LAPORAN PROYEK KECERDASAN BUATAN

“Gender Classification”

DOSEN

Liliana, S.T., M.Eng., Ph.D.

DISUSUN OLEH

Michael Dennis	C14180190
Ferdinant Pangestu	C14180197
Alfred Chandra	C14180206
Aaron Davis	C14180216

UNIVERSITAS KRISTEN PETRA SURABAYA
FAKULTAS TEKNOLOGI INDUSTRI
TEKNIK INFORMATIKA

2020/2021

A. Penjelasan Ide

Suatu sistem yang dapat memprediksi jenis kelamin seseorang berdasarkan atribut-atribut yang diberikan. Sistem tersebut dapat mempelajari pola dari dataset yang sudah ada, sehingga terdapat proses pelatihan sistem terlebih dahulu sebelum yang gunanya untuk melihat tingkat akurasi dari prediksi sistem yang sudah kita buat berdasarkan data train dan data test yang kita tentukan sendiri nilainya. Setelah mengetahui nilai akurasi dari prediksi yang kita sudah buat, maka sistem memberikan beberapa input kepada user yang dimana inputan tersebut adalah pertanyaan-pertanyaan mengenai atribut-atribut.

B. Metode

Proses pengerjaan proyek ini menggunakan metode Decision Tree Classifier. Konsep dari decision tree adalah mengubah data menjadi aturan-aturan keputusan. Manfaat utama dari penggunaan decision tree adalah kemampuannya untuk mem-break down proses pengambilan keputusan yang kompleks menjadi lebih simple, sehingga pengambil keputusan akan lebih menginterpretasikan solusi dari permasalahan.

Dikarenakan proyek ini memerlukan alur atau pola untuk menghasilkan suatu output yaitu hasil prediksi. Maka dalam proyek ini, penerapan Decision Tree Classifier akan diterapkan guna memperoleh hasil prediksi sesuai dengan pola yang sudah dibuat.

Selain daripada itu, proyek ini menggunakan bahasa pemrograman Python dan menggunakan google colab sebagai sarana untuk melakukan coding. Dalam Python sendiri terdapat banyak library dan pada proyek ini menggunaka library :

- Pandas
- IO
- Sklearn :
    - DecisionTreeClassifier
    - Train_test_split
    - Accuracy_score
    - Export_graphviz
- Six
- Ipython
- pydotplus

C. Cara Kerja

1. Input Data

```
[ ]  from google.colab import files
     uploaded = files.upload()
```

Include dari sistem bawaan google colab untuk import file dan untuk mengupload file dari PC kita.


```
[ ]  import pandas as pd
     import io
     data = pd.read_csv(io.BytesIO(uploaded['gender.csv']))
     data
```

Untuk membaca file yang sudah kita masukkan tadi. Terdapat import io untuk mengkonversi data csv menjadi byte.

2. Pre-Processing

```
[ ]  data['Favorite Color'].unique()
```

Mencari nilai unik dari atribut tersebut. Tujuannya untuk mempermudah proses pengubahan string dari isi dalam data menjadi angka.

```
[ ]  for i in range(66):
         if data['Favorite Color'][i] == "Cool":
             data['Favorite Color'][i] = 1
         elif data['Favorite Color'][i] == "Neutral":
             data['Favorite Color'][i] = 2
         elif data['Favorite Color'][i] == "Warm":
             data['Favorite Color'][i] = 3
```

Mengubah data yang nilainya sesuai di IF menjadi angka. Tujuan dari proses ini agar nilai-nilai dalam data menjadi angka agar dapat diproses dalam library DecisionTreeClassifier.

```
[ ]  data['Favorite Music Genre'].unique()
```

```
array(['Rock', 'Hip hop', 'Folk/Traditional', 'Jazz/Blues', 'Pop',
       'Electronic', 'R&B and soul'], dtype=object)
```

```
[ ]  for i in range(66):
        if data['Favorite Music Genre'][i] == "Rock":
            data['Favorite Music Genre'][i] = 1
        elif data['Favorite Music Genre'][i] == "Hip hop":
            data['Favorite Music Genre'][i] = 2
        elif data['Favorite Music Genre'][i] == "Folk/Traditional":
            data['Favorite Music Genre'][i] = 3
        elif data['Favorite Music Genre'][i] == "Jazz/Blues":
            data['Favorite Music Genre'][i] = 4
        elif data['Favorite Music Genre'][i] == "Pop":
            data['Favorite Music Genre'][i] = 5
        elif data['Favorite Music Genre'][i] == "Electronic":
            data['Favorite Music Genre'][i] = 6
        elif data['Favorite Music Genre'][i] == "R&B and soul":
            data['Favorite Music Genre'][i] = 7
```

```
[ ]  data['Favorite Beverage'].unique()
```

```
array(['Vodka', 'Wine', 'Whiskey', "Doesn't drink", 'Beer', 'Other'],
      dtype=object)
```

```
[ ]  for i in range(66):
        if data['Favorite Beverage'][i] == "Vodka":
            data['Favorite Beverage'][i] = 1
        elif data['Favorite Beverage'][i] == "Wine":
            data['Favorite Beverage'][i] = 2
        elif data['Favorite Beverage'][i] == "Whiskey":
            data['Favorite Beverage'][i] = 3
        elif data['Favorite Beverage'][i] == "Doesn't drink":
            data['Favorite Beverage'][i] = 4
        elif data['Favorite Beverage'][i] == "Beer":
            data['Favorite Beverage'][i] = 5
        elif data['Favorite Beverage'][i] == "Other":
            data['Favorite Beverage'][i] = 6
```

```
[ ]  data['Favorite Soft Drink'].unique()
```

```
array(['7UP/Sprite', 'Coca Cola/Pepsi', 'Fanta', 'Other'], dtype=object)
```

```
[ ]  for i in range(66):
         if data['Favorite Soft Drink'][i] == "7UP/Sprite":
             data['Favorite Soft Drink'][i] = 1
         elif data['Favorite Soft Drink'][i] == "Coca Cola/Pepsi":
             data['Favorite Soft Drink'][i] = 2
         elif data['Favorite Soft Drink'][i] == "Fanta":
             data['Favorite Soft Drink'][i] = 3
         elif data['Favorite Soft Drink'][i] == "Other":
             data['Favorite Soft Drink'][i] = 4
```

3. Data Training

```
[ ]  from sklearn.tree import DecisionTreeClassifier as dtc
     from sklearn.model_selection import train_test_split
     from sklearn.metrics import accuracy_score

     X = data.drop(columns=['Gender'])
     y = data['Gender']
     X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.15)
```

Meng-*assign* nilai X agar menampung semua data kecuali atribut gender → merupakan data atribut.

Meng-assign nilai Y agar mengambil nilai gender → merupakan data class.

Setelah itu memberikan variabel-variabel seperti X_train, X_test, y_train, dan y_test untuk dijadikan variabel yang menampung sebagian data untuk diuji dan ada yang untuk dilatih. Dalam proses ini, data test adalah 0.15 atau 15% dari total keseluruhan data.

4. Testing Akurasi

```
model = dtc(max_depth=5, criterion="entropy")
model = model.fit(X_train,y_train)
predictions = model.predict(X_test)

coba = accuracy_score(y_test, predictions)
print("Akurasi Prediksi :",coba)
```

```
Akurasi Prediksi : 0.6
```

Membuat variabel bernama model untuk menampung proses DecisionTreeClassifier. Maksimal akarnya adalah 5 dan kriterianya yaitu menggunakan entropy.

Setelah itu memasukkan nilai X_train dan y_train kedalam fungsi model. Setelah itu masukkan fungsi prediksi pada variabel predictions. Dan untuk menguji hasil prdeiksi tersebut dapat menggunakan fungsi accuracy_score. Hasil dari akurasi tersebut diprint sehingga dapat dilihat oleh user.

5. User Input

```
print("Favorite Color : 1.Cool 2.Neutral 3.Warm")
color = int(input("Pilih : "))

print("Favorite Music Genre : 1.Rock 2.Hip hop 3.Folk/Traditional 4.Jazz/Blues 5.Pop 6.Electronic 7.R&B and soul")
music = int(input("Pilih : "))

print("Favorite Beverage : 1.Vodka 2.Wine 3.Whiskey 4.Doesn't drink 5.Beer 6.Other")
beverage = int(input("Pilih : "))

print("Favorite Soft Drink : 1.7UP/Sprite 2.Coca Cola/Pepsi 3.Fanta 4.Other")
drink = int(input("Pilih : "))
```

Memberikan inputan yang nantinya akan diisi oleh user. Atribut sudah ditampilkan atau diprint sehingga user dapat memilih atribut dari pertanyaan yang ada sesuai nomor yang sudah diberi.

```
prediction = model.predict([ [color, music, beverage, drink] ])
if prediction[0] == 'M':
    print("Prediction Gender : Male")
elif prediction[0] == 'F':
    print("Prediction Gender : Female")
```

Setelah itu masukkan fungsi predict dalam variabel prediction guna untuk mendapat nilai class yang isinya antara *Male* atau *Female*.

6. Tampilan

```
from sklearn.tree import export_graphviz
from six import StringIO
from IPython.display import Image
import pydotplus
```

Ini merupakan library yang diperlukan untuk menampilkan tree dari proses yang sudah dilakukan diatas.

```
dot_data = StringIO()
export_graphviz(model, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True,feature_names = ['color', 'music', 'beverage', 'drink'],
                class_names=sorted(y.unique()))
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png())
```

Lalu kita membuat variabel dot_data yang isinya adalah StringIO() sehingga variabel tersebut bertibe IO.

Setelah itu terdapat export_graphviz yang di dalamnya terdapat model → variabel DecisionTreeClassifier, dot_data → data akhir atau hasilnya akan disimpan ke data tersebut, filled → visualisasi warna, rounded → visualisasi bentuk, special_characters → visualisasi tulisan yang khusus, feature_names → nilai atribut yang ada, dan class_names → nilai class.

Selanjutnya terdapat variabel graph yang mengambil nilai dari dot_data dan dilakukan visualisasi grafik. Lalu menggunakan fungsi image untuk membuat gambar.

D. Kesimpulan

Setelah melakukan proyek ini, dapat kami simpulkan bahwa jumlah dataset dapat mempengaruhi tingkat akurasi dari hasil train dan tes. Jumlah dataset yang semakin banyak memungkinkan nilai akurasi pada proses testing menjadi lebih tinggi, sedangkan pada dataset yang sedikit cenderung akurasi yang diperoleh acak atau *random* karena tergantung hasil prediksinya. Sangat sulit untuk mendapatkan nilai akurasi 80% keatas pada dataset yang saat ini kami gunakan.

Lalu setelah kami melakukan presentasi, dataset ini juga tidak sepenuhnya menggambarkan realita yang ada. Tidak semua atribut yang ada pada dataset ini bisa menjadi referensi jenis kelamin seseorang dalam beberapa negara. Hal ini menjadi masukkan bagi kelompok kami untuk melakukan research lebih dalam lagi kedepannya.