the Project manager to discuss improvements /amendments in the development of the section.
  - Developed a standalone SSN Encryption/Decryption utility to assist the support team.
  - Constructed the generic file-based logging mechanism using log4net.

- Developed an SSIS package to refresh the prod data in a test environment to ease our support activities.
- Created several POC's and Design documents for R & D activities.
- Involved in application development using C#, SQL Database, LINQ, WCF, Entity Framework, and Web API.
- Developed a JQGrid based complex dashboard with a subgrid that supports CRUD operations.
- Developed a fail-over mechanism for the Informix server to eliminate windows registries.
- Created a common architecture to remediate the group of applications that resulted in 50% increased productivity.

Projects

- **Fraud Detection System**
  https://github.com/asok-mirror/Real_Time_Fraud_Detection_System_MLOps
  - **Outline**: Build a real-time fraudulent transaction detection system that classifies the transaction based on the PCA transformed European customer transaction details. This end-to-end application is based on the google's MLOps architecture pattern.
  - **Airflow Pipelines (dataops & mlops)**: Created a Data pipeline that prepares the data-set for modeling, then triggers the mlops pipeline and serves the model for serving if it's efficient than the previous model.
  - **Feature Store**: Used Feast based store as a single point of truth for training and prediction flows. Integrated the incremental data refresh to the online store in the dataops pipeline.
  - **Hyperparameter Optimization**: Used Optuna for hyperparameter optimization, captured the study details in each trials using MLFlow, and persisted the best study trials and prams as json.
  - **Testing**: Used Great Expectations to valid the data set on statistical front and created checkpoints to validate it during every data refresh through dataops pipeline. Incorporated Locust to simulate the production traffic.
  - **Web**: Exposed the serving model using FastAPI along with metrics endpoint and performed API schema validation using Pydantic.
  - **Monitoring & Alert System**: Developed a Prometheus-based solution for continuous model monitoring and trigger alerts if model accuracy decreases over time. Wrote custom Grafana dashboard for real-time monitoring of various metrics.
  - **Model Decay**: Used alibi-detect and pickled the reference planes and then used it for outlier and drift detection. Explored the A/B testing through iter8 and deployed the solution using Kubernetes.
- **Customer Segmentation**
  https://github.com/asok-mirror/Customer_Data_Segmentation_MLOps_DVC
  - **Outline**: Created market basket analysis system to cluster the shoppers based on their shopping pattern and spending score for targeted marketing.
  - **Data Pipelines, WorkFlow & Versioning**: Developed stages to capture, process, train & evaluate and log best model, which in turn run as an automated workflow via DVC.
  - **ML Experiments & Evaluation**: Created Db & artifacts store to capture the model runs with various hyper-parameters in MLFlow and then developed the best model serving pipeline to persist the outcome of the experiments and serve it to the model registry. Performed hyper-parameter tuning to select the optimal cluster using the Elbow method & Silhouette score.
  - **Web**: Developed flask-based python application to serve the front end to predict the new data cluster.
  - **Container & Cloud**: Created docker-compose to serve the MLflow experiments and the web as docker images and then developed CI-CD workflows to deploy them to Azure container instances.
- **Car Price Estimator**
  https://github.com/asok-mirror/Car_Price_Prediction_Regression
  - **Outline**: Developed a regression-based model to determine the value of a used car based on the historic car sales from the Cardecko platform. The application has Flask based user interface to predict car value.
  - **Data Pre-processing**: Used label-based encoding for categorical features, applied group by to collate the data by car model, and used the average to fill the empty values.
  - **Model Search & Hyperparameter optimization**: Constructed model params json and applied it to the sklearn's RandomizedSearchCV to find the best model and its parameters.
  - **Model Evaluation**: Wrote querying logic to pickle the best model from the ML Experiments based on the adjusted R^2 metrics.
  - **Web & Cloud**: Developed Flask Dashboard for new price estimation and auto deployed the app to Azure app service using GitHub actions.