Forecasting the Number of Marriages in Kazakhstan Using AWS

1. Project Overview

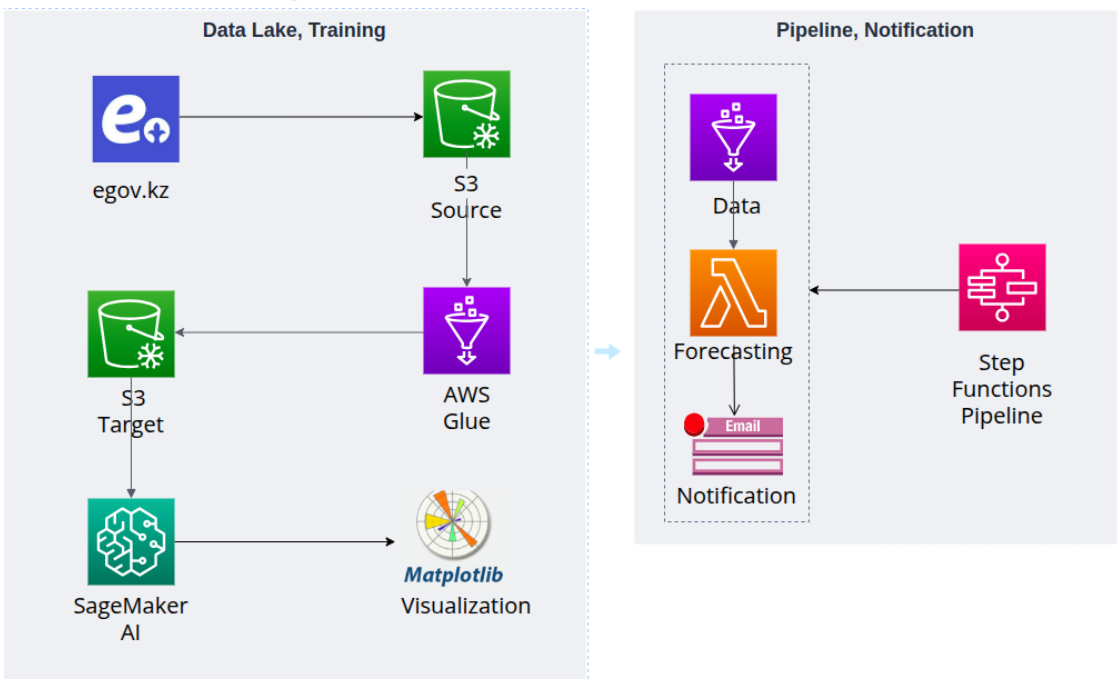
The goal of this project is to build a cloud-based data engineering and machine learning pipeline that predicts the number of marriages in Kazakhstan using historical data. The system is deployed using various AWS services to ensure scalability, modularity, and automation.

2. Architecture Components

The architecture is composed of the following components:

- **Amazon S3**: Used as a centralized data lake for storing raw and processed CSV datasets, as well as output predictions.
- **AWS Glue**: Performs the ETL (Extract, Transform, Load) job to clean and prepare the dataset for analysis.
- **Amazon Athena**: Executes SQL queries on top of the S3-stored data, offering fast exploratory data analysis and transformation.
- **AWS Lambda**: Triggers model inference on demand and manages communication with an EC2-hosted model server.
- **Amazon EC2**: Hosts a Flask application serving an LSTM model for time series forecasting.
- **AWS Step Functions**: Orchestrates the entire workflow from ETL to model inference to output storage.
- **Amazon SNS (optional)**: Notifies the user via email when a prediction is successfully completed.

AWS Architecture Diagram



- **Amazon S3** for data storage,
- **AWS Glue / Athena** for ETL and query processing,
- **Amazon SageMaker** for model training and forecasting,
- **AWS Step Functions** to orchestrate the pipeline,
- **Amazon SNS** for email notifications.

3. Data Processing

- Historical marriage data by region and year is stored in S3.
- AWS Glue reads and cleans this data at non-fields used.
- Athena can be used optionally to query and verify the structure of this data.

4. Forecasting Logic

- A pre-trained **LSTM model** is hosted on EC2 within a Flask app.
- It receives a region and dataset reference (S3 key) via an HTTP POST request.
- The model processes the input data and returns a numerical prediction for the next time step (e.g., the expected number of marriages in the next year).

5. AWS Lambda Function

- AWS Lambda serves as the intermediary between Step Functions and the EC2 model server.
- It:
    1. Accepts `region`, `dataset_key`, and `bucket_name` as input.
    2. Sends these to the EC2 `/predict` endpoint.
    3. Receives a forecast.
    4. Stores the prediction result as a `.json` file in S3.
- Lambda returns the S3 path to the output file and optionally forwards the result via email (if SNS is configured).

6. Workflow with Step Functions

The Step Functions state machine follows this sequence:

1. **ETL Step**: Runs a Glue job to prepare the dataset.
2. **Forecasting Step**: Invokes the Lambda function with parameters including the region and dataset location.

3. **Notification Step (optional)**: Sends a success message or the actual forecast to a user via Amazon SNS.

7. Example Output

Upon success, the system:

- Saves the prediction in S3 under a path like `s3://<bucket>/predictions/output_<uuid>.json`
- Returns a structured JSON output including the predicted value for the selected region.

8. Conclusion

This project demonstrates an end-to-end machine learning and data processing pipeline in the cloud using AWS services. It provides:

- Automated data flow and inference
- Flexible deployment (Flask model on EC2)
- Scalable and reusable architecture
- Integration of AI and serverless components