Nama : Fiqhri Mulianda Putra
NIM : G651180161
Mata Kuliah : Sistem Pakar dan Sistem Penunjang Keputusan

Sistem Pakar untuk Mendiagnosis Penyakit Jantung menggunakan Algoritma Random Forest

Latar Belakang

Fungsi inti dari penambangan data menerapkan berbagai metode dan algoritma untuk menemukan dan mengekstrak pola data yang disimpan. Dari dua dekade terakhir aplikasi data mining dan penemuan pengetahuan telah mendapatkan fokus yang kaya karena signifikansinya dalam pengambilan keputusan dan telah menjadi komponen penting dalam berbagai organisasi. Bidang penambangan data telah makmur dan diposisikan ke area baru kehidupan manusia dengan berbagai integrasi dan kemajuan di bidang Statistik, Database, Pembelajaran Mesin, Reorganisasi Pola dan perawatan kesehatan.

Penambangan Data Medis dalam layanan kesehatan dianggap sebagai tugas penting namun rumit yang perlu dijalankan secara akurat dan efisien. Penambangan data layanan kesehatan berupaya memecahkan masalah kesehatan dunia nyata dalam diagnosis dan pengobatan penyakit [6]. Makalah penelitian ini bertujuan untuk menganalisis beberapa teknik penambangan data yang diusulkan dalam beberapa tahun terakhir untuk diagnosis penyakit jantung. Banyak peneliti menggunakan teknik penambangan data dalam diagnosis penyakit seperti TBC, diabetes, kanker dan penyakit jantung di mana beberapa teknik penambangan data digunakan dalam diagnosis penyakit jantung seperti KNN, Neural Networks, klasifikasi Bayesian, Klasifikasi berdasarkan pengelompokan, Decision Tree, Genetic Algorithm, Naive Bayes, Decision tree, WAC yang menunjukkan akurasi pada level yang berbeda.

Dalam diagnosis penyakit jantung banyak pekerjaan dilakukan, para peneliti telah menyelidiki penggunaan teknik data mining untuk membantu para profesional. Banyak faktor risiko yang terkait dengan penyakit jantung seperti usia, jenis kelamin, nyeri dada, tekanan darah, kolesterol, gula darah, riwayat keluarga dengan penyakit jantung, obesitas, dan kurang aktivitas fisik. Pengetahuan tentang faktor-faktor risiko ini para profesional medis dapat mendiagnosis penyakit jantung pada pasien dengan mudah. Maka dibutuhkan sebuah sistem pakar untuk mengakomodir antara pengetuan pakar dengan penyakit jantung. Dikarenakan banyaknya parameter pada dataset. untuk pembuatan model mengugunakan algoritma Random Forest.

Heart Disease Diagnosis

Memprediksi adanya satu dari empat jenis penyakit jantung (atau tidak sama sekali) menggunakan data laporan tes medis pasien.

Dataset

Kumpulan data penyakit jantung diakses (https://archive.ics.uci.edu/ml/datasets/heart+Disease) terdiri dari data pasien dari Cleveland, Hongaria, Long Beach dan Swiss. Kumpulan data gabungan terdiri dari 14 fitur dan 916 sampel dengan banyak nilai yang hilang. Fitur yang digunakan di sini adalah:

1. age: The patients age in years
2. sex: The patients gender(1=male; 0=female)
3. cp: Chest pain type,
    *Value 1: typical angina
    *Value 2: atypical angina
    *Value 3: non-anginal pain
    *Value 4: asymptomatic
4. trestbps: Resting blood pressure (in mm Hg on admission to the hospital)
5. chol: Serum cholestoral in mg/dl
6. fbs: Fasting blood sugar > 120 mg/dl? (1=true; 0=false)
7. restecg: Resting electrocardiographic results
    *Value 0: normal
    *Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
    *Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria
8. thalach: Maximum heart rate achieved
9. exang: Chest pain(angina) after exercise? (1=yes; 0=no)
10. thal: Not described
    *Value 3=normal
    *Value 6=treated defect
    *Value 7=reversible defect
11. num: Target
    *Value 0: less than 50% narrowing of coronary arteries(no heart disease)
    *Value 1,2,3,4: >50% narrowing. The value indicates the stage of heart disease

Pembuat Dataset

1. Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.
2. University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.
3. University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.
4. V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D.

Perangkat yang dibutuhkan:

click==6.7

Flask==1.0.2
itsdangerous==0.24
Jinja2==2.10
MarkupSafe==1.0
numpy==1.15.1
pandas==0.23.4
python-dateutil==2.7.3
pytz==2018.5
scikit-learn==0.19.2
six==1.11.0
sklearn==0.0
Werkzeug==0.14.1

Cara menjalankan aplikasi web secara local:
- Install requirements
  `pip install -r requirements.txt`
- Run flask web app
  `python main_file.py`

Model dan Akurasi
Random Forest mencapai akurasi klasifikasi multi-kelas rata-rata 56-60% (183 sampel uji). Sedangkan akurasi klasifikasi biner rata-rata 75-80% (penyakit jantung atau tidak ada penyakit jantung).

Berikut ini adalah lampiran sourcecode dan screenshoot aplikasi yang dijalankan:
Preprocess.py
Berisi kodingan untuk mengolah dataset yang ada dari seleksi data sampai pembersihan data.

```
import pandas as pd

import numpy as np
import os
path = os.path.dirname(__file__)
path1 = os.path.join(path, 'dataset/processed_cleveland.csv')
path2 = os.path.join(path, 'dataset/processed_hungarian.csv')
path3 = os.path.join(path, 'dataset/processed_switzerland.csv')
path4 = os.path.join(path, 'dataset/processed_va.csv')

df1 = pd.read_csv(path1)
df2 = pd.read_csv(path2)
df3 = pd.read_csv(path3)
```

```
df4 = pd.read_csv(path4)

df1 = df1.replace('?', np.nan)
df2 = df2.replace('?', np.nan)
df3 = df3.replace('?', np.nan)
df4 = df4.replace('?', np.nan)

col = ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg',
       'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'num']
df1.columns = col
df2.columns = col
df3.columns = col
df4.columns = col

print("Cleveland data. Size={}\nNumber of missing values".format(df1.shape))
print(df1.isna().sum())

print("\nHungary data:. Size={}\nNumber of missing values".format(df2.shape))
print(df2.isna().sum())

print("\nSwitzerland data. Size={}\nNumber of missing values".format(df3.shape))
print(df3.isna().sum())

print("\nV.A Long Beach data. Size={}\nNumber of missing values".format(df4.shape))
print(df4.isna().sum())

df = pd.concat([df1, df2, df3, df4])
df=df.fillna(df.median())

df=df.drop(['oldpeak', 'slope','ca', 'thal'], axis=1)
print("Concatanated dataset. Size={}\nNumber of missing values".format(df.shape))
print(df.isna().sum())

df.to_csv(os.path.join(path, 'recons_dataset/combined_dataset.csv'), index=False)
```

Main_file.py

Berisi kodingan untuk membuat tampilan web dalam memprediksi penyakit jantung dengan menggunakan Flask.

```
from flask import Flask, render_template, url_for, request
from sklearn.externals import joblib
import os
import numpy as np
import pickle
```

```
app = Flask(__name__, static_folder='static')

@app.route("/")
def index():
    return render_template('home.html')


@app.route('/result', methods=['POST', 'GET'])
def result():
    age = int(request.form['age'])
    sex = int(request.form['sex'])
    trestbps = float(request.form['trestbps'])
    chol = float(request.form['chol'])
    restecg = float(request.form['restecg'])
    thalach = float(request.form['thalach'])
    exang = int(request.form['exang'])
    cp = int(request.form['cp'])
    fbs = float(request.form['fbs'])
    x = np.array([age, sex, cp, trestbps, chol, fbs, restecg,
                  thalach, exang]).reshape(1, -1)

    scaler_path = os.path.join(os.path.dirname(__file__), 'models/scaler.pkl')
    scaler = None
    with open(scaler_path, 'rb') as f:
        scaler = pickle.load(f)

    x = scaler.transform(x)

    model_path = os.path.join(os.path.dirname(__file__), 'models/rfc.sav')
    clf = joblib.load(model_path)

    y = clf.predict(x)
    print(y)

    # No heart disease
    if y == 0:
        return render_template('nodisease.html')

    # y=1,2,4,4 are stages of heart disease
    else:
        return render_template('heartdisease.htm', stage=int(y))


@app.route('/about')
def about():
    return render_template('about.html')
```

```
if __name__ == "__main__":
    app.run(debug=True)
```

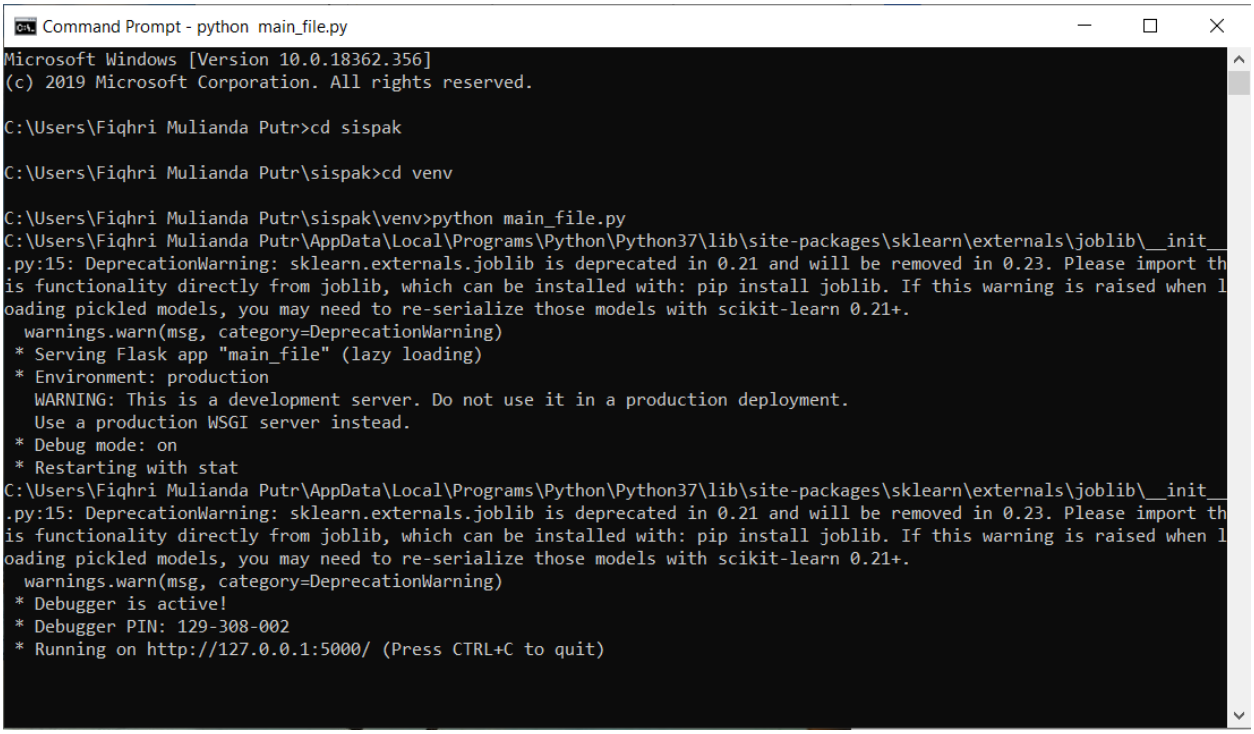
Model.py

Berisi kodingan untuk pembagian data, pembuatan model inferensi random forest kemudian menyimpan model sehingga bisa di akses melalui web menggunakan Flask. Adapun model disimpan didalam file dengan nama ref.sav .

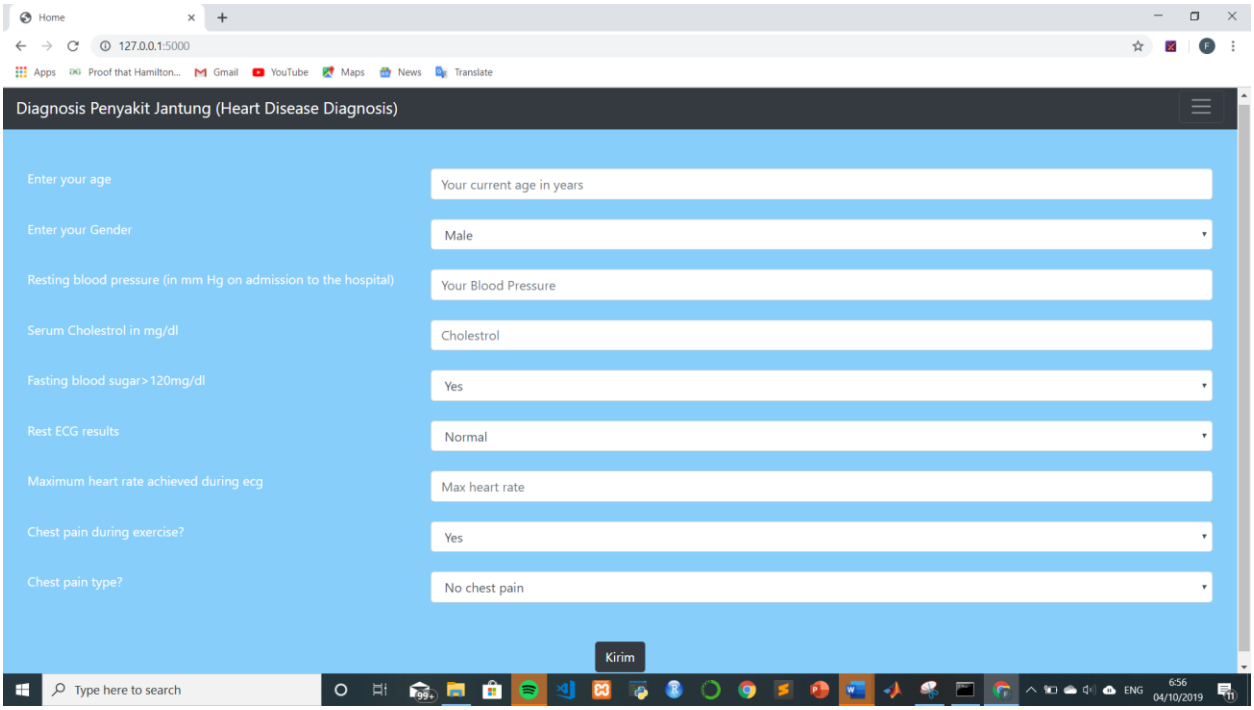
```
from sklearn.svm import SVC, LinearSVC
from sklearn.preprocessing import MinMaxScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.externals import joblib
import numpy as np
import pandas as pd
import pickle
import os
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

root = os.path.dirname(__file__)
path_df = os.path.join(root, 'recons_dataset/combined_dataset.csv')
data = pd.read_csv(path_df)

scaler = MinMaxScaler()
train, test = train_test_split(data, test_size=0.25)

X_train = train.drop('num', axis=1)
Y_train = train['num']

X_test = test.drop('num', axis=1)
Y_test = test['num']

# We don't scale targets: Y_test, Y_train as SVC returns the class labels not probability values
X_train = scaler.fit_transform(X_train)
X_test = scaler.fit_transform(X_test)

clf = RandomForestClassifier()

# Training the classifier
clf.fit(X_train, Y_train)
```

```
# Testing model accuracy. Average is taken as test set is very small hence accuracy varies a lot everytime the model is trained
acc = 0
acc_binary = 0
for i in range(0, 20):
    Y_hat = clf.predict(X_test)
    Y_hat_bin = Y_hat>0
    Y_test_bin = Y_test>0
    acc = acc + accuracy_score(Y_hat, Y_test)
    acc_binary = acc_binary +accuracy_score(Y_hat_bin, Y_test_bin)

print("Average test Accuracy:{}".format(acc/20))
print("Average binary accuracy:{}".format(acc_binary/20))

# Saving the trained model for inference
model_path = os.path.join(root, 'models/rfc.sav')
joblib.dump(clf, model_path)

# Saving the scaler object
scaler_path = os.path.join(root, 'models/scaler.pkl')
with open(scaler_path, 'wb') as scaler_file:
    pickle.dump(scaler, scaler_file)
```

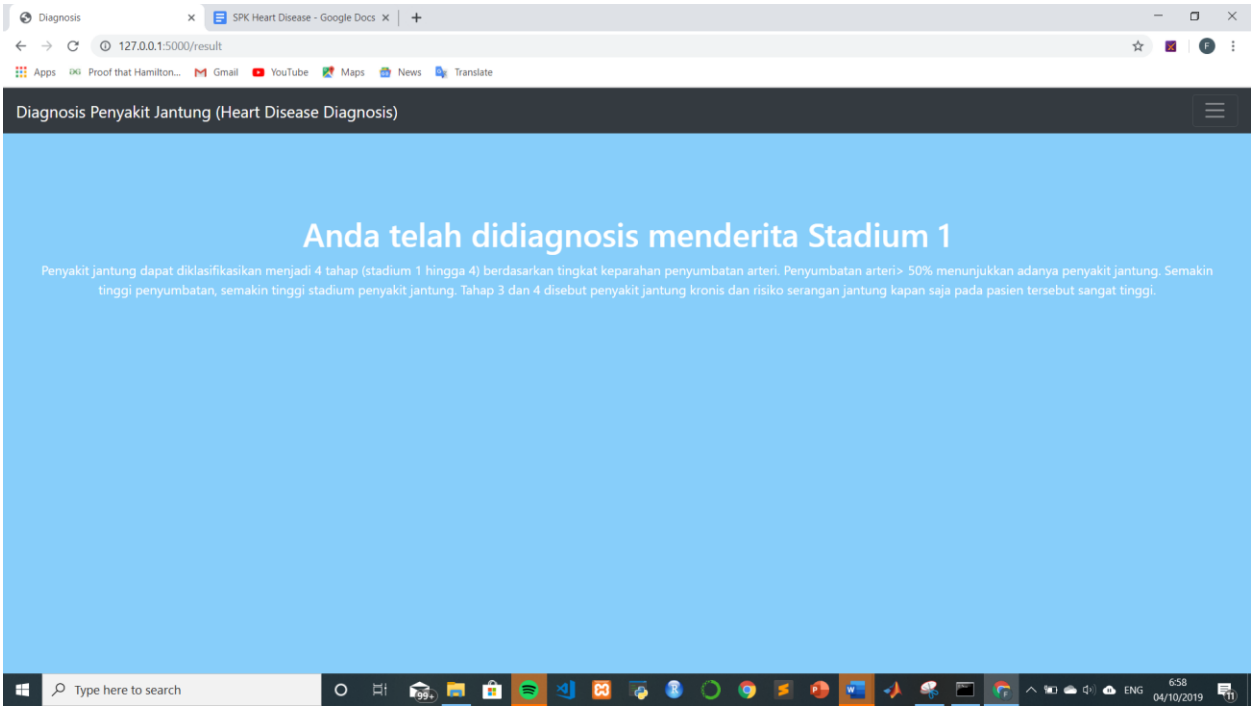
Tampilan menjalankan program di cmd



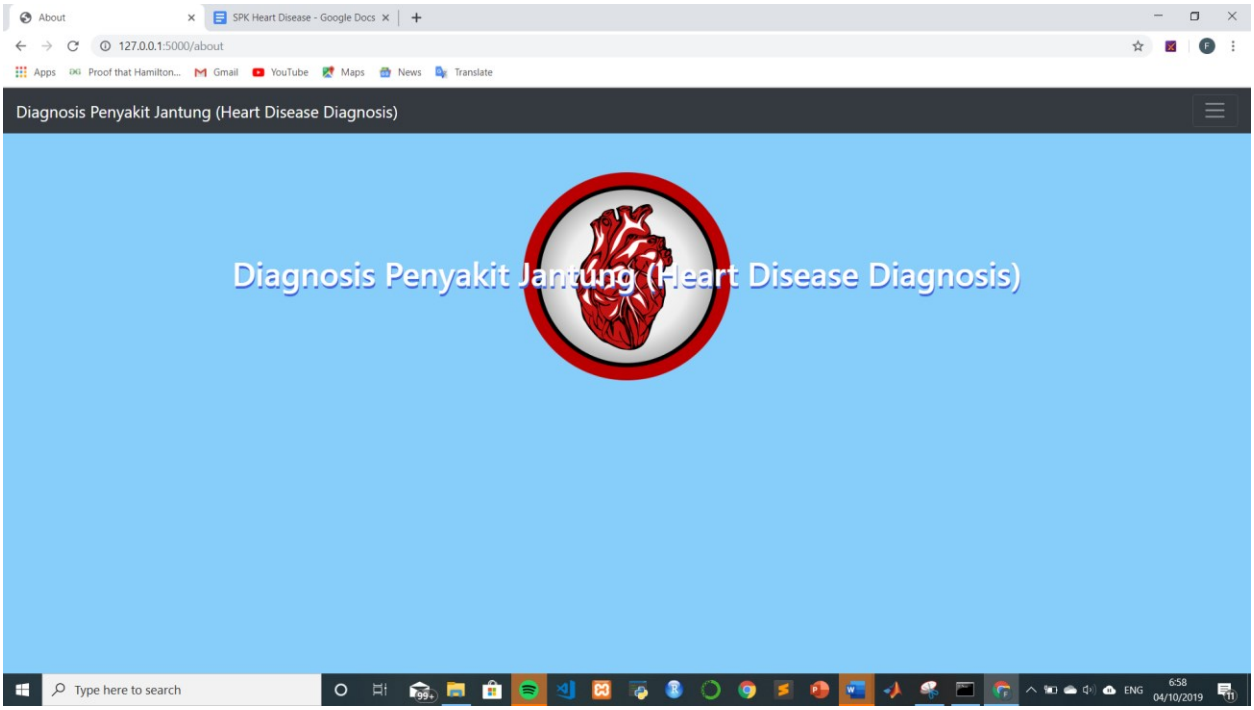
Menu Home



Tampilan Hasil Diagnosis



Menu About



Kesimpulan

Dari pembuatan model sampai penjalanan aplikasi dapat disimpulkan bahwa model sudah dapat digunakan dikarenakan akurasi sudah mencapai 80% disertai dengan pengawasan dokter untuk pengambilan keputusan akhir. Kemudian aplikasi yang dibuat sudah berjalan sesuai keinginan dan terintegrasi antara model dan web.