Health Insurance Big Data Analytics – Requirements Specification Documents

1. Introduction:
2. Purpose: The purpose of this document is to outline the detailed requirements for developing a big data pipeline for the Health Insurance company. The company seeks to leverage big data tools to analyze competitor and customer data gathered via web scrapping and third-party sources. The objective is to understand customer behavior and optimize business strategies to increase revenue by offering personalized insurance products and rewards.
3. Intended Audience and Use:

This document is intended for use by:

* Data Engineers: To develop data ingestion, transformation, and cleaning pipeline
* Data Analysts: To analyze the clean data created by data engineers
* Data Scientists: To design, build, and validate analytical models
* Project Managers: To oversee milestones and delivery times
* QA Testers: To validate datasets and query results against business requirements

1. Product Scope: The product will enable better customer targeting through behavioral insights and revenue enhancement by identifying profitable segments by creating centralized data warehouse on AWS Redshift for analytics.
2. Definitions and Acronyms:

* S3: Simple Storage Service
* RedShift: AWS Data Warehouse Service
* ETL: Extract, Transform, Load
* PySpark: Python API for Apache Spark
* Jira: Project Tracking Tool
* Claim: Request of payment from the insurance provider

1. Overall Description:
2. User Needs:

* **Insurance Analysts:** need to identify high-claim diseases and profitable customer segments.
* Marketing **Teams**: require insights to offer targeted policy upgrades or discounts.
* Executives: need visibility into overall claim trends and revenue contributors.
* IT**/Engineering Teams:** need automated data pipelines for scalable processing.
* Analysts: need clean, structured data to run business specific queries.
* Developers: require a scalable architecture to support growing datasets and analytical demands.

1. Assumptions and Dependencies:

* AWS services (S3, Redshift, EMR) will be available and provisioned.
* Databricks community edition will be used for development/testing
* Data will be initially simulated or scraped from public and 3rd-party sources
* Dependencies: PySpark, AWS SDK, JDBC connectors, JIRA for task management, GitHub for version control

1. System Features and Requirements:
   1. Functional Requirements:

**Data Ingestion**:

* Upload raw datasets to S3.
* Load data into Databricks using Spark.

**Data Cleaning & Validation**:

* Detect and replace null values with ‘NA’.
* Remove duplicates across all datasets.
* Clean and validate: Patients, Subscribers, Claims, Group\_Subgroup.

**ETL Pipeline**:

* Transform cleaned data and load it into AWS Redshift.
* Create separate tables for analytical use cases under schema project\_output.

**Analytics Use Cases**:

* Execute queries to answer the following business questions:
  + Disease with max claims
  + Subscribers under age 30 with subgroups
  + Groups with max subgroups
  + Hospital with most patients
  + Subgroups with most subscriptions
  + Total rejected claims
  + City with most claims
  + Policy group preference: Govt vs Private
  + Average monthly premium
  + Most profitable group
  + Cancer patients under 18
  + Cashless patients with charges ≥ 50,000
  + Female patients over 40 with knee surgery

**Sprint Management (JIRA)**:

* Week 1: Requirement gathering, SRS writing, solution design
* Week 2: Code implementation, testing, visualization
* Create JIRA user stories for each use case, task, and test case

## External Interface Requirements:

### User: Databricks Notebooks for processing and visualization

### Hardware: AWS EMR for compute, Redshift for data warehouse

### Software: PySpark, AS SDK, Redshift, JDBC, S3

### Communications: GitHub

## System Features:

* + 1. Modular Data Cleansing: Null checks, replacements, type casting, deduplication
    2. PySpark ETL Pipelines: Read, Clean, Transform, Load into Redshift
    3. Automated Query Module: Generates Redshift result tables for each business use case
    4. Scalable Architecture: Designed to process growing customer datasets over time
    5. Sprint Execution Tracking: Managed in JIRA with clear tasks and test validations

## Nonfunctional Requirements:

### Performance requirements: ETL pipelines should handle millions of rows efficiently within scheduled batch windows.

### Safety requirements: Data masking for sensitive patient data during testing and Access restrictions to production S3 buckets and Redshift via IAM policies.

### Security requirements: Role-based access control (RBAC) for all cloud services and Data encryption in-transit and at-rest using AWS KMS.

### Usability requirements: Cleaned data should follow a consistent schema across sources and Output results can be easily used to query by non-engineers

### Scalability requirements: Pipelines should scale horizontally in EMR to accommodate future larger datasets and code modularity should allow integration of new data sources with minimal change.

|  |  |
| --- | --- |