

**APLIKASI MUDAH ALIH BERASASKAN PENGKOMPUTERAN
AWAN UNTUK PENGESANAN DIABETES MELALUI
PEMBELAJARAN MESIN**

LEE YONG KENT

PROF. MADYA DR. ELANKOVAN A/L A SUNDARARAJAN

ABSTRAK

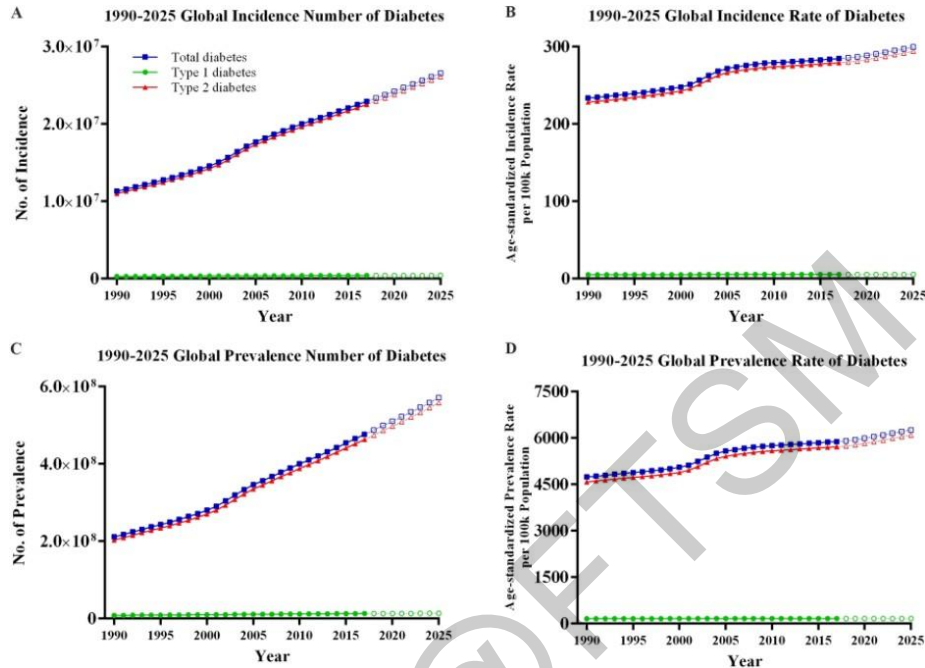
Diabetes ditakrifkan sebagai penyakit metabolik atau kronik yang dicirikan oleh tahap glukosa darah tinggi atau paras gula darah yang akan membawa kepada isu kesihatan yang serius dari masa ke masa. Diabetes boleh dikesan pada peringkat awal dengan simptom dan dapat mengurangkan risiko menghidap diabetes dan menghalangnya daripada menjadi serius. Menurut Pertubuhan Kesihatan Dunia (WHO), diabetes memberi kesan kepada kira-kira 422 juta orang di seluruh dunia dan mengakibatkan 1.6 juta kematian setahun. Sepanjang beberapa dekad yang lalu, kedua-dua kes dan kelaziman diabetes telah meningkat dengan ketara. Malaysia yang berada dalam kedudukan kedua untuk orang dewasa yang menghidap diabetes selepas Pakistan di Asia. Di Malaysia, 3.9 juta orang dewasa berusia 18 tahun ke atas mengalami diabetes dan telah meningkat berbanding dengan 3.5 juta pada tahun 2015. Ini menunjukkan bahawa 1 daripada 5 orang dewasa mempunyai diabetes dan kebanyakan mereka tidak menyedarinya. Malaysia harus mula mengambil langkah untuk mengurangkan risiko mendapat diabetes yang akan membawa kepada banyak masalah kesihatan. Objektif utama projek ini adalah untuk membangunkan aplikasi mudah alih yang menggunakan perkhidmatan awan AWS untuk mengesan risiko mendapatkan diabetes dan memberikan cadangan kepada pengguna melalui pembelajaran mesin. Pembelajaran Mesin adalah cawangan kecerdasan buatan (AI) yang membolehkan komputer belajar dan memperbaiki diri sendiri tanpa perlu diprogramkan secara eksplisit. Selain itu, algoritma pembelajaran mesin boleh menganalisis corak aktiviti pengguna tertentu dan memberi cadangan kepada mereka. Oleh itu, ia biasanya digunakan dalam aplikasi mudah alih.

1 PENGENALAN

Aplikasi mudah alih, yang biasanya dikenali sebagai aplikasi mudah alih atau hanya aplikasi, adalah program komputer atau aplikasi perisian yang berjalan pada peranti mudah alih seperti telefon, tablet atau jam tangan. Aplikasi mudah alih, walaupun sedikit unit perisian dengan fungsi yang terhad, berjaya memberikan pengguna dengan perkhidmatan dan pengalaman yang berkualiti tinggi. Terdapat banyak jenis aplikasi mudah alih yang dicipta untuk tujuan tertentu. Mengikut statistik yang disediakan oleh buildfire, purata pengguna telefon pintar menghabiskan masa 3 jam dan 10 minit setiap hari menggunakan peranti mereka. Kira-kira 2 jam dan 51 minit masa ini dibelanjakan untuk aplikasi dan purata menggunakan 9 aplikasi mudah alih setiap hari dan 30 aplikasi sebulan. Penyelidikan menunjukkan bahawa terdapat 80 aplikasi yang dipasang pada telefon pintar purata. (Buildfire. 2021) Menurut analisis baru mengenai trend kesihatan digital dari Institut IQVIA untuk Sains Data Manusia, bilangan aplikasi kesihatan digital pengguna bertambah dari tahun lepas dan 90,000 yang baru dijangkakan menjelang pada tahun 2021. (E.Olsen. 2021) Pengguna kini mempunyai akses kepada lebih daripada 350,000 aplikasi kesihatan digital, menurut kaji selidik itu. Walaupun ramai yang memberi tumpuan kepada kesihatan atau kecergasan umum, dan yang lain adalah kualiti biasa, bilangan aplikasi pengurusan penyakit juga meningkat. Selain itu, kebanyakan aplikasi mudah alih yang dibangunkan disambungkan ke pengkomputeran awan kerana skala mudah dan kebolehpercayaan yang lebih baik. Dari semua statistik yang disediakan, kita boleh menyimpulkan bahawa aplikasi mudah alih adalah penting untuk kehidupan seharian. Aplikasi mudah alih kesihatan digunakan secara meluas oleh pengguna untuk merakam dan memeriksa isu kesihatan mereka. Sains mendapatkan komputer untuk bertindak tanpa diprogramkan secara eksplisit dikenali sebagai pembelajaran mesin. Kereta memandu sendiri, pengecaman pertuturan yang realistik, carian web yang berjaya, dan pemahaman yang lebih baik mengenai genom manusia semua telah dijayakan oleh pembelajaran mesin dalam dekad yang lalu. Dengan bantuan model atau algoritma pembelajaran mesin, ramalan boleh dibuat dengan ketepatan yang lebih tinggi berbanding dengan teknik ramalan tradisional. Contoh ramalan yang digunakan secara meluas pada masa kini adalah pengiktirafan imej, perkhidmatan kewangan dan diagnosis perubatan. Ini penting kepada masyarakat kerana ia dapat menyelesaikan masalah yang dihadapi dengan lebih tepat dan

cepat. Terdapat tiga jenis pembelajaran mesin yang sedia ada iaitu *Supervised learning*, *Unsupervised Learning* dan *Reinforcement Learning*.

Diabetes adalah salah satu kebimbangan kesihatan awam global yang terbesar, mengenakan beban global yang berat terhadap kesihatan awam serta pembangunan sosio-ekonomi. Walaupun kejadian telah mula berkurangan di sesetengah negara, kelaziman diabetes telah meningkat dalam beberapa dekad kebelakangan ini di kebanyakan negara maju dan membangun yang lain. (Patterson, C. C. et al. 2019)(Wang, L. et al. 2017)(Dwyer-Lindgren et al. 2016) Kehadiran diabetes dikaitkan dengan peningkatan kematian akibat jangkitan, penyakit kardiovaskular, strok, penyakit buah pinggang kronik, penyakit hati kronik, dan kanser. (Bragg, F. et al. 2017) (Policardo, L. et al. 2015) Antara tahun 1990 dan 2017, beban penyakit diabetes di seluruh dunia meningkat secara mendadak (Rajah.1.1). Insiden diabetes meningkat 102.9 peratus di peringkat global daripada 11.3 juta (95 peratus UI 10.6-12.1) pada tahun 1990 kepada 22.9 juta (21.1-25.4) pada tahun 2017. Daripada 233.6 (218.4–249.4) kepada 284.6 (262.2–309.7), kadar kejadian standard umur meningkat. Kelaziman diabetes meningkat sebanyak 129.7% daripada 211.2 juta orang (196.0-228.5) orang pada tahun 1990 kepada 476.0 juta orang (436.6-522.8) pada tahun 2017. Rajah 1A-D menunjukkan kadar kelaziman standard umur meningkat daripada 4,738.5 (4,404.1–5,111.2) kepada 5,886.9 (5,403.6-6,458.5). (Lin, X., Xu, Y., Pan, X. et al. 2020)



Rajah 1.1 Bebanan global diabetes mellitus dari tahun 1990 hingga 2025. (A) Nombor kejadian; (B) Kadar kejadian standard umur; (C) Nombor kelaziman; (D) Kadar kelaziman standard umur

Beban pesakit kencing manis meningkat secara beransur-ansur dari tahun 1990 hingga 2017 dan dijangka meningkat dari 2018 hingga 2025 dari segi kejadian, kelaziman, dan kematian (Gamb.1 A dan C). [12] Lebih tepat lagi, ramalan dinaikkan kepada 26.6 juta dan 570.9 juta orang. Tujuan projek ini adalah untuk membangunkan aplikasi mudah alih untuk pengguna mengesan diabetes melalui gejala dan memberi cadangan kepada pengguna kencing manis dan bukan kencing manis menggunakan pembelajaran mesin dan dengan bantuan perkhidmatan awan AWS. Diabetes boleh dicegah dan mengurangkan risiko menjadi lebih teruk jika didapati pada peringkat awal.

2 PENYATAAN MASALAH

Diabetes adalah masalah serius yang akan merosakkan organ-organ badan dan menyebabkan kesan jangka panjang seperti kerosakan kepada arteri darah utama (makrovaskular) dan kecil (mikrovaskular), yang boleh membawa kepada serangan jantung, strok, dan isu-isu dengan buah pinggang, mata, gusi, kaki, dan saraf. Sebab

mengapa kadar diabetes semakin tinggi dari tahun ke tahun adalah bahwa kebanyakan individu tidak tahu mereka mempunyai gejala diabetes dengan mereka. Mereka hanya mengetahui bahwa mereka mempunyai masalah diabetes apabila diabetes mula campur tangan dalam gaya hidup harian mereka yang sudah terlambat. Risiko mendapat diabetes boleh berkurangan atau menghalangnya daripada menjadi lebih teruk apabila ia ditemui pada peringkat awal. Selain itu, individu tidak tahu bagaimana untuk menangani gejala diabetes. Beberapa gejala biasa yang kebanyakan individu tidak menarik perhatian pada peringkat awal yang akan membawa kepada diabetes adalah poliuria, polidipsia, dan penurunan berat badan secara tiba-tiba, gatal-gatal, dan penyembuhan luka yang ditangguhkan. Aplikasi yang tersedia di pasaran tidak mempunyai fungsi yang menyeluruh. Kebanyakan aplikasi diabetes di pasaran menyediakan sistem pemeriksaan yang hanya menggunakan umur, ketinggian, lilitan pinggang yang tidak tepat berbanding dengan menggunakan gejala sebagai input.

Banyak jenis algoritma pembelajaran mesin boleh digunakan untuk latihan dan menguji set data untuk memberikan hasil dan cadangan. Sebagai contoh, regresi linear, regresi logistik, pokok keputusan, SVM, dan banyak lagi. Setiap algoritma pembelajaran mesin memiliki kekuatan dan kelemahan mereka. Untuk mendapatkan keputusan yang terbaik dan ketepatan tertinggi dan memberikan cadangan yang sesuai kepada pengguna, algoritma pembelajaran mesin harus dipilih dengan betul untuk melatih set data. Cadangan yang diberikan perlu difahami dengan mudah oleh pengguna supaya mereka dapat mengikuti dan mengurangkan risiko mendapat diabetes atau penjejasan diabetes dalam kehidupan seharian mereka.

3 OBJEKTIF KAJIAN

Secara keseluruhan, objektif bagi projek ini tertumpu kepada tiga aspek ini:

- a. Membangunkan aplikasi mudah alih yang mesra pengguna.
- b. Mengesan risiko mendapat diabetes.

- c. Berikan cadangan kepada pengguna tentang diabetes.

4 METHOD KAJIAN

Kaedah yang dipilih untuk membangunkan aplikasi mudah alih dalam projek ini adalah model Agile. Model Agile adalah salah satu model dari SDLC (*Software Development Life Cycle*). SDLC adalah satu kaedah untuk menghasilkan perisian berkualiti tinggi kos rendah dalam masa yang paling sedikit. SDLC adalah aliran peringkat yang tersusun yang membolehkan syarikat dengan cepat membangunkan perisian berkualiti tinggi yang telah diuji dengan teliti dan bersedia untuk pengeluaran.



Rajah 4.1 Rajah Model Agile

4.1 Fasa Perancangan (*PLAN*)

Ini adalah fasa awal, yang bermula dengan perbincangan dengan penyelia dan pemerolehan keperluan dan maklumat berkaitan kajian melalui Internet. Kertas kerja akan disediakan dan penyelidikan akan dilakukan. Jadual waktu disediakan dan semua kerja akan diteruskan mengikut jadual waktu. Contohnya, penyelidikan, objektif projek, skop dan sasaran, bagaimana aplikasi akan dibangunkan, cara aplikasi akan diuji dan cara untuk mengemas kini dan menyahpejijat aplikasi.

4.2 Fasa Reka Bentuk (DESIGN)

Dalam fasa ini, semua keperluan dan maklumat yang dikumpul akan digunakan untuk mereka bentuk aplikasi mudah. Untuk memahami matlamat dan cabaran kajian, analisis mendalam akan dijalankan. Antaramuka akan direka bentuk dalam fasa ini menggunakan perisian *Invision* untuk mempunyai gambaran keseluruhan bagaimana aplikasi akan dihasilkan. Penulisan kod akan dijalankan dalam fasa ini untuk memastikan aplikasi boleh dilaksanakan untuk bahagian teknikal. Fasa ini juga akan menentukan fungsi dan spesifikasi mana yang paling sesuai dengan keperluan pengguna.

4.3 Fasa Pembangunan (BUILD)

Fasa ini akan bermula dengan kod aplikasi dan antaramuka pengguna. Penulisan kod akan ditulis mengikut keperluan pengguna dan spesifikasi reka bentuk. Antaramuka akan disahkan dalam fasa ini untuk memastikan ia mesra pengguna. Aplikasi mudah alih yang berfungsi dan lengkap akan dibina pada akhir fasa ini. Langkah ini juga akan mengambil masa yang lama kerana tugas-tugas teknologi yang terlibat.

4.4 Fasa Menguji (TEST)

Dalam fasa menguji ini, aplikasi mudah alih yang dibangunkan akan diuji. Sebarang pepijat (*bug*) atau ralat akan dinyahpepijat (*debugged*) jika timbul dan antaramuka akan berubah jika ia tidak sesuai. Aplikasi mudah alih yang dibangunkan akan dihantar ke platform seperti *Google Playstore* untuk memastikan ia boleh dimuat turun dan dapat berjalan dengan lancar pada kebanyakan telefon *Android*. Semua fungsi dan reka bentuk dan kelancaran aliran akan diuji untuk menentukan seberapa baik ia berfungsi.

4.5 Fasa Penilaian (REVIEW)

Maklum balas pengguna akan diambil selepas aplikasi mudah alih digunakan, sama ada sebagai demo atau penggunaan sebenar. Selepas mengumpul semua ulasan, aplikasi mudah alih akan diperbaiki atau *debug*, dan versi baru aplikasi akan dikeluarkan.

5 HASIL KAJIAN

Proses pembangunan projek ini dibahagikan kepada 2 bahagian iaitu pembangunan model pembelajaran mesin dan pembangunan aplikasi mudah alih. Perisian *Google Colab* dan bahasa *Python* akan digunakan untuk menjalankan pembersihan data dan membangunkan model pembelajaran mesin.

5.1 Pembangunan Model

```

Train and Test Split

[ ] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, stratify=y, random_state = 12345)

## checking distribution of target variable in train test split
print('Distribution of target variable in training set')
print(y_train.value_counts())

print('Distribution of target variable in test set')
print(y_test.value_counts())

Distribution of target variable in training set
1    256
0    160
Name: class, dtype: int64
Distribution of target variable in test set
1     64
0     40
Name: class, dtype: int64

[ ] X_FS = X[['Polyuria', 'Polydipsia', 'Age', 'Gender', 'partial paresis', 'sudden weight loss', 'Irritability', 'delayed healing', 'Alopecia', 'Itching']]

[ ] X_train, X_test, y_train, y_test = train_test_split(X_FS, y, test_size = 0.2, stratify=y, random_state = 12345)

Data Normalization

[ ] minmax = MinMaxScaler()
X_train[['Age']] = minmax.fit_transform(X_train[['Age']])
X_test[['Age']] = minmax.transform(X_test[['Age']])

```

Rajah 5.1 Kod sumber untuk membahagikan keseluruhan set data kepada set latihan, set ujian dan set pengesahan

Langkah-langkah pengesahan taburan pembolehubah sasaran dalam set data kepada set latihan, set ujian dan set pengesahan ditunjukkan dalam rajah 5.1. Proses pemilihan ciri digunakan untuk mengenal pasti simptom dan indikator yang berkemungkinan besar

berpunca daripada diabetes. Berikutan itu, ciri yang dipilih akan tertakluk kepada pembahagian set latihan, set ujian dan set pengesahan dan lajur "Umur" akan tertakluk kepada MinMaxScaler, yang menolak nilai minimum ciri dan kemudian membahagikan dengan julat untuk menukar ciri dengan menskala setiap ciri kepada julat tertentu.

```
Random Forest

[ ] # Random Forest
rf = RandomForestClassifier(criterion='gini',n_estimators=100)
rf.fit(X_train,y_train)

RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
criterion='gini', max_depth=None, max_features='auto',
max_leaf_nodes=None, max_samples=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, n_estimators=100,
n_jobs=None, oob_score=False, random_state=None,
verbose=0, warm_start=False)

[ ] # Cross Validation
kfold = model_selection.KFold(n_splits=10, random_state=7, shuffle=True)
scoring = 'accuracy'

acc_rf = cross_val_score(estimator = rf, X = X_train, y = y_train, cv = kfold,scoring=scoring)
acc_rf.mean()

0.9662020905923348

# Model Evaluation
y_predict_r = rf.predict(X_test)
roc=roc_auc_score(y_test, y_predict_r)
acc = accuracy_score(y_test, y_predict_r)
prec = precision_score(y_test, y_predict_r)
rec = recall_score(y_test, y_predict_r)
f1 = f1_score(y_test, y_predict_r)

model_results = pd.DataFrame([['Random Forest',acc, acc_rf.mean(),prec,rec, f1,roc]], columns = ['Model', 'Accuracy', 'Cross Val Accuracy', 'Precision', 'Recall', 'F1 Score', 'ROC'])
results = results.append(model_results, ignore_index = True)
results
```

	Model	Accuracy	Cross Val Accuracy	Precision	Recall	F1 Score	ROC
0	Logistic Regression	0.942308	0.894251	0.926471	0.984375	0.954545	0.929688
1	Random Forest	0.990385	0.966202	0.984615	1.000000	0.992248	0.987500

Rajah 5.2 Kod sumber untuk melatih model pembelajaran mesin hutan rawak (*Random Forest*)

Algoritma hutan rawak (*Random Forest*) digunakan untuk melatih set data, seperti yang dilihat dalam rajah di atas. Dari segi ketepatan, ketepatan pengesahan silang (*cross validation accuracy*), ingatan tepat (*precision recall*), Skor F1, dan ROC, keputusan keseluruhan hutan rawak (*Random Forest*) adalah sangat baik.

Jupyter

Open JupyterLab Quit Logout

Files Running Clusters SageMaker Examples Conda

Select items to perform actions on them. Upload New

	Name	Last Modified	File size
<input type="checkbox"/>	data	4 days ago	
<input type="checkbox"/>	AWS_Train.ipynb	Running 4 days ago	115 kB
<input type="checkbox"/>	startup_prediction.py	4 days ago	3.57 kB

Rajah 5.3 Antara muka di AWS yang menyimpan fail

```
In [21]: import sagemaker
        from sagemaker import get_execution_role

In [22]: sagemaker_session = sagemaker.Session()

In [23]: # Get a SageMaker-compatible role used by this Notebook Instance.
        role = get_execution_role()

In [24]: role
Out[24]: 'arn:aws:iam::125307025009:role/service-role/AmazonSageMaker-ExecutionRole-20220510T162230'

In [25]: train_input = sagemaker_session.upload_data("data")

In [26]: train_input
Out[26]: 's3://sagemaker-us-east-1-125307025009/data'

In [27]: from sagemaker.sklearn.estimator import SKLearn

        script_path = 'startup_prediction.py'

        sklearn = SKLearn(
            entry_point=script_path,
            instance_type="ml.m4.xlarge",
            framework_version="0.20.0",
            py_version="py3",
            role=role,
            sagemaker_session=sagemaker_session)

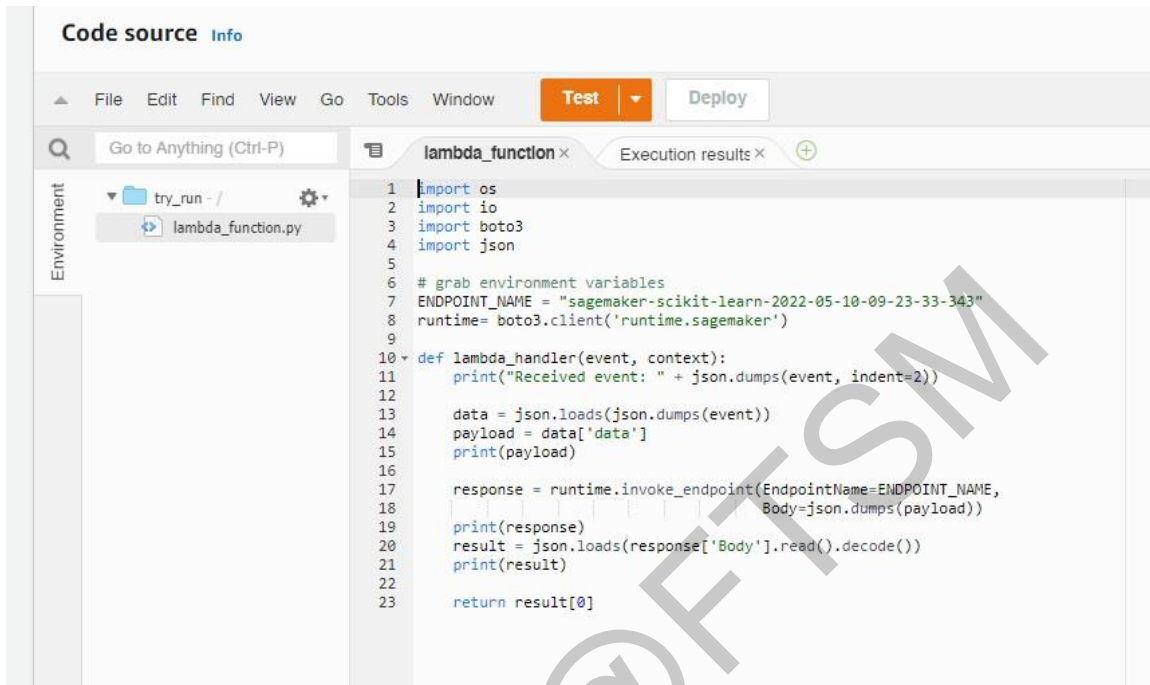
In [28]: sklearn.fit({'train': train_input})

2022-05-10 09:18:48 Starting - Starting the training job...ProfilerReport-1652174328: InProgress
...
2022-05-10 09:19:49 Starting - Preparing the instances for training.....
2022-05-10 09:21:11 Downloading - Downloading input data...
2022-05-10 09:21:49 Training - Downloading the training image.....
2022-05-10 09:22:49 Uploading - Uploading generated training model2022-05-10 09:22:40,218 sagemaker-containers INFO Imported framework sagemaker_sklearn_container.training
2022-05-10 09:22:40,222 sagemaker-training-toolkit INFO No GPUs detected (normal if no gpus installed)
2022-05-10 09:22:40,235 sagemaker_sklearn_container.training INFO Invoking user training script.
2022-05-10 09:22:40,776 sagemaker-training-toolkit INFO No GPUs detected (normal if no gpus installed)
2022-05-10 09:22:40,790 sagemaker-training-toolkit INFO No GPUs detected (normal if no gpus installed)

In [29]: deployment = sklearn.deploy(initial_instance_count=1, instance_type="ml.m4.xlarge")
        .....!

In [30]: deployment.endpoint
```

Rajah 5.4 Antara muka di AWS kod pembelajaran mesin hutan rawak (Random Forest)



```

Code source Info
File Edit Find View Go Tools Window Test Deploy
Go to Anything (Ctrl-P) lambda_function x Execution results x
Environment
try_run - /
lambda_function.py
1 import os
2 import io
3 import boto3
4 import json
5
6 # grab environment variables
7 ENDPOINT_NAME = "sagemaker-scikit-learn-2022-05-10-09-23-33-343"
8 runtime= boto3.client('runtime.sagemaker')
9
10 def lambda_handler(event, context):
11     print("Received event: " + json.dumps(event, indent=2))
12
13     data = json.loads(json.dumps(event))
14     payload = data['data']
15     print(payload)
16
17     response = runtime.invoke_endpoint(EndpointName=ENDPOINT_NAME,
18                                     Body=json.dumps(payload))
19     print(response)
20     result = json.loads(response['Body'].read().decode())
21     print(result)
22
23     return result[0]

```

Rajah 5.5 Antara muka di AWS kod sumber di AWS Lambda

Rajah 5.3 menunjukkan fail model pembelajaran mesin dan set data yang telah dimuatkan ke atas Awan Aws. Rajah 5.4 menunjukkan kod pembelajaran mesin hutan rawak (*Random Forest*) yang telah siap dilatih telah dimuat naik ke atas *Amazon Sage Maker* yang berfungsi untuk menyimpan model pembelajaran mesin yang siap dilatih. Rajah 5.5 menunjukkan kod di *AWS Lambda* yang berfungsi untuk membuat ramalan ke atas jawapan yang dipilih oleh pengguna dan menghantar keputusan.

Kualiti ramalan algoritma Keputusan telah digunakan menggunakan laporan klasifikasi (Classification Report). Ketepatan (Precision), recall, dan skor F1 bagi setiap metrik klasifikasi utama kelas dinyatakan dalam laporan ini. Metrik ini dikira dengan menggunakan positif dan negatif benar (*True Positive and True Negative*), positif dan negatif salah (*False Positive and False Negative*).

Laporan ini menunjukkan model ini mendapat ketepatan 99% pada set data ujian sebanyak 104 rekod. Laporan ini menunjukkan model ini terdapat ketepatan yang amat baik. Rajah 5.6 menunjukkan laporan klasifikasi model ini.

```
print(classification_report(y_test, y_predict))
```

	precision	recall	f1-score	support
0	1.00	0.97	0.99	40
1	0.98	1.00	0.99	64
accuracy			0.99	104
macro avg	0.99	0.99	0.99	104
weighted avg	0.99	0.99	0.99	104

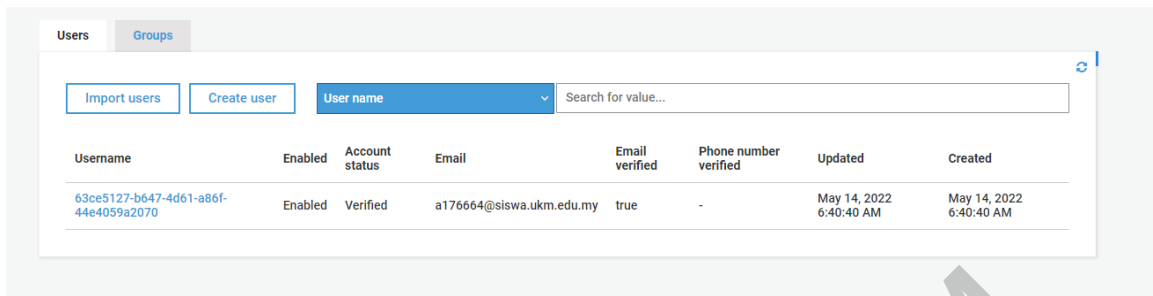
Rajah 5.6 Laporan Klasifikasi (Classification Report)

5.2 Pembangunan Aplikasi

Perisian *Flutter* dan bahasa *Dart* akan digunakan untuk membangunkan aplikasi mudah alih ini. Aplikasi ini akan dibangunkan dengan menggunakan *AWS Amplify* kerana servis ini mudah untuk digunakan dan terdapat antaramuka yang moden untuk dipilih. *AWS Command Line Interface* telah digunakan untuk menyambung projek ini dengan perkhidmatan *Amplify*.

Amazon Cognito

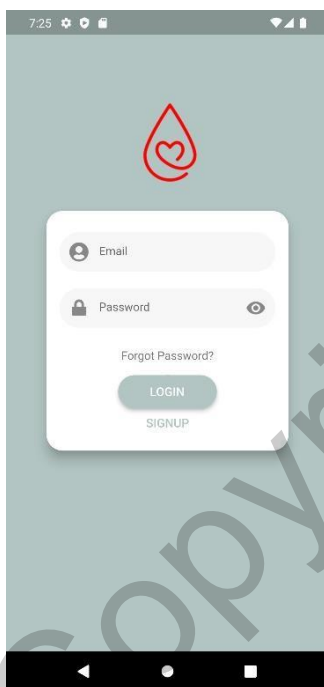
Platform *Amazon Cognito* memberikan pengesahan (*authentication*), kebenaran (*authorization*), dan pengurusan pengguna (*user management*) untuk aplikasi web dan aplikasi mudah alih. Pengguna boleh log masuk terus dengan nama pengguna dan kata laluan. Terdapat dua komponen utama dalam *Amazon Cognito* iaitu kumpulan pengguna dan kumpulan identiti. Kumpulan pengguna adalah direktori pengguna yang menyediakan pilihan pendaftaran dan log masuk untuk pengguna aplikasi anda. Pendaftaran aplikasi ini merangkumi Log masuk, Log keluar, Tukar kata laluan dan Lupa kata laluan. Rajah berikut merupakan contoh maklumat pengguna yang telah berjaya mendaftar.



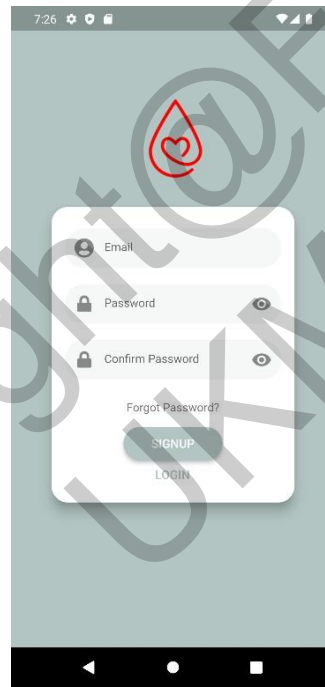
Username	Enabled	Account status	Email	Email verified	Phone number verified	Updated	Created
63ce5127-b647-4d61-a86f-44e4059a2070	Enabled	Verified	a176664@siswa.ukm.edu.my	true	-	May 14, 2022 6:40:40 AM	May 14, 2022 6:40:40 AM

Rajah 5.7 Antara muka di AWS setelah pengguna baru telah mendaftar akaun

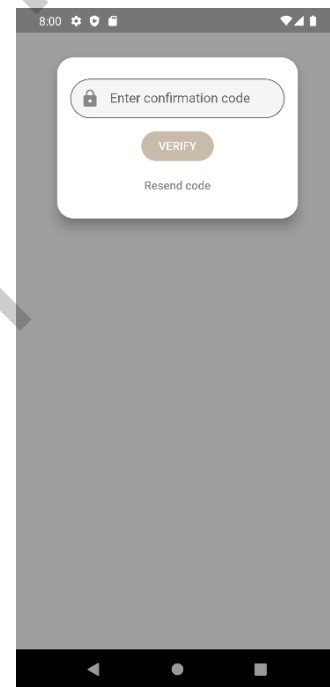
5.3 Antaramuka Aplikasi



Rajah 5.8 Antara muka log masuk



Rajah 5.9 Antara muka pendaftaran akaun



Rajah 5.10 Antara muka memasukkan OTP

Rajah 5.8, Rajah 5.9 serta Rajah 5.10 menunjukkan antara muka yang berkait dengan fungsi log masuk dan pendaftaran akaun. Rajah 5.8 merupakan antara muka yang pertama jika pengguna belum log masuk atau selepas log keluar dari sistem. Selepas pengguna memasukkan e-mel dan kata laluan yang telah berdaftar, aplikasi akan mengalih ke antara

muka halaman menu utama. Di Rajah 5.9, antara muka yang ditunjukkan adalah untuk pengguna baru untuk mendaftar akaun. Sekiranya pengguna siap mengisi maklumat yang perlu dengan betul, kod OTP akan dihantar ke e-mel yang diisi dan pengguna boleh memasukkan kod OTP yang diterima ke antara muka di Rajah 5.10. Selepas memasukkan kod OTP yang betul dan telah disahkan, pengguna akan berjaya log masuk dan mengalih ke antara muka halaman menu utama.



Rajah 5.11 Antara muka Halaman Utama



Rajah 5.12 Antara muka Soalan 10



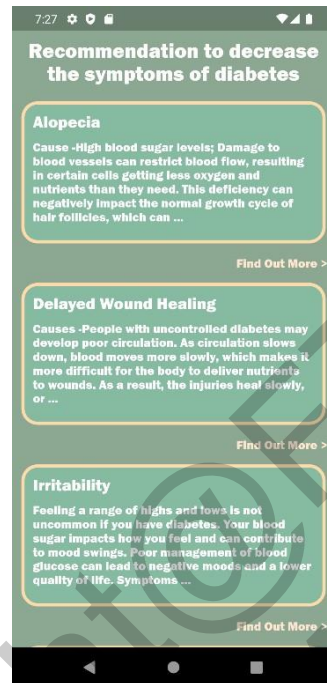
Rajah 5.13 Antara muka keputusan

Rajah 5.11, Rajah 5.12 serta Rajah 5.13 menunjukkan antara muka yang berkaitan dengan fungsi memeriksa risiko menghidap diabetes. Rajah 5.11 merupakan antara muka halaman utama selepas pengguna berjaya log masuk, Rajah 5.12 menunjukkan soalan yang ke-10 untuk pengguna. Terdapat sebanyak 10 soalan untuk dijawab oleh pengguna untuk menjalankan ramalan untuk memberi keputusan kepada pengguna yang ditunjukkan di Rajah 5.13. Jawapan yang dipilih oleh pengguna akan dihantar dari aplikasi mudah alih ke pengkomputeran *AWAN AWS* melalui *REST API* yang berfungsi sebagai jambatan penyambung untuk menjalankan ramalan menggunakan model pembelajaran mesin yang

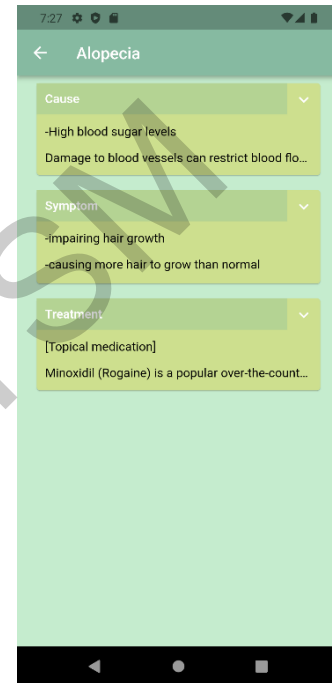
dihoskan di *AWAN AWS*. Keputusan yang didapati akan dihantar balik dari *AWS* ke aplikasi mudah alih.



Rajah 5.14 Antara muka cadangan



Rajah 5.15 Antara muka cadangan untuk mengurangkan simptom diabetes

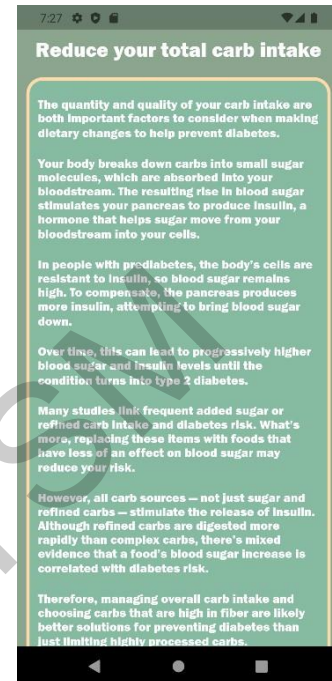


Rajah 5.16 Antara muka cadangan yang teliti berkaitan gejala

Rajah 5.14, Rajah 5.15 serta Rajah 5.16 menunjukkan antara muka yang berkait dengan fungsi memberi cadangan untuk mengurangkan simptom diabetes. Rajah 5.14 menunjukkan antara muka cadangan untuk dipilih oleh pengguna dengan pilihan “Ya” atau “Tidak”. Sekiranya pengguna memilih “Ya”, Rajah 5.15 merupakan antara muka cadangan untuk mengurangkan simptom diabetes. Untuk mendapat butiran yang lebih lanjut, pengguna boleh menekan butang “Find Out More” untuk mengalih ke Rajah 5.16 yang merupakan antara muka cadangan yang teliti berkaitan gejala. Dalam antara muka tersebut, pengguna akan mendapat cadangan yang teliti berkaitan dengan symptom yang dipilih.



Rajah 5.17 Antara muka cadangan yang teliti



Rajah 5.18 Antara muka cadangan untuk mengurangkan risiko menghidap diabetes

Rajah 5.17 dan Rajah 5.18 menunjukkan antara muka yang berkait dengan fungsi memberi cadangan untuk mengurangkan risiko diabetes. Sekiranya pengguna memilih “Tidak”, Rajah 5.17 merupakan antara muka cadangan untuk mengurangkan risiko menghidap diabetes. Untuk mendapat butiran yang lebih lanjut, pengguna boleh menekan butang “Find Out More” untuk mengalih ke Rajah 5.18 yang merupakan antara muka cadangan yang teliti. Dalam antara muka tersebut, pengguna akan mendapat cadangan yang teliti berkaitan dengan cadangan yang dipilih.

6 KESIMPULAN

Secara kesimpulan, bab ini memberi gambaran keseluruhan tentang projek serta kandungan dokumen ini. Selain itu, kekangan bagi aplikasi yang akan dibangunkan tersebut serta penambahbaikan untuk aplikasi juga dibincang

7 RUJUKAN

Buildfire. 2021. Mobile App Download Statistics & Usage Statistics. <https://buildfire.com/app-statistics/>

E.Olsen. 2021. Digital health apps balloon to more than 350,000 available on the market, according to IQVIA report. <https://www.mobihealthnews.com/news/digital-health-apps-balloon-more-350000-available-market-according-iqvia-report>

Patterson, C. C. et al. 2019. Trends and cyclical variation in the incidence of childhood type 1 diabetes in 26 European centres in the 25 year period 1989–2013: a multicentre prospective registration study. *Diabetologia* 62, 408–417.

Wang, L. et al. 2017 Prevalence and ethnic pattern of diabetes and prediabetes in China in 2013. *JAMA* 317, 2515–2523.

Dwyer-Lindgren, L., Mackenbach, J. P., Van Lenthe, F. J., Flaxman, A. D. & Mokdad, A. H. 2016. Diagnosed and undiagnosed diabetes prevalence by County in the U.S., 1999–2012. *Diabetes Care* 39, 1556–1562.

Bragg, F. et al. 2017. Association between diabetes and cause-specific mortality in rural and urban areas of China. *JAMA* 317, 280–289.

Policardo, L. et al. 2015. Effect of diabetes on hospitalization for ischemic stroke and related in-hospital mortality: a study in Tuscany, Italy, over years 2004–2011. *Diabetes. Metab. Res. Rev.* 31, 280–286.

Lin, X., Xu, Y., Pan, X. et al. 2020. Global, regional, and national burden and trend of diabetes in 195 countries and territories: an analysis from 1990 to 2025. *Sci Rep* 10, 14790 <https://doi.org/10.1038/s41598-020-71908-9>

International Diabetes Federation. 2019. Worldwide toll of diabetes. <https://www.diabetesatlas.org/en/sections/worldwide-toll-of-diabetes.html>

Ann Pietrangelo. 2020. The Effects of Diabetes on Your Body. <https://www.healthline.com/health/diabetes/effects-on-body>

Centers for Disease Control and Prevention, 2021. Diabetes Symptoms. <https://www.cdc.gov/diabetes/basics/symptoms.html>

A.Ramachandran. 2014. Know the signs and symptoms of diabetes. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4311308/pdf/IJMR-140-579.pdf>

Nathaniel G. Clark. et al. 2007. Symptoms of Diabetes and Their Association With the Risk and Presence of Diabetes. *Diabetes Care* 2007 Nov; 30(11): 2868-2873.

Coffman, Maren et al. 2012. Diabetes symptoms, health literacy, and health care use in adult Latinos with diabetes risk factors. *Journal of cultural diversity*. 19. 4-9.

Aoife Garrahy et al. 2018. Diagnosis and management of central diabetes insipidus in adults <https://onlinelibrary.wiley.com/doi/full/10.1111/cen.13866>

Dr. V.Vasudha Rani et al. 2021. Significance of Multilayer Perceptron Model for Early Detection of Diabetes over ML Methods <https://jusst.org/wp-content/uploads/2021/08/Significance-Of-Multilayer-Perceptron-Model-For-Early-Detection-Of-Diabetes-Over-ML-Methods.pdf>

Su L et al. 2013. Association of Androgenetic Alopecia With Mortality From Diabetes Mellitus and Heart Disease. *JAMA Dermatol.* 2013;149(5):601–606. doi:10.1001/jamadermatol.2013.130

Maryam Ferdousi et.al. 2021. Diagnosis of Neuropathy and Risk Factors for Corneal Nerve Loss in Type 1 and Type 2 Diabetes: A Corneal Confocal Microscopy Study. *Diabetes Care* 1 January 2021; 44 (1): 150–156. <https://doi.org/10.2337/dc20-1482>