|  |
| --- |
| Maruti Component Health Status  Solution Design Approach |

**High Level Defect Prediction Process**

1. Data Collection, Ingestion from various sources & Data Engineering
2. Primary Data Exploration
3. Clearing Issues, observations, Queries, pending items mentioned in attached Excel
4. Understanding / discussion on data - Primary key’s in various tables to understand uniqueness and foreign fields to join with tables.
5. Business Process mapping / Data Mapping to critical features for defect prediction
6. Data Exploration - with Selected key Features
7. Data Modelling –Multiple Iterations of Classification Algorithms
   1. We used Linear Learner
   2. We used XGBoost
   3. Also used HPO functionality of AWS platform – Hyper parameter optimization
8. Training the Model
9. Test Model / Model Accuracy

**Our Solution approach is multistep** –

1. Setup AWS Account
2. Setup AWS security
3. MFA Token for Root Account
4. Identity and Access management with defined roles and rights for AWS users
5. Virtual Private Cloud (a virtual private network) as per the required configuration
6. Setup S3 Bucket for creating Data Lake.
7. Configure pipeline to ingest data from IBM to Data Lake.
8. Create EMR Cluster to transform data to CSV.
9. Create and configure Lambda to store the data in hive partitions in S3.
10. Create Glue job to convert data to Parquet format.
11. Configure Athena to query parquet data in S3.
12. Create QuickSight dashboard with Athena as data set.
13. Used SageMaker to build/train model.
14. Done PCA for Dimension Reduction.
15. Modelling the data for existing patterns using various statistical / Machine Learning model e.g. Linear Learner, XGBoost etc.
16. Comparing the results for various models
17. Test Model / Model Accuracy
18. Choose one best model
19. Training the Model
20. Defect Prediction

**Learnings**

* Partitioning of data improved the data query performance.
* EMR Cluster calculation of nodes required for data preprocessing.
* Using parquet format gave many benefits like less space required for storing the data.
* Do weightage analysis for inclusion of more parameters and then apply combinations accordingly for secondary data elements in a model running to identify best fit parameters.
* Need to use bagging/boosting for improving the accuracy

**Test cases**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case No.** | **Description** | **Test Data** | **Expected Result** | **Actual Result** | **Status (pass/fail)** |
| #1 | Data should be successfully copied into the HDFS using shell script. | Compressed GZIP data | Copy in HDFS | Successfully Copied | Pass |
| #2 | All Data should be copied in HDFS | Compressed GZIP data | Size of data before and after copy must be the  same. | Equal Size | Pass |
| #3 | Cluster should be Automatically Terminate after Transformation. | Compressed GZIP Data | Must Terminate after Transformation | Successfully Terminated | Pass |
| #4 | Information of data must available before transformation in separate bucket. | Compressed GZIP Data | .Txt file of all files in compressed data must be  stored in separate bucket. | Successfully Stored | Pass |
| #5 | Script should not fail in between. | Compressed Zip Data | Script should be successfully run on master node | Ran Successfully | Pass |

Solution Summary - Training & Handover Notes

**About Challenge(s)**

Maruti wants to build a Health Monitoring System for various components/parts of car manufactured to prevent defective parts to pass through to customers. This will help to reduce cost of vehicle call backs and subsequent repair.

This includes ingestion of 20 TB data from different cloud providers to AWS cloud and performing preprocessing on 20 TB once and then 2 TB monthly data. After ingestion to do analytics on data and build machine learning models and storing data in S3.

Following were few technical challenges

* Selection of features for Modelling
* Dimensionality Reduction without loss of useful information
* Selection of hyperparameters for XGBoost
* Cost for data storage.
* Errors while transferring data from different cloud provider to AWS.
* Efficient Distribution of data to executors on worker nodes to process large volumes of data.

The main challenge was Ingestion of 20 TB data from different cloud providers to AWS cloud and performing preprocessing on 20 TB once and then 2 TB monthly data. After ingestion to do analytics on data and build machine learning models and storing data in S3.

**Proposed Solution**

* Data collected & ingested from various sources for storing semi-structured data.
* Transformed old csv and json data into parquet format and did partitioning.
* Running Amazon EMR on incremental data daily, partitioning and storing it in parquet format for preprocessing.
* PCA (Principle Component Analysis) is applied on data set for feature reduction and selection.
* After that Machine Learning algorithms were used to predict car component health status using modeling techniques like Linear Learner, XGBoost etc.
* Did hyper parameter tuning for XGBoost using SageMaker Hyperparameter optimization (HPO)
* Deployed the ML model using SageMaker endpoint.

**AWS Services used**

* Amazon SageMaker
* AWS Glue
* Amazon Athena
* AWS Lambda
* Amazon EMR
* Amazon Quick sight

**Solution Outcome**

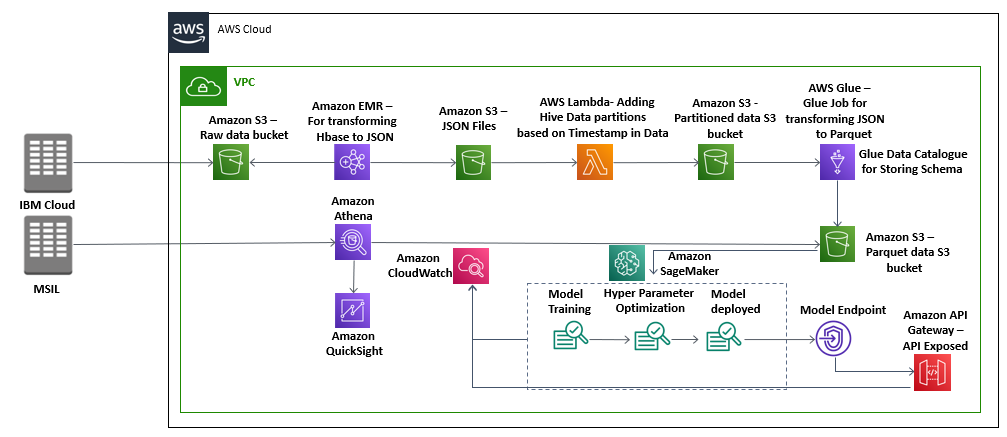
* XGBoost model created using high power GPU instances.
* XGBoost model can be inference using model API, which is deployed on Amazon SageMaker.
* Training Cost saving by using spot Instance for Training the model.
* A Status of car component can be fetched using API
* Pre-Processing of data without moving to a separate system, also allows to generate different types of insights for parameter selection and further preprocessing.
* Data lake allows us to scale to data of any size, without defining schemas and data structures.
* A dashboard which showed the status of all the TCU (Telematics Control Unit) on a specific day.

Increased profit due to fewer number of call backs rate for repair after sale

 Reduction in production time due to timely detection of defective parts

 Overall customer satisfaction increased which resulted in many intangible benefits e.g. helps to increase trust in brand which eventually leads to customer loyalty

**Architecture Diagram**



**How AWS services helped in building the model for sales Forecasting**

**Amazon SageMaker**

1. Amazon SageMaker is used to create and manage Jupyter notebooks that were used to prepare and process data and to train and deploy the machine learning models.
2. Amazon SageMaker High Power GPU Instance used for training of XGBoost Model.
3. Model is optimized using Amazon SageMaker Hyper Parameter Optimization Service followed by deploying through SageMaker built-int deployment service

**Amazon Athena to query the partitioned data for insights**

It is an interactive query service that made it easy to analyze the output parquet data stored in Amazon S3 using standard SQL. It is serverless, so there is no infrastructure to manage, simply point to your data in Amazon S3, define the schema, and start querying using standard SQL and the results were delivered within seconds.

**AWS Glue to load partitioned data**

It is a fully managed extract, transform, and load (ETL) service that made it easy for us to prepare and load the processed, partitioned data for analytics. One can create and run an ETL job by simply pointing AWS Glue to the data stored in S3 bucket, which is then immediately searchable, query-able, and available for ETL.

**AWS Lambda to handle the backend API calls**

It helped to initialize and validate the input and acted as the backend of the whole task. AWS Lambda lets us run code without provisioning or managing servers. Also, it helped to connect with various AWS API’s to acquire various insights from the inputs.

**Amazon S3 to store JSON and CSV raw documents and the output parquet files**

It is an object storage service that offers industry-leading scalability, data availability, security, and performance. In this solution it.

**Amazon EMR to transform telematics data**

It is a tool for big data processing and analysis, Amazon EMR is based on Apache Hadoop, a Java-based programming framework that supports the processing of large data sets in a distributed computing environment. Once the transformation of data is done then business analytics will be done using Athena and Quick Sight

**Amazon Quick sight for visualization through dashboard**

It is a fast, cloud-powered business intelligence service offering by AWS which here made it easy to deliver insights related to the output data like TCU per day etc.