Predicting Credit Card Defaults Using Machine Learning

Introduction and Objectives:
The goal of this project is to develop a ML model pipeline to predict credit card defaults using the Kaggle American Express (Amex) Default Prediction dataset. Using Aws services to accurately predict defaults can help financial institutions manage risk more effectively in real time and provide better customer service and reduce losses. By utilizing advanced machine learning techniques, this project aims to create a robust predictive model pipeline that can handle large-scale datasets and provide reliable predictions for use.

Data Source & Reason:
Dataset: Kaggle Amex Default Prediction dataset
Dataset Size: 5.99 million rows (20 GB +)
Features: 190 features (179 numeric, 11 Categorical)

1. Real world use case and very useful in finance industry.
2. Large Dataset with 5.99 million rows. Test dataset which has more rows (Future work).
3. To understand the scale of data for machine learning models.
4. Huge Feature Set with 190 Feature.
5. 47 Important features will be selected for model training.

EMR Cluster:
1. **Version**: EMR-7.1.0
2. **Application Bundle:** Hadoop 3.3.6, Hive 3.1.3, JupyterEnterpriseGateway 2.6.0, Livy 0.8.0, Spark 3.5.0
3. **Primary, Core, Task:** m4.large
4. **Core Size, Task Size:** [1,3]

Data Processing and Transformation Steps:
Data processing pipeline: using Spark, to read CSV files from the S3 bucket, transform the data, and save it back to S3 in Parquet format.

Functions:
1. get_destination_path: Constructs the destination path for the transformed data.
2. source_to_write: Reads the source CSV, Transformations renames columns to replace spaces with underscores, and writes the data to the specified S3 path in Parquet format.

Data Processing:
1. Reads CSV files from S3 using Spark.
2. Renames columns to ensure they don't contain spaces.
   Handling missing values, Normalization, one hot encoding based on the feature importance of the data (Future work)
3. Writes the transformed data back to S3 in Parquet format.

Execution:
1. Specifies S3 paths for source data and destination.

2. Processes two files (train_data.csv and train_labels.csv) by calling source_to_write for each, storing the transformed data in the specified destination.

This pipeline automates the ETL process, ensuring data consistency and efficiency, making it suitable for further analysis.

Learning: CSV vs Parquet format (Delta format (future work))

1. Parquet format is highly efficient in columnar storage format and significant compression which can lead to smaller file sizes.
2. Parquet is optimized for heavy operations and analytics.

Machine Learning Model Development:
Developing a machine learning model using Spark and Amazon Athena to predict credit card default users.

Pipeline Design:

1. Data Query and Preprocessing: Athena Query
2. Model Selection: Gradient Boosted Trees Classifier
3. Hyperparameter Tuning: Ray Tune (Future Work)

Athena Query:

1. Modified Athena query based on feature selection using feature importance.
2. Remove columns with high Null values.
3. Inner joined Customer ID from two tables for target feature.

Model Training Process, Evaluation and Results:

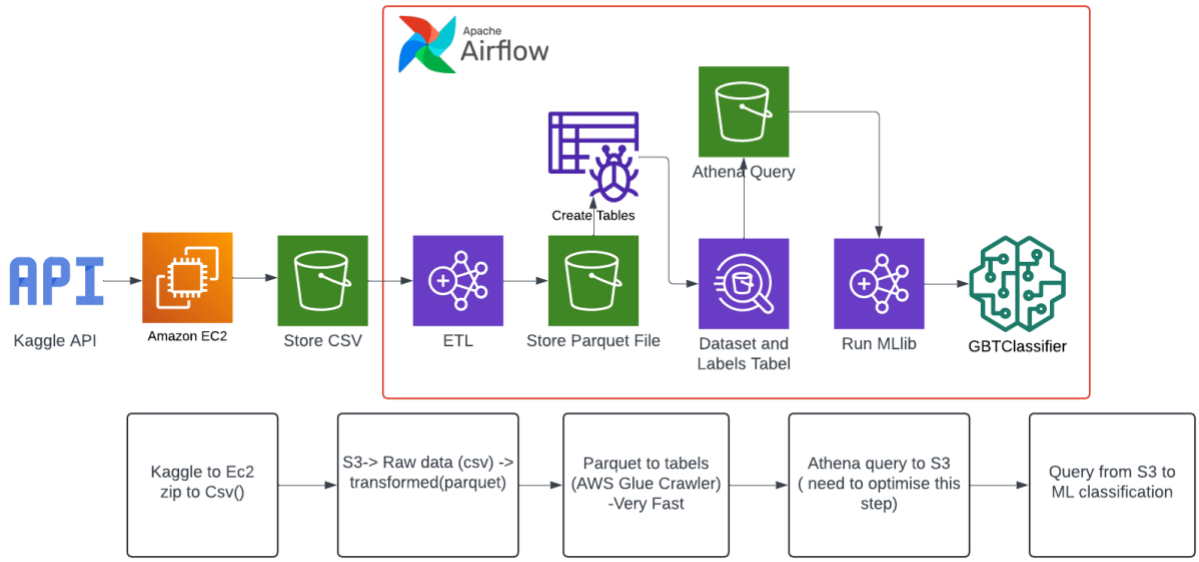
1. **Install Required Packages:** boto3, pandas, matplotlib, numpy.
2. **Import Libraries:** boto3 for AWS services, time for delays, matplotlib for plotting, pandas for data manipulation, and relevant PySpark modules.
3. **Athena Client Configuration:** Configure with AWS credentials and region information.
4. **Execute Athena Query:** Start and monitor query execution, specifying the S3 location for query output.
5. **Load Query Results:** Read the result data from S3 into a Spark DataFrame upon query completion.
6. **Data Preparation:**
    1. Drop customer_id column.
    2. Convert all remaining columns from string to double for numeric processing.
7. **Model Training and Evaluation:**
    1. Split the data into training and testing sets (80%-20%).
    2. Define feature columns and use VectorAssembler to assemble features.
    3. Define and train a GBTClassifier model.
    4. Evaluate model performance using Accuracy and AUC-ROC metrics.

Test Accuracy is 82.66% and AUC-ROC is 90.60%

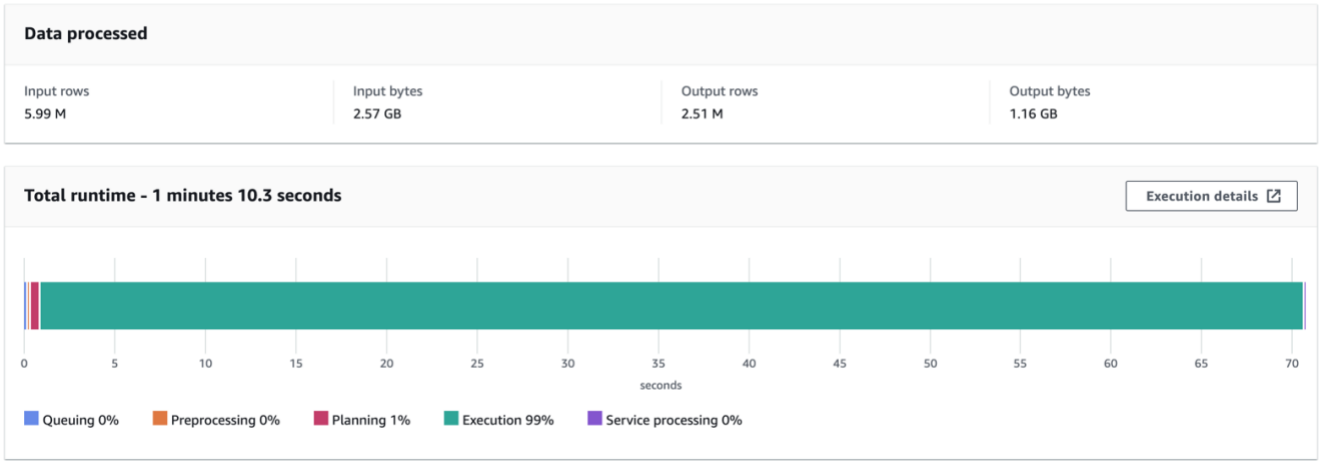
Summary:

Utilizing Spark for scalable data processing and model training, the procedure combines data preparation, querying, and machine learning model creation. A serverless querying solution for retrieving data from S3 is offered by Amazon Athena. After that, the Gradient Boosted Tree classifier is trained and assessed, yielding metrics for model evaluation including accuracy and AUC.

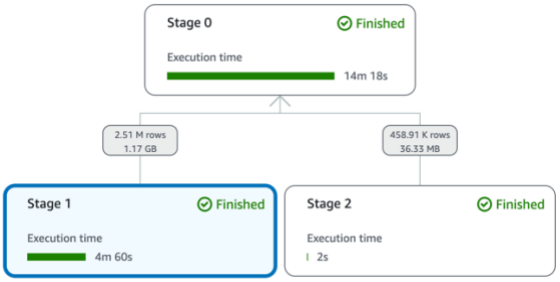
Visualizations and Insights



Athena Query Stats



Execution stages



Services Used: AWS EC2, S3, EMR, Glue, Athena, Spark

Challenges and Approach:
1. Data Download Issues: Initial attempt to download the dataset from Kaggle API to a local machine and upload to S3 failed. Solution: Used an EC2 instance to pull data from the API and push to S3.
2. Event Trigger with Lambda: Attempted to include Lambda to automatically trigger events once data is added to S3; understanding limitations is ongoing.
3. Athena Query to Spark DataFrame: Plan to use Athena.Client.get_query_execution to convert dict to Spark DataFrame (future work).

Benefits of Distributed Computing:
The massive dataset was managed by distributed computing, which allowed for parallel processing and greatly shortened the time needed for model training and data preprocessing. The scalability and efficiency of the model were improved by this method, which also made it possible to handle larger datasets than could be handled on a single system. Utilizing parquet versus a CSV file.

Conclusion and Future Work:
With the use of the Kaggle Amex Default Prediction dataset, this project effectively created a pipeline for machine learning model to forecast credit card defaults. We were able to efficiently handle a big dataset using spark and Athena for serverless querying. We have used Parquet file format for better performance in data processing and effectively train the model by utilizing distributed computing resources with spark ML library, notably AWS EMR, and a strong data processing pipeline with Spark. The accuracy and AUC-ROC measures of the Gradient Boosted Trees classifier showed strong performance, indicating the model's dependability and practicality.

1. Handling Missing values in better way.
2. Athena Query to Spark data frame
3. Using Airflow to automate the process to handle periodic update.
4. Deploying the classification model for real time inference.

These improvements will greatly improving financial industry's capacity in risk management by improving the current model and paving the way for a scalable, effective, and automated credit risk prediction system to avoid any high defaults rate.