A Cloud Data Lakehouse-Based AI Diagnostic Solution for Small and Medium-Sized Health Facilities

Ruichen Li
Master of Science, Data Science
Amity Global Institute
Singapore
saqpourc@hotmail.com

Murphy Choy
adjunct
School of Computing, Engineering & Digital Technologies
Teesside University
United Kingdom
goladin@gmail.com

Ma Nang Laik
Associate Professor in the School of Business
Singapore University of Social Sciences
Singapore
nlma@suss.edu.sg

Abstract

This research aims to provide an AI diagnostic solution for small and medium-sized health facilities. The problem of medical resources shortages worldwide due to the growing population and the COVID-19 pandemic in recent years has become more severe than ever. However, some evolutionary techniques such as big data, AI, and cloud computing have been leveraged to provide digital transformation solutions with increased productivity in certain specifics. To prove that these techniques can bring benefit and health facilities by increasing productivity with faster, easier decision-more accessible abilities and end-to-end cloud data lakehouse-based solution will be designed and provisioned. Patient's dataset collected from the real world will be used for test and evaluation purposes. Through the research, the architecture of the solution with AI techniques will be designed, tested, and evaluated. The solution will need to be provisioned to the production-grade environment to help targeted health facilities make more efficient diagnosis decision-making and save more lives as possible.

Keywords

CAPEX (capital expense), OPEX (operating expense), TCO (Total Cost of Ownership), cardiovascular diseases (CVDs), IaC (infrastructure as code)

1. Introduction

Lack of medical resources has become a common sense not only in some third-world countries but also in developed countries, especially in recent years (Wu et al. 2016). Huge numbers of health facilities such as clinics or hospitals are working under high pressure, e.g., ICU, bed shortages, and overworked, exhausted hospital staff, mainly caused by the COVID-19 pandemic (Alharbi et al. 2020).

With the development of evolutionary big data, AI and cloud computing have been widely leveraged to provide digital transformation solutions in many industries, e.g., medical and healthcare. A typical example would be some healthcare facilities that have been successfully creating business value and increasing productivity by using big data and AI techniques to process massive amounts of data (Vatandoost et al. 2019). Those processes include making analyses by using SQL queries, Business Intelligence (BI) with data visualization techniques, or even making predictions with machine learning techniques, etc. from multiple types of data stored in data storage like databases, data warehouses, and data lakes. Data may originally come from sources such as historical patient records, CT-scan images, etc., and even social public knowledge bases or IoT device metrics and logs. Those research results may enable those healthcare facilities to identify and respond to business growth opportunities faster by more precise, customized service to attract and retain target customers (patients), increasing productivity by making efficient decisions even though some of those facilities are facing problems with staff resource shortages.

1.1 Objectives

- To investigate background knowledge of data lakehouse, characteristics, and evolution. Why is it useful for health facilities?
- To analyze the benefits of using a cloud-based data lakehouse for small and medium-sized health facilities
- To design the solution architecture based on cloud data lakehouse
- To explore an AI-powered diagnostic system based on cloud data lakes-detailed process
- To evaluate the features and benefits of this solution

2. Literature Review

AI-powered disease diagnosis has become more and more popular recently (SMA 2019). Research using the KNN algorithm for various disease detection (Rezaeijo et al. 2021), and a tumor category predicting solution using deep learning (Nasser and Abu-Naser 2019) proved the feasibility of AI-based solutions. Moreover, leveraging multiple models trained by different machine learning algorithms simultaneously for disease diagnosis prediction can optimize the accuracy (Tarawneh and Embarak 2019). Some successful cases use multiple algorithms such as KNN, SVM, Decision Tree, Random Forest, etc. simultaneously with a voting-based ensemble method layer that can optimize the prediction results better than using any single model through research-related studies (Javid et al. 2020).

Those research or case studies are focused on how to optimize the model's accuracy to extract useful information, and values and then make predictions from datasets no matter whether the data is structured, semi-structured, or even unstructured, e.g., CT-scan images. The amount of data storage might not be very large (TB to PB level storage) for small and medium health facilities at this moment, but with business growth, the volume of the data could also grow very fast, and the growth rate of their data's volume, velocity, veracity, and variety in the short term could be several or even ten times more than nowadays.

Therefore, this solution will aim to help small and medium-sized health facilities provide a high accuracy AI diagnosis solution like other similar AI solutions but with a more scalable, more advanced, and cost-efficient architecture design that makes it easier to maintain and use because the target group usually does not have quite enough IT technicians or departments compared to the typical large hospital.

Common data storage solutions in the industry include databases, data warehouses, and data lakes, which have become popular in recent years. First of all, traditional relational databases such as MySQL can be ignored directly in this research because the relational database is not natively designed for AI and data analytics workloads. Relational databases are mostly good at OLTP but not suitable for OLAP, and it is hard to handle read-intensive workloads on large data sets (Raut and A.B.P.D 2017). The data warehouse and data lake will be more focused on for this research because those types of storage solutions are natively designed to support OLAP, running well for data query, analysis, or BI workloads while satisfying the demand of massive data storage.

Unlike data warehouses which are good at dealing with structured data and mostly have extremely good query performance, the data lake is a centralized storage repository that can store structured, unstructured, or semi-structured data at any scale. Enterprises can capture and store the data as it is (no need to pre-define or transform the data first) and then integrate it with backend big data services to process different types of data analysis, data visualizations, real-time analysis, and machine learning workloads (Miloslavskaya and Tolstoy 2016).

There is a new solution – the data lakehouse that can combine the functions of the data lake and the data warehouse to take advantage of low-cost, massive storage for semi-structured and unstructured data from the data lake while simultaneously providing high-performance data query, analysis, ACID, and version control, etc. as data warehouse capabilities. This research will use the data lakehouse solution to help small and medium health facilities meet the demands for data storage, analysis, and forecast.

The data lakehouse combines the benefits of data warehouses and data lakes, which implement similar data structures and data management functions as in a data warehouse directly on cost-effective storage for the data lake. Especially for small and medium health facilities that need to run big data, machine learning workloads on either structured, semi-structured, or unstructured data such as CT-scan images, the data lakehouse is a perfect choice with low cost, secure, reliable, and scalable features (Armbrust et al. 2021). (Table 1)

Table 1. Data warehouse **VS** data lake **VS** data lakehouse.

Solutions	Pros	Cons	Use cases
Data warehouse	Data are stored with high quality. Data schemas are well designed and transformed to meet specific business demands.	High start-up cost. Data schemas need to be pre-defined and pre-processed based on requirements.	Data analysis, sql query, BI using processed, structured data. Read intensive workloads.
Data lake	Low start-up cost. Economic for long-term store massive data store, no matter which is structed or not.	Poor performance for read-intensive query workloads. The massive size and types of the unprocessed, raw data store can cause chaos. Data governance could be a challenge.	Long-term, cost effective data storage. Support comprehensive big data analytics, machine learning, deep learning workloads.
Data lakehouse	Low-term cost effective storage for different types of data. Provide high performance for query intensive, data analysis workloads. ACID transactions and updates to data with versioning capabilities.	Less case studies. Less community support.	Long-term, cost effective data storage. Read intensive sql query workloads. BI, Data analysis, machine learning, deep learning etc.

After choosing data lakehouse as the target group's data platform, the next thing to consider is how and where to deploy it: in traditional local on-premise infrastructure or on the cloud? After reading through the following paragraph on cloud computing's basic knowledge and features, people will believe using the cloud computing-based data lakehouse would be a better choice than using a traditional on-premise data center.

3. Methods

The high-level overview of the solution architecture: (Figure 1)

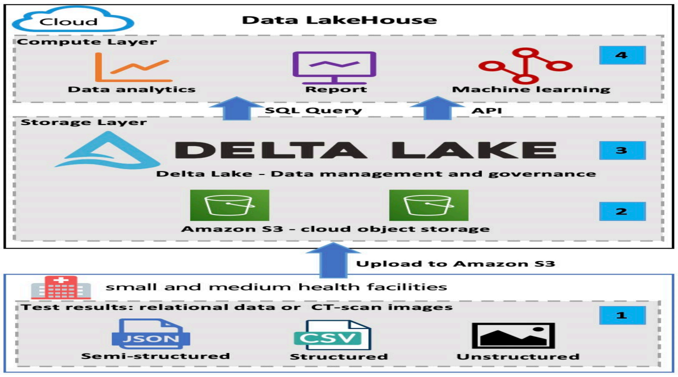


Fig 1. Solution Architecture Design – The Logic Architecture

1. Test results may contain structured, unstructured, and semi-structured data.

2. Upload these test reports to the cloud object storage automatically or periodically through programs or scripts. We chose Amazon S3 for data lake storage. (Optional choice: GCP storage)

3. Use the open-source data management system: delta lake, which brings ACID transaction support, version control, high-performance SQL query, and other capabilities to the underlayer S3 data storage and technically realizes the formation of the data lakehouse. (Optional choice: Hudi, Iceberg.)

4. BI reports, data analytics, machine learning, and other applications can use in-memory computing by reading data from the delta lake table, no longer needing to load the data to the server's hard disk as before, which reduces costs and improves performance by leveraging the separate storage and computing method.

The infrastructure architecture diagram and related comments are presented below: (Figure 2)

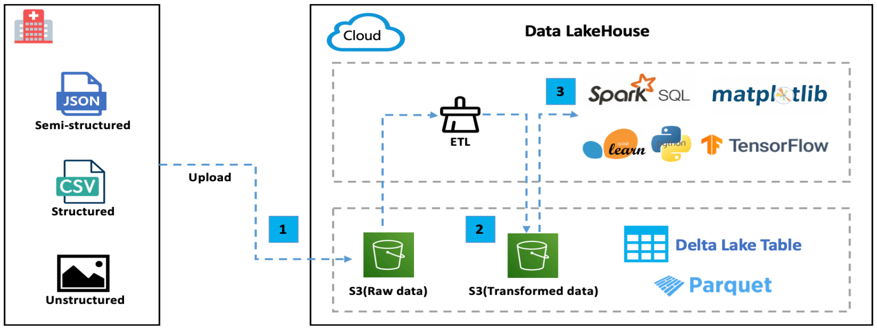


Fig 2. Solution Architecture Design – The Infrastructure Architecture Diagram

1. Upload structured, unstructured, and semi-structured formatted test reports to the Amazon S3 cloud object storage bucket as it is done automatically or periodically through programs or scripts.

2. Extract, transform, and then load those structured or semi-structured formatted test reports into another Amazon S3 cloud object storage bucket. Transformed data should be in parquet format and read from the delta lake table, which brings ACID transaction support, version control, high-performance SQL query, and other capabilities.

3. Open-sourced data analytics and machine learning applications will be used. Data will be read from the delta lake table from the S3 data lake using in-memory computing technology instead of loading the data to the server's hard disk as before. An open-sourced visualization library can help target groups easily gain further insights into the data to support decision-making.

The solution architecture is practical and light-weighted with small resource consumption. Cloud servers and cloud object storage are being used, and those resources can be expanded at any time to meet the demand. In addition, all the software being selected is open-source and free to use, which saves cost and avoids vendor locking risk.

4. Data Collection

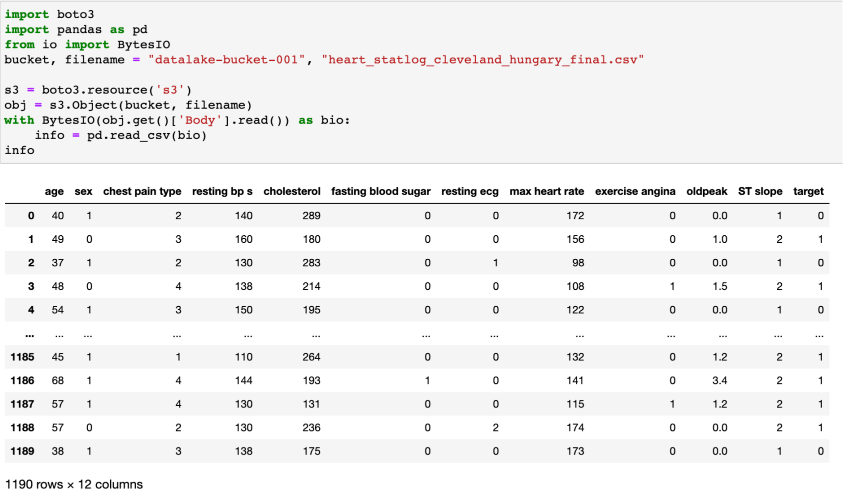
A collection of heart disease test data collected from the real world will be used for POC test purposes. Heart disease is the leading deadly disease and also the number one cause of death in this world. Early detection and relevant medicines or treatments are very important to reduce or even prevent the risk of heart attack (WHO 2021).
Data source: https://www.kaggle.com/sid321axn/heart-statlog-cleveland-hungary-final

The dataset contains 11 features and 1 target variable, as shown below: (Table 2)

Table 2. The description of the dataset.

No	Columns	Definitions	Type
1	Age	Age of Patients.	Numeric
2	Sex	Gender of Patients 1=male, 0=female.	Nominal
3	Chest Pain Type	Types of chest pain experienced. 1=typical, 2=typical angina, 3=non-anginal pain, 4=asymptomatic.	Nominal
4	resting bp s	Level of the blood pressure at resting mode.	Numerical
5	cholestrol	Serum cholestrol.	Numeric
6	fasting blood sugar	Levels of blood sugar on fasting > 120 mg/dl represents as 1=true, 0=false.	Nominal
7	resting ecg	Electrocardiogram's result while at rest are represented in values 0=Normal 1=Abnormality in ST-T wave 2=Left ventricular hypertrophy.	Nominal
8	max heart rate	Maximum value of heart rate achieved.	Numeric
9	exercise angina	Angina caused by exercise 0=NO 1=Yes.	Nominal
10	oldpeak	Compare with the state of rest, the exercise caused ST-depression.	Numeric
11	ST slope	During peak exercises, the ST segment measured in terms of slope 0=Normal 1=Upsloping 2=Flat 3=Downsloping.	Nominal
12	target	1=patient has CVDs 0=patient is normal.	Nominal

There are multiple ways to get data from the data lakehouse, Use Python to get data from S3 by calling the S3 API. (Figure 3)



```
import boto3
import pandas as pd
from io import BytesIO
bucket, filename = "datalake-bucket-001", "heart_statlog_cleveland_hungary_final.csv"

s3 = boto3.resource('s3')
obj = s3.Object(bucket, filename)
with BytesIO(obj.get()['Body'].read()) as bio:
    info = pd.read_csv(bio)
info
```

	age	sex	chest pain type	resting bp s	cholesterol	fasting blood sugar	resting ecg	max heart rate	exercise angina	oldpeak	ST slope	target
0	40	1	2	140	289	0	0	172	0	0.0	1	0
1	49	0	3	160	180	0	0	156	0	1.0	2	1
2	37	1	2	130	283	0	1	98	0	0.0	1	0
3	48	0	4	138	214	0	0	108	1	1.5	2	1
4	54	1	3	150	195	0	0	122	0	0.0	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...
1185	45	1	1	110	264	0	0	132	0	1.2	2	1
1186	68	1	4	144	193	1	0	141	0	3.4	2	1
1187	57	1	4	130	131	0	0	115	1	1.2	2	1
1188	57	0	2	130	236	0	2	174	0	0.0	2	1
1189	38	1	3	138	175	0	0	173	0	0.0	1	0

1190 rows × 12 columns

Fig 3. Use Python to get data from S3 by calling the S3 API.

5. Results and Discussion

5.1 Numerical Results

For data analysis:

Using SparkSQL to query and detect whether there is an imbalanced class problem in the transformed, processed data.

Table 3. The query results of the two classes.

Heart disease	Normal
55%	45%

There is about a 10% difference between the two classes, which is not considered a typical imbalanced class problem dataset.(Table 3)

Table 4. Use Spark SQL to run a customized query to discover patterns from patients with different chest pain types.

Target **chest pain type**	**0**	**1**
1	6.340000	3.940000
2	36.340000	4.720000
3	31.950000	14.170000
4	25.370000	77.170000

Patients with CVDs who have chest pain type 4 are much more than patients without CVDs, while patients without CVDs who have chest pain types 3 and 2 are much more than those who have CVDs.(Table 4)

Table 5**.** Use Spark SQL to run another customized query to discover patterns from patients with different ST slope segments.

Target **ST slope**	**0**	**1**
1	0.000000	0.200000
2	77.320000	15.350000
3	19.270000	74.800000
4	3.410000	9.650000

Patients with CVDs who have ST slope 2 are much more numerous than patients without CVDs, while patients without CVDs who have ST slope 1 are much more numerous than those who have CVDs. (Table 5)

For machine learning:

Table 6. Accuracy and precision scores of the training results.

Rounds	**Accuracy score**	**Precision score**
1	KNN: 0.853	KNN: 0.844
	NN: 0.8478260636329651	NN: 0.8407079646017699
	SVM: 0.8369565217391305	SVM: 0.8596491228070176
	RFC: 0.8260869565217391	RFC: 0.8130841121495327
	Bayes: 0.8641304347826086	Bayes: 0.8775510204081632
2	KNN: 0.8695652173913043	KNN: 0.8686868686868687
	NN: 0.85326087474823	NN: 0.8482142857142857
	SVM: 0.8586956521739131	SVM: 0.8585858585858586
	RFC: 0.842391304347826	RFC: 0.8867924528301887
	Bayes: 0.842391304347826	Bayes: 0.9029126213592233
	KNN: 0.8586956521739131	KNN: 0.8148148148148148

3	NN: 0.8913043737411499	NN: 0.8969072164948454
	SVM: 0.8804347826086957	SVM: 0.8717948717948718
	RFC: 0.8913043478260869	RFC: 0.9074074074074074
	Bayes: 0.8695652173913043	Bayes: 0.9032258064516129
4	KNN: 0.8804347826086957	KNN: 0.9052631578947369
	NN: 0.89673912525177	NN: 0.8785046728971962
	SVM: 0.8804347826086957	SVM: 0.8811881188118812
	RFC: 0.8641304347826086	RFC: 0.8557692307692307
	Bayes: 0.8478260869565217	Bayes: 0.875
5	KNN: 0.8641304347826086	KNN: 0.883495145631068
	NN: 0.8913043737411499	NN: 0.9
	SVM: 0.8913043478260869	SVM: 0.8918918918918919
	RFC: 0.8586956521739131	RFC: 0.9145299145299145
	Bayes: 0.8695652173913043	Bayes: 0.8942307692307693
6	KNN: 0.8695652173913043	KNN: 0.9035087719298246
	NN: 0.8804347826086957	NN: 0.8811881188118812
	SVM: 0.8478260869565217	SVM: 0.8811881188118812
	RFC: 0.8641304347826086	RFC: 0.83
	Bayes: 0.8858695652173914	Bayes: 0.8921568627450981
7	KNN: 0.8478260869565217	KNN: 0.8921568627450981
	NN: 0.9021739363670349	NN: 0.9230769230769231
	SVM: 0.8586956521739131	SVM: 0.8476190476190476
	RFC: 0.8913043478260869	RFC: 0.8653846153846154
	Bayes: 0.8804347826086957	Bayes: 0.9029126213592233
8	KNN: 0.875	KNN: 0.8717948717948718
	NN: 0.9021739363670349	NN: 0.9150943396226415
	SVM: 0.875	SVM: 0.9051724137931034
	RFC: 0.875	RFC: 0.8909090909090909
	Bayes: 0.8695652173913043	Bayes: 0.8854166666666666
9	KNN: 0.8532608695652174	KNN: 0.8557692307692307
	NN: 0.8695651888847351	NN: 0.8504672897196262
	SVM: 0.8695652173913043	SVM: 0.9035087719298246
	RFC: 0.875	RFC: 0.9134615384615384
	Bayes: 0.8315217391304348	Bayes: 0.8910891089108911
10	KNN: 0.8804347826086957	KNN: 0.8738738738738738
	NN: 0.8913043737411499	NN: 0.9181818181818182
	SVM: 0.8913043478260869	SVM: 0.8969072164948454
	RFC: 0.8804347826086957	RFC: 0.8632478632478633
	Bayes: 0.8804347826086957	Bayes: 0.9010989010989011

The accuracy and precision scores of those models trained by different algorithms are close.(Table 6)

5.2 Graphical Results

For data analysis:

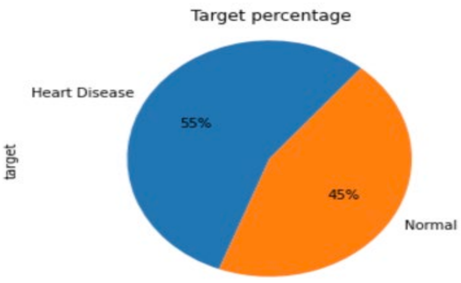


Fig 4. Visualized Spark SQL query results show there is about a 10% difference between the two classes, which is not considered a typical imbalanced class problem dataset.

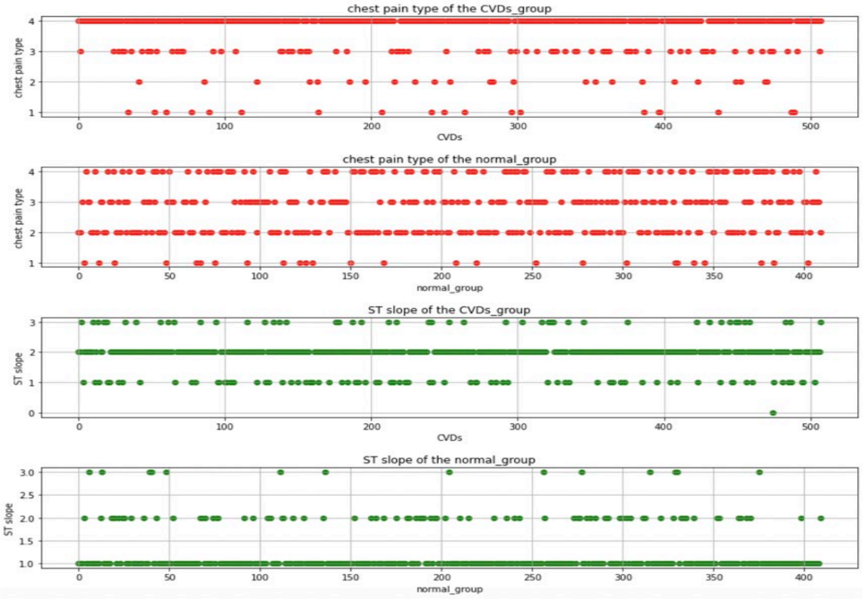


Fig 5. Significant differences in chest pain type and ST slope features exist between the two classes of patients.

For machine learning:

The following chart displays the multiple rounds accuracy and precision scores for all models. (Figure 6)

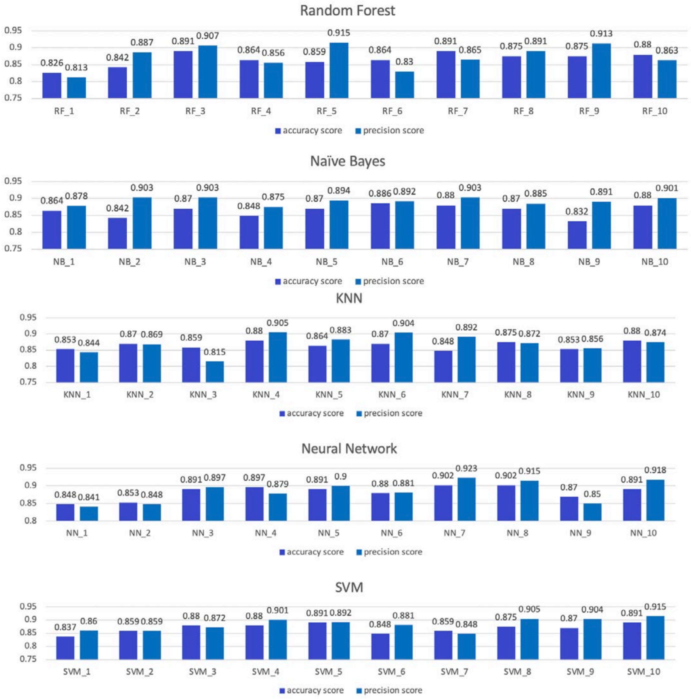
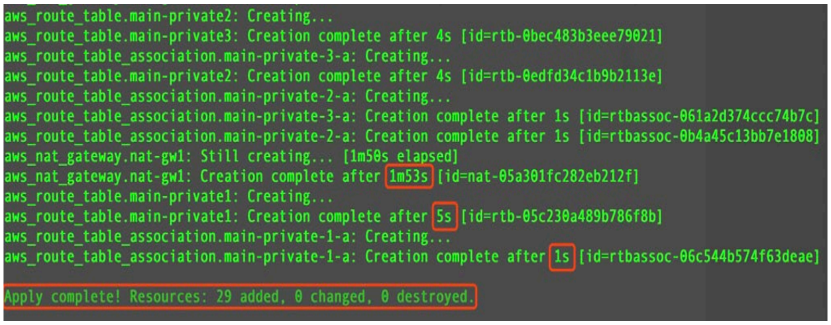


Fig 6. The scores of models trained by different algorithms are close, with no significant difference between them.

5.3 Proposed Improvements

Firstly, this data lakehouse solution's relevant resource creation time is only about **2 minutes b**y using the IaC technique. However, deploying these resources in a traditional data center will usually take days or weeks.(Figure 7)



```
aws_route_table.main-private2: Creating...
aws_route_table.main-private3: Creation complete after 4s [id=rtb-0bec483b3eee79021]
aws_route_table_association.main-private-3-a: Creating...
aws_route_table.main-private2: Creation complete after 4s [id=rtb-0edfd34c1b9b2113e]
aws_route_table_association.main-private-2-a: Creating...
aws_route_table_association.main-private-3-a: Creation complete after 1s [id=rtbassoc-061a2d374ccc74b7c]
aws_route_table_association.main-private-2-a: Creation complete after 1s [id=rtbassoc-0b4a45c13bb7e1808]
aws_nat_gateway.nat-gw1: Still creating... [1m50s elapsed]
aws_nat_gateway.nat-gw1: Creation complete after 1m53s [id=nat-05a301fc282eb212f]
aws_route_table.main-private1: Creating...
aws_route_table.main-private1: Creation complete after 5s [id=rtb-05c230a489b786f8b]
aws_route_table_association.main-private-1-a: Creating...
aws_route_table_association.main-private-1-a: Creation complete after 1s [id=rtbassoc-06c544b574f63deae]

Apply complete! Resources: 29 added, 0 changed, 0 destroyed.
```

Fig 7. All those data lakehouse relevant resources are deployed automatically in a few minutes.

Also, compared to the traditional way of doing data processing and analysis on CSV files, this data lakehouse-based solution uses the Parquet format for better performance and much lower cost. Due to the fact that big data-relevant workloads are mostly read-intensive, read performance should be as high as possible. The Figure 8 below is a comparison of the read time of the same data in CSV format and Apache Parquet format using Spark SQL.

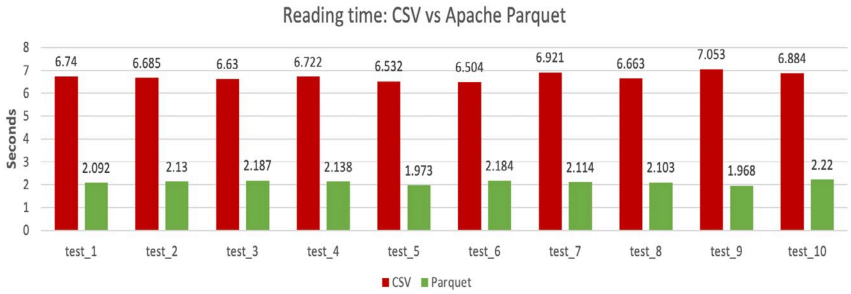


Fig 8. Test results show that reading in CSV formatted data costs much more time than reading Parquet formatted data. the reading speed of the same data in Apache Parquet format is about **3 times** faster than in the CSV format.

Another key benefit would be saving storage space as well as reducing costs. In the previous cloud benefits section, the cost of computing and storage in the cloud has been calculated in detail, which has obvious cost advantages over the traditional data center method of purchasing-deploying-maintaining equipment, especially at the start-up stage of the user. The point here is that the amount of storage space taken by the Apache Parquet format is much lower than the CSV format, which will bring great cost advantages, especially for cloud storage under the pay-as-you-go model such as Amazon S3. (Figure 8) The following part is the analysis of storage size and relevant cost comparison between CSV format storage and Parquet format storage. (Table 7)

Table 7. The size difference of the data in different formats.

Name	Type	Size	Note
heart_statlog_cleveland_hungary_final.csv	CSV	38.8 KB	Raw data in CSV format
part-00000-c2f8acb0-92ce-4119-b16b-2ae7885a39af-c000.snappy.parquet + crc	Parquet	12.1 KB	Raw data in Parquet format

It can be seen that the S3 storage space taken by the data in the Parquet format is only 30% of that stored in CSV format. The storage cost of this data lakehouse solution will be **60% cheaper** if the original data is in CSV format.

The original dataset that comes with limited training samples may cause a tiny difference in the trained model's accuracy. Some well-known powerful algorithms such as SVM and neural networks will perform close to some normal, traditional algorithms even with Grid Search CV to optimize the model's accuracy. To ensure the accuracy of the prediction results, the ensemble bagging method will be leveraged to optimize the accuracy of trained models. The methodology of how the method works is shown Figure 9 below.

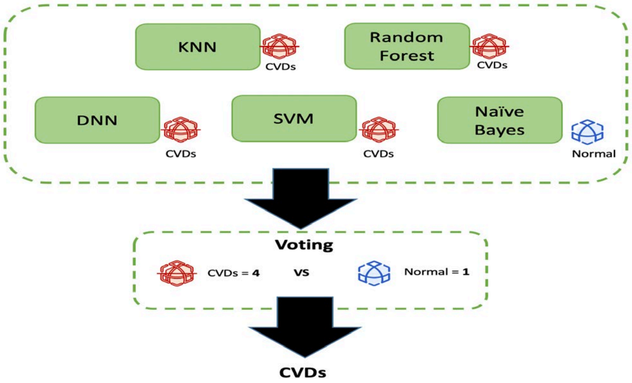
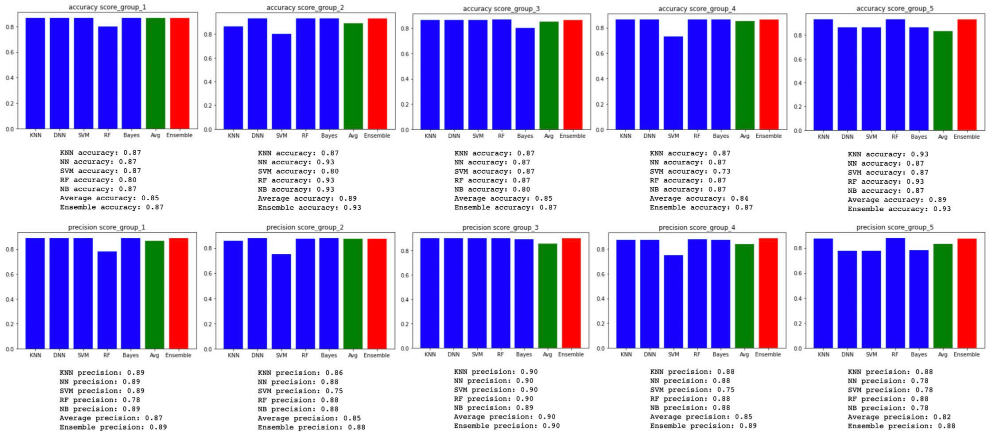


Fig 9. The final output results will be determined by the majority voting-based ensemble method.

The following Figure 10 displays the accuracy score by using trained models to make predictions on a group of random patient's data as well as their average accuracy score, compared with the accuracy score of the ensemble bagging method. The accuracy score of using different models is still close, but the score of ensemble bagging models is mostly higher than the average score, which makes for reducing the misjudgment risk caused by any single model.



accuracy score_group_1

```
KNN accuracy: 0.87
NN accuracy: 0.87
SVM accuracy: 0.87
RF accuracy: 0.80
NB accuracy: 0.87
Average accuracy: 0.85
Ensemble accuracy: 0.87
```

accuracy score_group_2

```
KNN accuracy: 0.87
NN accuracy: 0.93
SVM accuracy: 0.80
RF accuracy: 0.93
NB accuracy: 0.93
Average accuracy: 0.89
Ensemble accuracy: 0.93
```

accuracy score_group_3

```
KNN accuracy: 0.87
NN accuracy: 0.87
SVM accuracy: 0.87
RF accuracy: 0.87
NB accuracy: 0.80
Average accuracy: 0.85
Ensemble accuracy: 0.87
```

accuracy score_group_4

```
KNN accuracy: 0.87
NN accuracy: 0.87
SVM accuracy: 0.73
RF accuracy: 0.87
NB accuracy: 0.87
Average accuracy: 0.84
Ensemble accuracy: 0.87
```

accuracy score_group_5

```
KNN accuracy: 0.93
NN accuracy: 0.87
SVM accuracy: 0.87
RF accuracy: 0.93
NB accuracy: 0.87
Average accuracy: 0.89
Ensemble accuracy: 0.93
```

precision score_group_1

```
KNN precision: 0.89
NN precision: 0.89
SVM precision: 0.89
RF precision: 0.78
NB precision: 0.89
Average precision: 0.87
Ensemble precision: 0.89
```

precision score_group_2

```
KNN precision: 0.86
NN precision: 0.88
SVM precision: 0.75
RF precision: 0.88
NB precision: 0.88
Average precision: 0.85
Ensemble precision: 0.88
```

precision score_group_3

```
KNN precision: 0.90
NN precision: 0.90
SVM precision: 0.90
RF precision: 0.90
NB precision: 0.89
Average precision: 0.90
Ensemble precision: 0.90
```

precision score_group_4

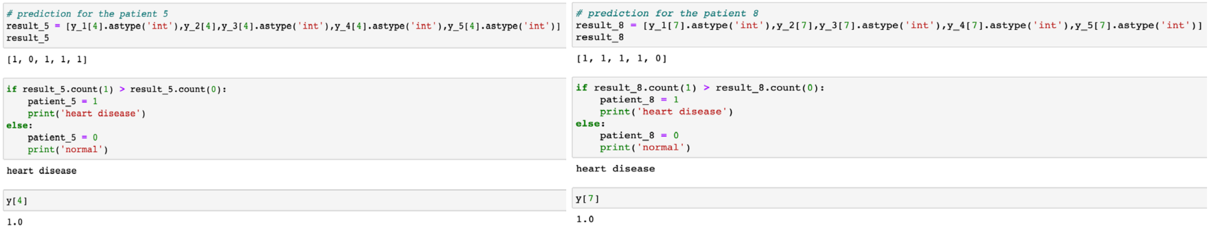
```
KNN precision: 0.88
NN precision: 0.88
SVM precision: 0.75
RF precision: 0.88
NB precision: 0.88
Average precision: 0.85
Ensemble precision: 0.89
```

precision score_group_5

```
KNN precision: 0.88
NN precision: 0.78
SVM precision: 0.78
RF precision: 0.88
NB precision: 0.78
Average precision: 0.82
Ensemble precision: 0.88
```

Fig 10. The accuracy score of the ensemble bagging models tends to be higher than the average for most rounds.

5.4 Validation



```
# prediction for the patient 5
result_5 = [y_1[4].astype('int'),y_2[4],y_3[4].astype('int'),y_4[4].astype('int'),y_5[4].astype('int')]
result_5

[1, 0, 1, 1, 1]

if result_5.count(1) > result_5.count(0):
    patient_5 = 1
    print('heart disease')
else:
    patient_5 = 0
    print('normal')

heart disease

y[4]

1.0
```

```
# prediction for the patient 8
result_8 = [y_1[7].astype('int'),y_2[7],y_3[7].astype('int'),y_4[7].astype('int'),y_5[7].astype('int')]
result_8

[1, 1, 1, 1, 0]

if result_8.count(1) > result_8.count(0):
    patient_8 = 1
    print('heart disease')
else:
    patient_8 = 0
    print('normal')

heart disease

y[7]

1.0
```

Fig 11. The majority voting-based ensemble bagging method can reduce the risk of a single model's “mistake”.

It was verified that in the case of the accuracy, the difference between the use of various algorithms was small, which was caused by a small-sized dataset. This situation can make people struggle to select the algorithm with significantly higher accuracy. Fortunately, leveraging the majority voting-based ensemble bagging method has been proven through experiments to eliminate the risks of a single model's misjudgment in a case like this.(Figure 11)

6. Conclusion

The experimental goals of this solution have been successfully achieved, and the cloud-based data lakehouse that can provide AI diagnosis services for small and medium-sized health facilities has been proved. Moreover, the solution is more advanced than other traditional data science solutions because it provides benefits such as better performance, more cost-effectiveness, etc.

The entire solution is very beneficial for small and medium-sized health facilities where doctors and medical resources are frequently insufficient. All data analysis and prediction processes can be executed in batches, and the result can be provided to doctors through various charts to achieve a faster and more efficient diagnosis.

References

Alharbi, J., Jackson, D. and Usher, K., The potential for COVID-19 to contribute to compassion fatigue in critical care nurses. *Journal of clinical nursing*, 29(15-16), pp.2762–2764, 2020.

Armbrust, M., Ghodsi, A., Xin, R. and Zaharia, M., Lakehouse: a new generation of open platforms that unify data warehousing and advanced analytics. *11th Annual Conference on Innovative, Data Systems Research (CIDR '21)*, January 11–15, 2021, Online.

Javid, I. Khalaf, A and Ghazali, R., Enhanced accuracy of heart disease prediction using machine learning and recurrent neural networks ensemble majority voting method*, IJACSA.,* vol. 11, no. 3, 2020.

Miloslavskaya, N and Tolstoy, A., Big Data, Fast Data and Data Lake Concepts*, Procedia Computer Science*, Vol. 88, pp. 300-305, 2016.

Nasser, I. M. and Abu-Naser, S. S., Predicting Tumor Category Using Artificial Neural Networks*,* Vol. 3 Issue 2, pp: 1-7, 2019.

Raut, A.B.P.D., NOSQL database and its comparison with RDBMS. *International Journal of Computational Intelligence Research*, 13(7), pp.1645-1651, 2017.

Rezaeijo, S.M., Ghorvei, M., Abedi-Firouzjah, R., Mojtahedi, H. and Zarch, H. E., Detecting COVID-19 in chest images based on deep transfer learning and machine learning algorithms. *Egypt J Radiol Nucl Med* 52, 145, 2021.

SMA., Artificial Intelligence in Medical Diagnosis, Available: https://sma.org/ai-in-medical-diagnosis/, Dec 15, 2021.

Tarawneh, M and Embarak, O., Hybrid approach for heart disease prediction using data mining techniques*. Acta Sci. Nutritional Health*, vol. 3, no. 7, pp. 147–151, 2019.

Vatandoost, M. and Litkouhi, S., The future of healthcare facilities: how technology and medical advances may shape hospitals of the future. *Hospital Practices and Research*, 4(1), pp.1-11, 2019.

WHO., cardiovascular diseases (CVDs), Available: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds), Mar 22, 2022.

Wu, Q., Zhao, L. and Ye, X.C., Shortage of healthcare professionals in China. *BMJ*, *354*, 2016.

Biography

Ruichen Li is a senior cloud architect with solution architecture, big data and security skills, familiar with VMware, GCP, AWS cloud, and Kubernetes environments. He has more than 10 years of IT industry experience on deal with complex enterprise's global IT infrastructure with data science solution deployment, management, optimization services. He holds lists of IT certifications include: Cisco Certified Network Professional, Google Certified Professional Cloud Architect, AWS Certified Solutions Architect – Professional, etc.

Murphy Choy has over a decade of experience in data analytics and machine learning. His primary expertise lies in the use of data to drive business results and developing solutions and products driven by data. Murphy holds a Doctorate of Professional Studies in Business Analytics from Middlesex University London, Masters of Finance from University College Dublin and Bachelor of Statistics from National University of Singapore. He also holds a diploma of Economics from University of London and Post Graduate Certificate in Business Research from Heriot-Watt University.

Ma Nang Laik is an Associate Professor in the School of Business, Singapore University of Social Sciences (SUSS). She has more than ten years of academic career. She teaches courses in the areas related to business modeling using spreadsheets, data analytics, and logistics and supply chain operations. She holds a Ph.D. from Imperial College, London where her research focused on operations research (OR). Her research expertise lies in the simulation and modeling of large-scale real-world problems and the development of computationally efficient algorithms for better decision-making in the organization.