UTM

UNIVERSITI TEKNOLOGI MALAYSIA

SEMESTER I 2021/2022

TECHNOLOGY AND INFORMATION SYSTEMS

(SECP1513)

SECTION 01

PROJECT 1 - LOW FIDELITY PROTOTYPE PART 1

GROUP 2

GROUP MEMBERS:

No.	STUDENT NAME	MATRIC NUMBER
1	AIN BATRISYIA BINTI NORAZLAN	A21EC0009
2	LEE RONG XIAN	A21EC0043
3	MUHAMMAD AKMAL BIN SHAMSUL HAMIDI	A21EC0057
4	NOOR HANNANI SYAMIMI BINTI MOHD SUFFIAN	A21EC0104
5	SITI NURKAMILAH BINTI SAIFUL BAHARI	A21EC0131

CONTENT

1.0 INTRODUCTION

Prototype

A prototype is a conceptual model or sample created to test a proposed product or process. This step in developing a new concept or procedure is usually the first step towards the formalisation of an idea. Prototypes are often used in the design process to test ideas before they are mass-produced. Many designers carry out this practice in different disciplines. A prototype is a conceptual model that shows potential solutions to the problems discussed during the initial concept stage. Instead of relying on a supposed solution, a prototype allows designers to get feedback from real users quickly. When prototypes fail, designers send them back to the drawing process to refine or repeat the ideas based on the feedback gathered by users. This saves them from wasting time and money.

Low-fidelity prototype

A low-fidelity prototype is the first step in developing a website or application. It is usually carried out at an early stage to confirm that the concept is correct. It is also an excellent way to ask users about their product vision, a specified block, or flow. There are two categories of low-fidelity prototypes that are paper and digital. Paper prototyping shows how even the most detailed hand-drawn drawings can effectively express a product's interface and architecture. Prototyping editors are used to generating digital prototypes. Many tools purposely imitate the look of a sloppy drawing to emphasise the structure of the project rather than the appearance of the picture. This method aids in determining the comprehensibility of a given concept for consumers. These diagrams depict page chunks, transitions, and fundamental functionality.

Low fidelity prototype gives many advantages. Firstly, low-fidelity prototyping has a definite advantage in terms of cost. Other than that, this form of prototyping encourages collaboration. More people can participate in the design process because no special skills are required for lo-fi prototyping. Even non-designers can contribute to the idea generation process.

Our low-fidelity prototype project will be focused on the Fourth Industry Revolution (IR 4.0) technology, which is machine learning. The machine learning that we choose is tertiary analysis. From the viewpoint of an imaginary client, we will create a case study related to bioinformatics that can provide input and problem scenarios for the project.

SELECTION OF 4TH IR TECHNOLOGY

Machine Learning

Machine learning is a type of artificial intelligence (AI) that allows computers to develop without being precisely programmed. The purpose of machine learning is to create computer programs that are able to access input or information and learn independently.

The learning process starts with data observations, such as samples and direct experience, to identify patterns in data and make better decisions later based on the models provided. The main target is for computers to learn independently, without human intervention, and alter behaviour accordingly. Machine learning is being applied in many sectors and businesses right now. Examples include medical analysis, image processing, prognosis, categorisation, learning association, and many more.

Genomic data interpretation is a data-intensive science that requires a lot of processing capacity. This data is translated into biological insight utilising machine learning and high-performance computing using AWS and AWS Partner technologies. Computational efficiency at scale, reproducible data processing, data integration capabilities for pulling in multi-modal datasets, and public data for clinical annotation are all improved in a compliance-ready environment.

Cloud architecture

Cloud architecture is the process of combining several technological components to produce a cloud-like platform. A front-end platform, a back-end platform, a network, and a storage platform are cloud architecture components. Cloud computing helped businesses reduce or stop relying on an on-premises server, storage, and networking infrastructure.

The cloud computing architecture simplifies management by shifting servers, storage, and network equipment to the public cloud. As a result, on-premises hardware is no longer required, allowing organisations to reduce capital expenditures and replace them with a monthly IT budget. The shift from capital to operating expense is one of the critical reasons behind cloud computing's present popularity.

Tertiary Analysis

The analysis of all of a person's genes (the genome), and their interactions with one another and with the surrounding, is called genomics. Tertiary analysis in bioinformatics refers to the application of advanced computer science methodologies, algorithms, and tools to comprehend as well as analyse sequencing results retrieved from raw genetic data. The tertiary analysis is the final step in the bioinformatics pipeline, which starts with recognising unprocessed data and creating sequencing reads (primary analysis), followed by their arrangement (secondary analysis). The scope of tertiary bioinformatics analysis has expanded dramatically due to the increased availability of genetic material for study. Many technologies have been developed to aid this procedure. As a result, plenty of tools has been designed to assist researchers in their work. For the development of new medications, the capacity to interpret medically significant human differences is critical. This paper provides a model that can predict whether a variant has conflicting classifications. A model that can predict the presence of a contradictory classification for a variant can save time for researchers looking for one.

Integration of multi-modal datasets and understanding bases, high computational power, big data analytics, and large-scale machine learning are all required for analysis, which in the past may take longer time until months, delaying time to insights. AWS speeds up the study of enormous genomics data by combining machine learning and supercomputing. Researchers can leverage AWS to gain more computing capacity at scale, repeatable data processing, data integration tools to pull in multi-modal datasets, and public data for scientific annotation—all while remaining compliant.

2.0 CLIENT INFORMATION

Name: Alvin Nesakumar

Position: Bioinformatics Software Engineer (Executive Director)

Company: Malaysian Genomics Resource Centre Berhad (MGRC)

Problem:

Malaysian Genomics Resource Centre Berhad (MGRC) is a genomics and biopharmaceutical company. This Company is engaged in providing bioinformatics analysis and genetic screening services. MGRC plans to create a machine-learning model based on tertiary analysis to predict if a variant has a conflicting classification. According to National Centre Biotechnology Information, variant classification is the process of determining whether a DNA variant causes disease, is in clinical practice generally the province of genetic testing laboratories and relies on multiple lines of evidence. Hence, the issue they encountered previously is the researchers take too much time looking for the conflict in the classification of the genomic variant.

Requirement:

The machine-learning model should utilize tertiary analysis to classify variants and achieve high performance in multiple genes and different health conditions.

3.0 ARCHITECTURE PLANNING AND DESIGN

A machine-learning model for variant classification was developed using the solution of tertiary analysis in Amazon Web Services(AWS). The expected tertiary is performed through a web application to assist bioinformatics scientists and researchers in a clinical reporting workflow.

Reasons for choosing Amazon Web Service (AWS) as a platform

- Security
- Scalability
- Adaptability
- Improved productivity
- Cost-effective

How does AWS help to solve the problem?

Tertiary Analysis in Genomics Using the Amazon SageMaker offering, users may construct machine learning models on genomic datasets using AWS managed services. The tertiary analysis employs complicated computer science methods, algorithms, and tools to comprehend and analyze the sequencing results retrieved from raw genomic data. Variant classification is the process of determining whether a DNA variant can cause a disease depending on its size and whether it can impact our health or not. Here, variant classification is an example of a scientifically significant problem that may embark on using this platform, which provides a comprehensive platform for genomic machine learning on AWS. A model that can predict the existence of a conflicting categorisation for a variant can help researchers save time by reducing the amount of time they spend searching for such conflicts.

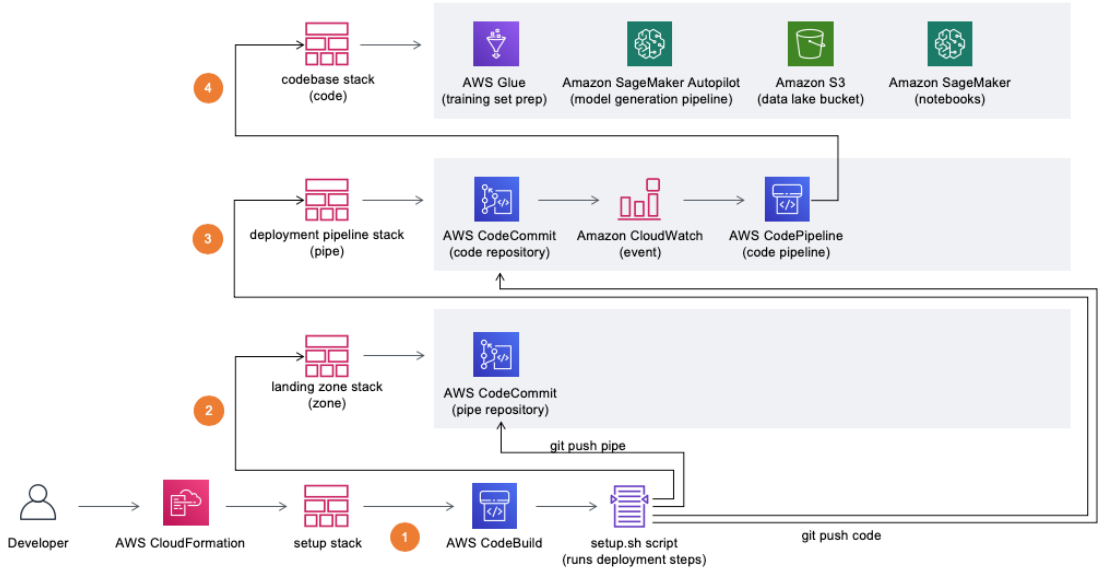
This solution shows:

1. How to automate the production of a genomics machine learning training dataset.
2. Create genomics machine learning model training and deployment pipelines
3. Make predictions and evaluate model performance using test data.

For their individual use cases, users can repeat or alter these procedures.

The overview of AWS Solutions Implementation

The architecture below enables users to deploy automatically using the solution's implementation guide and AWS CloudFormation template.



(Ratan et al., n.d.)

The AWS CloudFormation template builds four CloudFormation stacks in the user's AWS account.

1. A landing zone (zone) stack containing common solution resources and artefacts.

   The CodeCommit pipe repository is created by the landing zone (zone) stack. The setup.sh script publishes source code to the CodeCommit pipe repository after the landing zone (zone) stack has completed its setup.

2. A deployment pipeline (pipe) stack defining the solution's continuous integration and continuous delivery (CI/CD) pipeline.

   The CloudCommit code repository, an Amazon CloudWatch event, and the AWS CodePipeline code pipeline are all created by the deployment pipeline (pipe) stack. The setup.sh script publishes source code to the CodeCommit code repository after the deployment pipeline (pipe) stack is built up.

3. A codebase (code) stack containing ETL scripts, jobs, crawlers, a data catalogue, and notebook resources are among the other stacks.

The CloudFormation stack is deployed via the CodePipeline (code) pipeline. After the AWS CodePipeline pipelines are set up, Amazon Simple Storage Service (Amazon S3) buckets are created in your account for storing objects, access logs, building artefacts, and data. Source code repositories on CodeCommit. An AWSCodeBuild project for creating code artefacts (for example, third-party data processing libraries). AWS CodePipeline pipeline for automating resource builds and deployment, with AWS Glue jobs and an Amazon SageMaker Jupyter notebook as an example.

4. Setup stack for installing the solution.

   The setup stack of the solution creates an AWS CodeBuild project with the setup.sh script in it. This script generates the remaining CloudFormation stacks and includes the source code for the AWS CodeCommit pipe repository and the code repository. The example code offers all of the resources needed to create machine learning models and generate genomics data predictions quickly.

Notes:

- A CloudFormation stack is a group of AWS resources managed as a single entity.
- ETL stands for Extract, Transform and Load, and ETL scripts is a script that contains a code that does the extracting, transforming and loading of the data.
- AWS Artifact is the go-to, central repository for all compliance-related information. It gives the user access to AWS' security and compliance reports and several online agreements on-demand.

Components

1. CI/CD pipeline

   A continuous integration/delivery (CI/CD) pipeline is a set of procedures that must be followed in order to release a new version of the software. This pipeline is built with AWS support. AWS CodeCommit acts as the initiator code, AWS CodeBuild as a platform to copy resources to the "ResourcesBucket" in Amazon S3 bucket, and AWS CodePipeline to run the project and use AWS CloudFormation to automate deployment after publishing the updated source code.

2. Solution demonstration datasets

   This solution will copy the variants with VEP annotations public datasets to the "DataLakeBucket" in S3 bucket. The datasets combine information about genomic variation and what it has to do with human health. Then, the final training dataset will be copied into the "DataLackBucket" S3 bucket, where the AWS Glue is non-compulsory in this part.

3. AWS Glue Jobs

   AWS Glue is a cloud service that is simple to prepare data for analysis and combine data for machine learning and application development. This solution creates the "create-training-set" in AWS Glue, which changes the datasets with variant effect predictors where these predictors are then added into variant datasets.

4. SageMaker Notebook Instance

   This solution creates an Amazon SageMaker notebook instance that shows how to use AWS Glue and Amazon SageMaker Autopilot to create a machine learning model generation pipeline.

5. Amazon S3 Bucket

   This solution provides a few buckets where each bucket has encryption and logging enabled. They are DataLakeBucket, ResourcesBucket, BuildBucket, and LogsBucket. Each of them has its function. DataLakeBucket stores ClinVar variant annotation data and the solution training dataset, ResourcesBucket stores notebooks and shell scripts, BuildBucket stores build artefacts deployed through the pipeline and LogsBucket stores solution log files.

4.0 CONCLUSION

In conclusion, in order to help our imaginary client to overcome the conflict in variant classification, we developed a low fidelity machine learning prototype, which is Tertiary Analysis. To afford satisfaction to our imaginary client, we decided to use Amazon Web Services (AWS) to develop a Tertiary Analysis. We are confident AWS could help the client in boosting productivity. The AWS sagemaker provided by AWS enables developers to create, train, and deploy machine-learning models in the cloud. Users may construct machine learning models on genomic datasets using AWS managed services. There are some solutions and components that should be implemented to develop this Tertiary Analysis. In contemplation of developing the prototype of this machine learning, we researched AWS to know more about this platform. Using AWS as a platform to perform Tertiary Analysis will give many benefits as it provides a scalable environment to run genomics analysis. Toward the end of this project, we are in no doubt with this low fidelity prototype that uses AWS to perform Tertiary Analysis. We succeed in relating machine learning and cloud architecture. We did satisfy the requirement of our imaginary client.

5.0 REFERENCES

1. Amazon Web Services. (2021). Genomics tertiary analysis and machine learning. Retrieved from https://aws.amazon.com/solutions/implementations/genomics-tertiary-analysis-and-machine-learning-using-amazon-sagemaker/
2. Hum Mutat. (2020). LEAP: Using machine learning to support variant classification in a clinical setting. Retrieved from https://www.ncbi.nlm.nih.gov/books/NBK236037/
3. *Working with Scripts on the AWS Glue Console - AWS Glue*. (2021). Amazon.com. https://docs.aws.amazon.com/glue/latest/dg/console-edit-script.html
4. *AWS Artifact - Amazon Web Services (AWS)*. (2021). Amazon Web Services, Inc. https://aws.amazon.com/artifact/
5. Combes, P. (2021, July 15). *Introducing AWS for Health – Accelerating innovation from benchtop to bedside*. Amazon. Retrieved December 26, 2021, from https://aws.amazon.com/blogs/industries/announcing-aws-for-health/
6. *Hospitality management*. Lasell University. (2021). Retrieved December 26, 2021, from https://www.lasell.edu/academics/20-21-academic-catalog/undergraduate-catalog-20-21/programs-of-study-20-21/hospitality-management-20-21.html
7. Shuhalii, A. (2020, October 15). *What is the difference between low and high fidelity prototypes?* Medium. Retrieved December 26, 2021, from https://bootcamp.uxdesign.cc/what-is-the-difference-between-low-and-high-fidelity-prototypes-b1f3612f85f7