Artificial Intelligence

Case Studies

TresVista

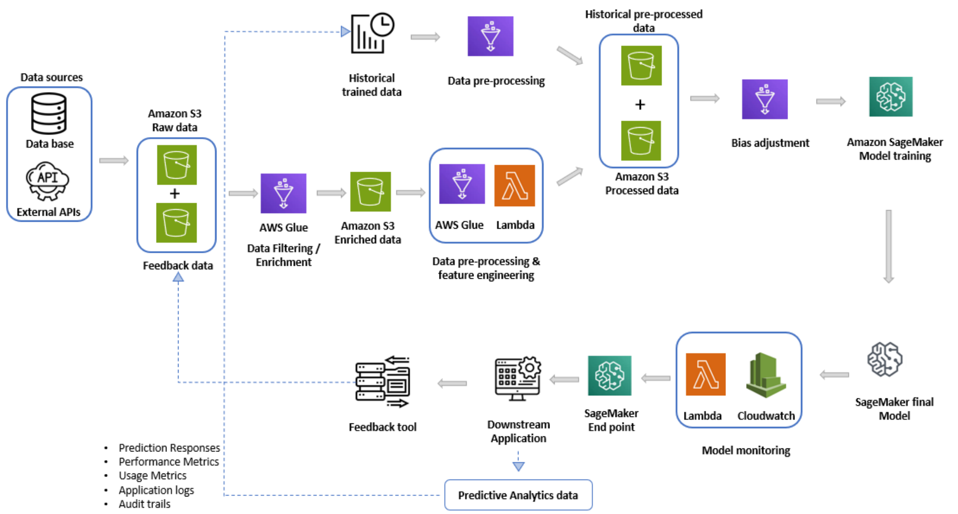
Catalyzing Your Impact

Artificial Intelligence

Credit Risk Scoring using Machine Learning

Request and Guidelines Provided

- With financial landscapes evolving, traditional models falter at predicting defaults, adhering to newer regulations, and capturing the nuances of borrower behavior. We proposed an AI-driven framework to revolutionize credit risk assessment, aiming to significantly reduce default rates and ensure comprehensive regulatory compliance
- Utilized a dual approach by integrating traditional financial histories with alternative data (e.g., online behavior, transaction patterns), enriching the predictive power of our models



Methodology and Final Deliverable

- Raw data is stored in Amazon S3 buckets which is then passed to Data filtering/ Enrichment layer along with feedback data. Utilizing AWS Glue we clean, normalize, and transform the data
- AWS Lambda functions are employed to generate new features that improve model predictive power, based on domain knowledge and data analysis insights
- ML models are developed and trained using Amazon SageMaker. Models are evaluated rigorously using performance metrics. The chosen model is deployed as a SageMaker endpoint, allowing real-time risk assessments through API calls
- AWS Lambda and Amazon CloudWatch is used to monitor the model's performance over time, automatically triggering re-training processes with updated data to ensure the model remains accurate as new data patterns emerge
- Credit Risk Scoring API: A fully managed, scalable API endpoint for real-time credit risk assessments, hosted on Amazon SageMaker

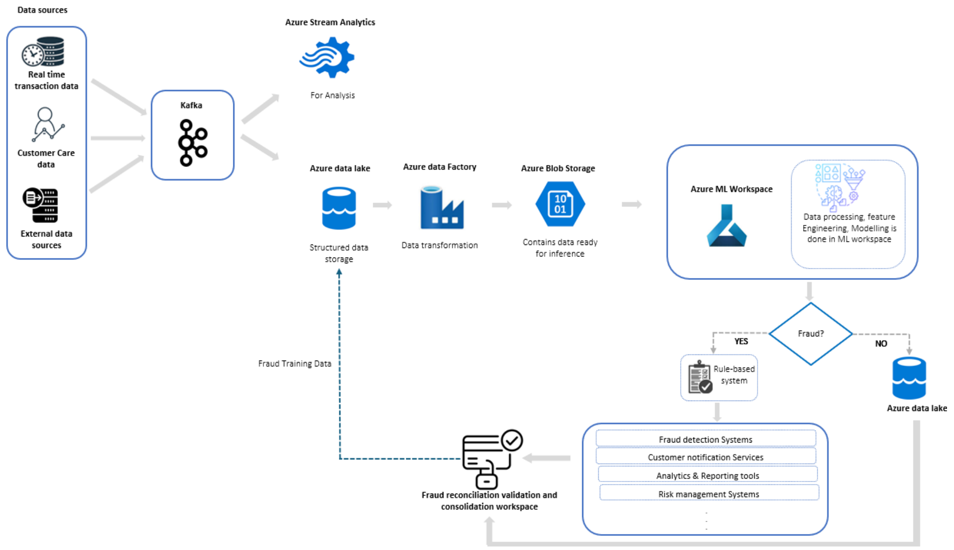
Tools/Technology used: AWS services(Amazon S3,AWS Glue, Lambda, SageMaker, CloudWatch), Python

Machine Learning

Fraud Analytics using Machine Learning

Request and Guidelines Provided

- In an era of digital transactions and sophisticated cyber threats, traditional fraud detection systems are increasingly unable to keep pace with the complexity and volume of modern fraud. Deployed an advanced AI framework for fraud analytics, significantly reducing fraud early on, enhancing operational efficiency, and increasing customer trust and satisfaction
- Employed a multi-dimensional approach that combines real-time transaction data, customer care data, and external data sources (such as digital footprints and public records) to enrich the analysis. This strategy ensures a holistic view of customer behavior, enhancing the accuracy of fraud detection models



Methodology and Final Deliverable

- Ingested and streamed real-time transaction data, customer care data, and external data sources securely through Apache Kafka
- Processed and analyzed streamed data using Azure Stream Analytics for real-time insights
- Stored and organized structured data within Azure Data Lake, providing a robust foundation for large-scale data storage. Leveraged Azure Data Factory for orchestrating data transformation processes
- Utilized Azure Blob Storage to house data that is ready for inference, allowing for scalable and on-demand access to data inputs for the model. Conducted data processing, feature engineering, and machine learning modeling within the Azure ML Workspace
- The ML model predicts whether a transaction is fraudulent and is processed into the rule-based system for initial decision-making, augmenting machine learning models with predefined logic to enhance prediction accuracy and further processed into Fraud Reconciliation Validation and Consolidation workspace along with Non-fraud transactions, which are identified
- Facilitated feedback loops for continuous model improvement with a feedback tool capturing data back into the Azure ecosystem

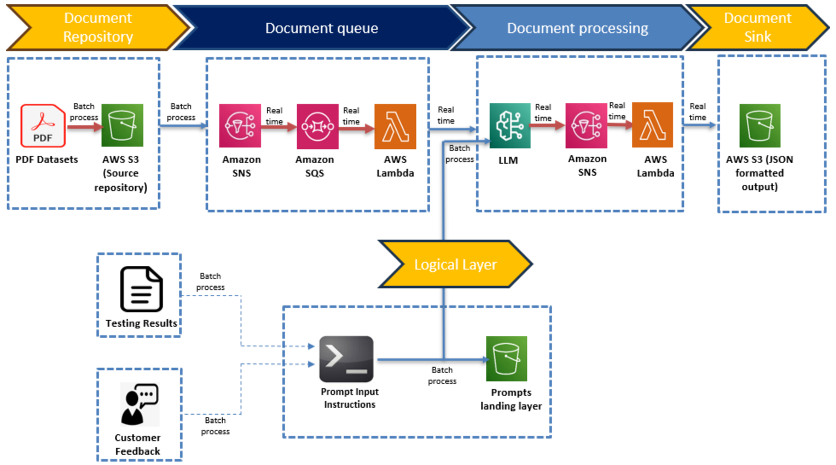
Tools/Technology used: Azure Services(Azure data lake, Data Factory, Blob Storage, ML Workspace), Python

Artificial Intelligence

Entity Extraction using Gen AI

Request and Guidelines Provided

- Client: Private Equity Firm
- Currently, entity data extraction is done manually, where the necessary entities are extracted from the source file, and the details are captured in the Excel file and finally converted into a JSON file
- As it involves more time and is error-prone, we implemented the automation process to classify the valid files, extract the necessary entities, and create the JSON file where the end-user application would consume it



Methodology and Final Deliverable

- AWS services were leveraged to establish an automated entity extraction pipeline, enabling the extraction of data from source files upon their upload to S3, finally producing the entity-extracted results in JSON format
  - Once we receive the notification of files arrival in S3, we read the files sequentially and classify the files using Amazon Textract
  - After the classification, we process the valid files into the LLM (Anthropic Claude 2) to extract the necessary entities based on the provided final output structure
  - During fine-tuning, we used prompt engineering to have the valid prompts processed to LLM to avoid the hallucinated results being generated
  - We leverage the services of SQS, SNS, Lambda, CloudWatch, etc.. as part of the pipeline to have seamless integration from the source to the output being generated
  - The final JSON output is converted to CSV equivalent files and format to MS Excel reporting standards and shared over E-mails

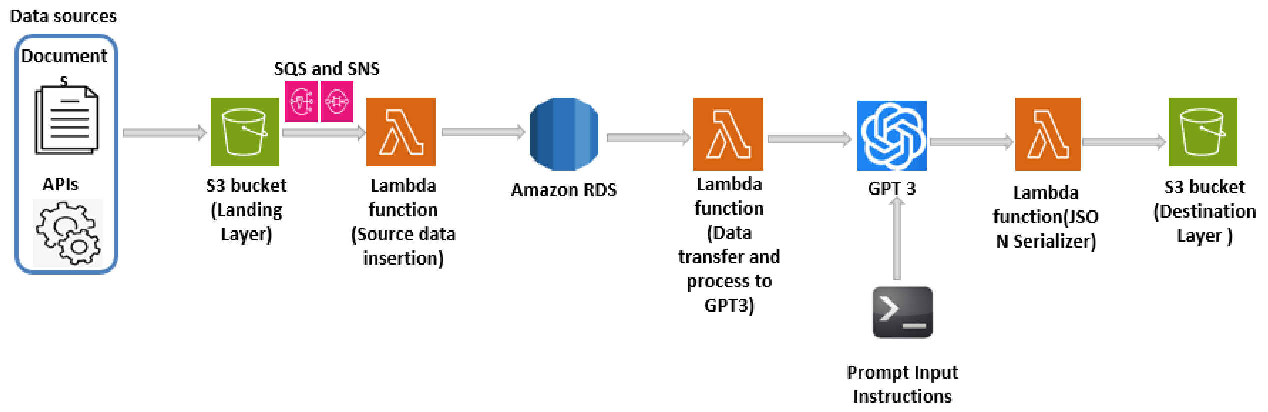
Tools/Technology used: AWS services, Python, LLM (Anthropic Claude 2)

Artificial Intelligence

Sentiment Analysis using Gen AI

Request and Guidelines Provided

- Client: Private Limited Firm
- As there was a decrease in the customer turnout ratio to customers visiting the website, we suggested identifying the sentiment of the customers who visited the website using the customer activity data
- Consolidate the data across multiple sources from different geographical locations to understand the sentimental analysis of the customer and enrich the business functionalities in those areas



Methodology and Final Deliverable

- The data preparation process involves splitting of datasets into training and testing based on the homogeneity to accustom different types of data
  - The data is stored in S3 which acts as a data lake and then it is processed into Amazon RDS (Maria DB) to store the data in row-column format through the AWS Lambda function
  - Once the data gets ingested into RDS then the lambda function is invoked to ingest the data into GPT 3 through Open AI’s API
  - Further prompt engineering is implemented so the prompts are passed into the LLM to extract the sentiments of the customer from the data
  - Lambda function is leveraged to serialize the output which is given from the LLM into JSON format and further the JSON output is stored in the S3 destination layer

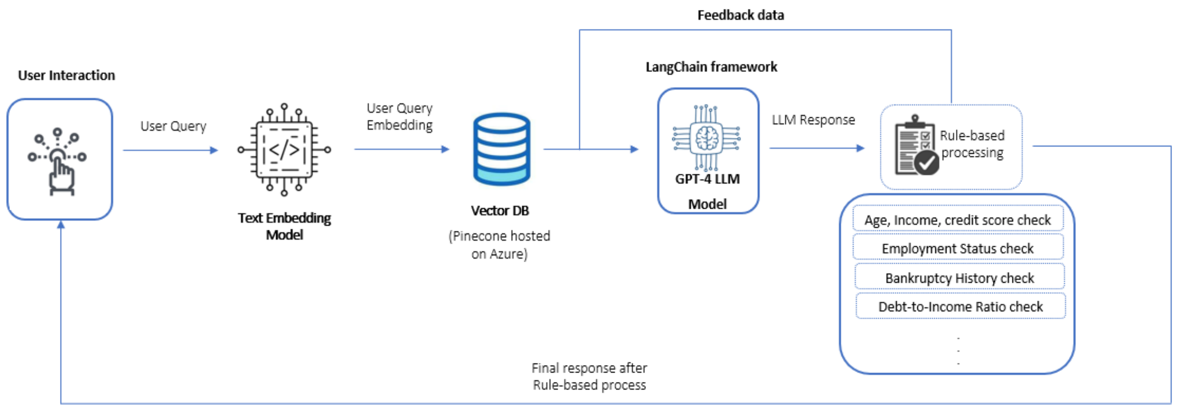
Tools/Technology used: AWS services, Python, Open AI (GPT -3)

Artificial Intelligence

Chat Bot developed using Langchain

Request and Guidelines Provided

- Client: Commercial Bank
- To expedite the process of loan application eligibility check, which involves multi-stage human involvement
- A chatbot is introduced to enhance the customer experience by providing real-time status updates, loan eligibility validation, and accurate and timely responses following predefined rules and guidelines consistently



Methodology and Final Deliverable

- Chatbot UI is provided in the Web application which allows the users to submit queries with little relevant information which is used for validation
  - The user-submitted queries are embedded using the Text embedding model for retrieving the relevant response
  - The user query embeddings are then submitted to Vector Database(Pinecone) which is hosted on Azure
  - Further it is processed to GPT-4 LLM model to generate the response according to the prompts submitted
  - The output is then validated against the Rule-based check mechanism based on different companies and the result is passed to the User
  - Once the response is sent to the end-user we have the Feedback mechanism from the user which helps us to fine-tune the LLMs and result in the most accurate information

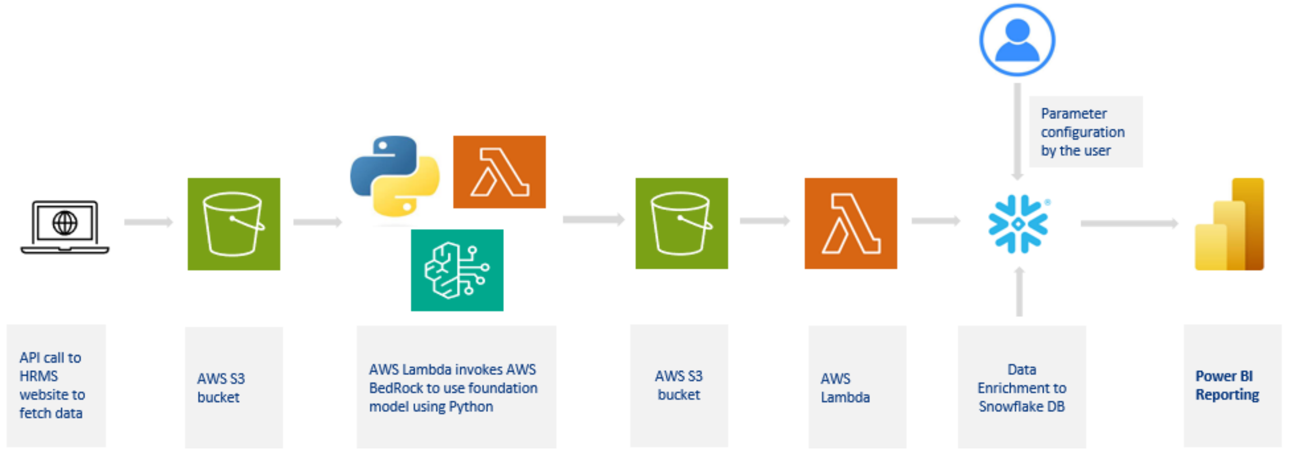
Tools/Technology used: Azure,Python, Langchain framework, LLM (GPT-4)

Artificial Intelligence

Profile screening using Gen AI

Request and Guidelines Provided

- Client: Private Firm
- The procedure of candidate screening for job application. The process involves data inputs from JAF (job application form), resume/CV, and candidate cover letter; it is a manual task with multi-stage human involvement with subject-matter know-how
- To save on human involvement and streamline the process, an LLM is introduced to help the recruiters with data consolidation, CV/resume data extraction and cover letter segregation. The output is a Power BI dashboard to visualize the data



Methodology and Final Deliverable

- The data flows through a hiring website (HRMS tool). The structured (employee details tables in CSV) and unstructured data (resume and cover letters) is stored in AWS S3 buckets. The data goes through an AWS pipeline and outputs relevant candidate's basis parameter selection (experience, background, education, skill-sets)
- Descriptive textual data is extracted from the CV and cover letters using LLM
- The data from CV is stored as key-value pairs in JSON format and cover letters are converted to single contextual paragraphs using prompt engineering to normalize the outputs
- The data is pushed to SnowFlake DBMS and pushed onto MS Power BI for a comprehensive dashboard
- The dashboard provides a comprehensive reporting platform to best present the set of candidates’ using slicers and interactive visuals basis the job role requirement

Tools/Technology used: AWS services, Python, LLM and Power BI

salessupport@tresvista.com | www.tresvista.com