

Judul Proyek : Klasifikasi Agama Suatu Negara Berdasarkan Karakteristik Bendera Menggunakan Machine Learning dan Deep Learning

Nama Mahasiswa: Arenta Putri Maharani

NIM : 233307034

Program Studi : Teknologi Informasi

Mata Kuliah : Data Science

Dosen Pengampu : Gus Nanang Syaifuddiin, S.Kom., M.Kom.

Tahun Akademik : 2025/5

Link GitHub Repository : <https://github.com/arentapm/Proyek-Machine-Learning.git>

Link Vidio Pembahasan :

<https://drive.google.com/drive/folders/1Zk7q7y50MBDyAvmMCL3RCEDM7rAKR9fL?usp=sharing>

1. Learning Outcomes
2. Project Overview
- 2.1 Latar Belakang

Penerapan *machine learning* dalam analisis data sosial dan budaya menjadi semakin penting seiring meningkatnya ketersediaan dataset multivariat yang kompleks. Salah satu bentuk data yang menarik untuk dianalisis adalah bendera nasional, yang tidak hanya berfungsi sebagai identitas visual suatu negara, tetapi juga mempresentasikan nilai historis, budaya, dan keagamaan. Dengan memanfaatkan karakteristik visual dan atribut pendukung pada bendera, machine learning memungkinkan dilakukannya klasifikasi terhadap karakteristik sosial suatu negara, seperti agama dominan, secara sistematis, dan berbasis data (Kalampokas dkk., 2023). Pendekatan machine learning dan deep learning mampu mengekstraksi pola bermakna dari bendera nasional dan menghubungkannya dengan identifikasi.

Permasalahan umum dalam domain ini adalah keterbatasan penelitian yang secara eksplisit mengaitkan atribut visual simbolik dengan aspek sosial – keagamaan, serta tantangan pengelolaan data multivariat yang mengandung kombinasi fitur numerik, kategorikal, dan biner. Dataset Flags memiliki ukuran data yang relatif kecil dan distribusi kelas yang tidak seimbang, sehingga berpotensi menimbulkan masalah *overfitting* dan penurunan akurasi model. Selain itu, pemilihan algoritma klasifikasi yang tepat menjadi faktor penting agar hubungan antara simbol visual bendera dan agama dapat dipelajari secara optimal, dengan berbagai algoritma *machine learning* dapat digunakan untuk klasifikasi bendera, namun masih diperlukan pengembangan dan evaluasi lebih lanjut dengan fokus pada label sosial seperti agama.

Proyek ini bertujuan untuk membangun model *machine learning* yang mampu mengklasifikasikan agama dominan suatu negara berdasarkan karakteristik bendera dan atribut pendukung lainnya. Manfaat dari penelitian ini adalah memberikan pemahaman baru mengenai keterkaitan simbol visual dengan struktur sosial-keagamaan, menjadi

studi kasus pembelajaran *machine learning* pada data multivariat, serta memperkaya kajian akademik dalam analisis data sosial berbasis simbol(Jafarigol dkk., 2023). Serta *machine learning* efektif digunakan untuk mempelajari afiliasi agama dari data sosial berskala global, sehingga memperkuat relevansi pendekatan yang digunakan dalam proyek ini.

3. Business Understanding / Problem Understanding

3.1 Problem Statements

1. Dataset Flags memiliki karakteristik multivariat yang terdiri dari fitur numerik, kategorikal, dan biner, sehingga diperlukan metode preprocessing agar data dapat digunakan dengan baik untuk proses klasifikasi.
2. Hubungan antara karakteristik visual bendera dan agama dominan suatu negara tidak bersifat langsung, sehingga dibutuhkan model machine learning yang mampu mengenali pola dari kombinasi fitur fitur tersebut.
3. Ukuran dataset yang kecil relatif berpotensi menyebabkan model mengalami overfitting, sehingga perlu dilakukan pemilihan algoritma dan evaluasi model yang sesuai.

3.2 Goals

1. Membangun model machine learning untuk mengklasifikasikan agama dominan suatu negara berdasarkan karakteristik bendera menggunakan dataset Flags.
2. Mengevaluasi kinerja model klasifikasi menggunakan metrik evaluasi seperti akurasi, precision, recall, dan confusion matrix.
3. Menentukan model klasifikasi yang memberikan performa terbaik dan stabil pada dataset yang digunakan.

3.3 Solution Approach

Model 1 : baseline

Naive Bayes : menggunakan model ini dikarenakan dataset Flags berisi banyak fitur biner dan kategorikal, seperti warna, simbol, dan pola bendera, yang sesuai dengan asumsi dasar Naive Bayes dalam menghitung probabilitas berdasarkan fitur-fitur independen. Naive Bayes mampu memberikan performa yang cukup baik pada dataset kecil dan multivariat.

Model 2 : Advanced / ML Model

Random Forest : dapat menangani data tabular dengan karakteristik yang beragam, baik numerik maupaun kategorikal. Model ini bekerja dengan banyak pohon keputusan dan menggabungkan hasil prediksinya, sehingga mampu mengurangi risiko overfitting, terutama pada dataset kecil. Random Forest juga tidak memerlukan asumsi distribusi data tertentu dan cukup stabil terhadap noise, sehingga cocok digunakan untuk mengklasifikasikan agama negaar berdasarkan kombinasi atribut bendera yang kompleks.

Model 3 : Deep Learning

MLP (Multilayer Perceptron) : dikarenakan dataset yang digunakan berbentuk tabular, bukan sitra, teks, atau deret waktu. MLP sederhana untuk menangani data tabular serta mampu mempelajari hubungan non-linier antar fitur, MLP dapat menangkap pola yang lebih kompleks dibandingkan model machine learning konvensional.

4. Data Understanding

4.1 Informasi Dataset

Sumber dataset : <https://archive.ics.uci.edu/dataset/40/flags>

Deskripsi dataset :

1. Jumlah baris (rows) : 194
2. Jumlah kolom (columns/features) : 30
3. Tipe data : Tabular (numerik, kategorikal, dan biner)
4. Ukuran dataset : 15KB
5. Format file : data

4.2 Deskripsi Fitur

Nama Fitur	Tipe Data	Deskripsi	Contoh Nilai
Nama	Categorical	Nama Negara	Indonesia, France, Japan
Landmass	Integer	Benua tempat negara berada (1 = Amerika Utara, 3 = Eropa, 4 = Afrika, 5 = Asia, 6 = Oseania)	3, 5
Zone	Integer	Zone geografis berdasarkan Greenwich dan Equator (1 = NE, 2 = SE, 3 = SW, 4 = NW)	1, 4
Area	Integer	Luas wilayah negara dalam rubah km^2	238, 912
populationn	integer	Jumlah penduduk dalam jutaan	10, 270
language	integer	Bahasa utama negara (1 = english, 2 = spanish, 8= Arabic, dll)	1,8
Religion	Integer (label)	Agama dominan negara (0 = chatolic, 1 = other christian, 2 Muslim, dll)	2, 0
Bars	Integer	Jumlah garis vertikal pada bendera	0, 3
stripes	Integer	Jumlah garis horizontal pada bendera	2, 5
colours	Integer	Jumlah warna berbeda pada bendera	3, 5
Red	Binary (0/1)	Keberadaan warna merah pada bendera	0, 1
green	Binary (0/1)	Keberadaan warna hijau pada bendera	0, 1
blue	Binary (0/1)	Keberadaan warna biru pada bendera	0, 1
Gold	Binary (0/1)	Keberadaan warna emas/kuning pada bendera	0, 1
White	Binary (0/1)	Keberadaan warna putih pada bendera	0, 1
Black	Binary (0/1)	Keberadaan warna hitam pada bendera	0, 1
Orange	Binary (0/1)	Keberadaan warna oranye/coklat pada bendera	0, 1

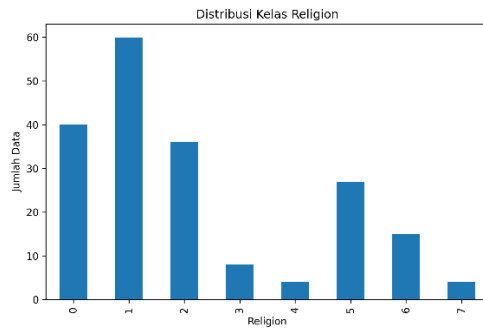
Mainhue	Categorical	Warna dominan utama pada bendera	Red, green
Circles	Integer	Jumlah simbol lingkaran pada bendera	0, 1
Crosses	Integer	Jumlah simbol salib tegak pada bendera	0, 1
Saltires	Integer	Jumlah simbol salib diagonal	0, 1
Quarters	Integer	Jumlah bagian sperempat pada bendera	0, 4
Sunstars	Integer	Jumlah simbol matahari atau bintang	0, 12
crescent	Binary (0/1)	Keberadaan simbol bulan sabit	0, 1
Triangle	Binary (0/1)	Keberadaan bentuk segitiga	0, 1
icon	Binary (0/1)	Keberadaan ikon benda mati (misal : kapal)	0, 1
Animate	Binary (0/1)	Keberadaan ikon makhluk hidup	0, 1
Text	Binary (0/1)	Keberadaan teks atau tulisan pada bendera	0, 1
Topleft	Categorical	Warna di sudut kiri atas bendera	
botright	Categorical	Warna di sudut kanan bawah bendera	

4.3 Kondisi Data

1. Missing Values : Tidak ada (0%)
2. Duplicate Data : Tidak ada (0)
3. Outliers : Ada, deteksi dilakukan menggunakan metode IQR. Terdapat beberapa fitur terindikasi memiliki nilai ekstrem, outlier tersebut valid dan tidak dihapus dikarenakan merepresentasikan karakteristik unik dari data.
4. Imbalanced Data : Ada, kondisi imbalanced data pada variabel religion, dengan rasio kelas 15 : 1. Kelas terbesar memiliki 30,93% data, sedangkan kelas terkecil hanya 2,06%.
5. Noise : Ada, dataset tidak mengandung noise berat atau kesalahan data. Namun, terdapat noise ringan berupa nilai ekstrem (outliers) pada beberapa fitur numerik, khususnya *area*, *population*, dan *sunstars*. Nilai-nilai tersebut bersifat valid secara nyata dan merepresentasikan perbedaan karakteristik antarnegara, sehingga tidak dihapus dari dataset. Noise ini ditangani secara tidak langsung melalui penggunaan algoritma yang robust terhadap outliers, seperti Random Forest, serta melalui proses normalisasi pada model deep learning.
6. Data Quality Issues : -

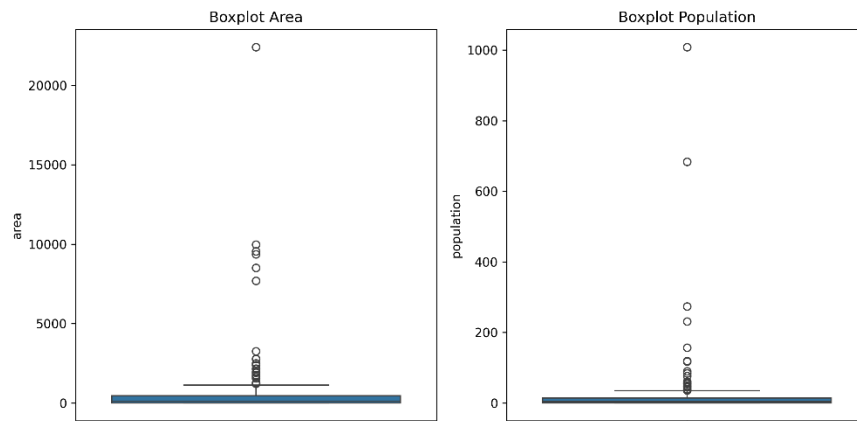
4.4 Exploratory Data Analysis (EDA)

1. Visualisasi 1 :



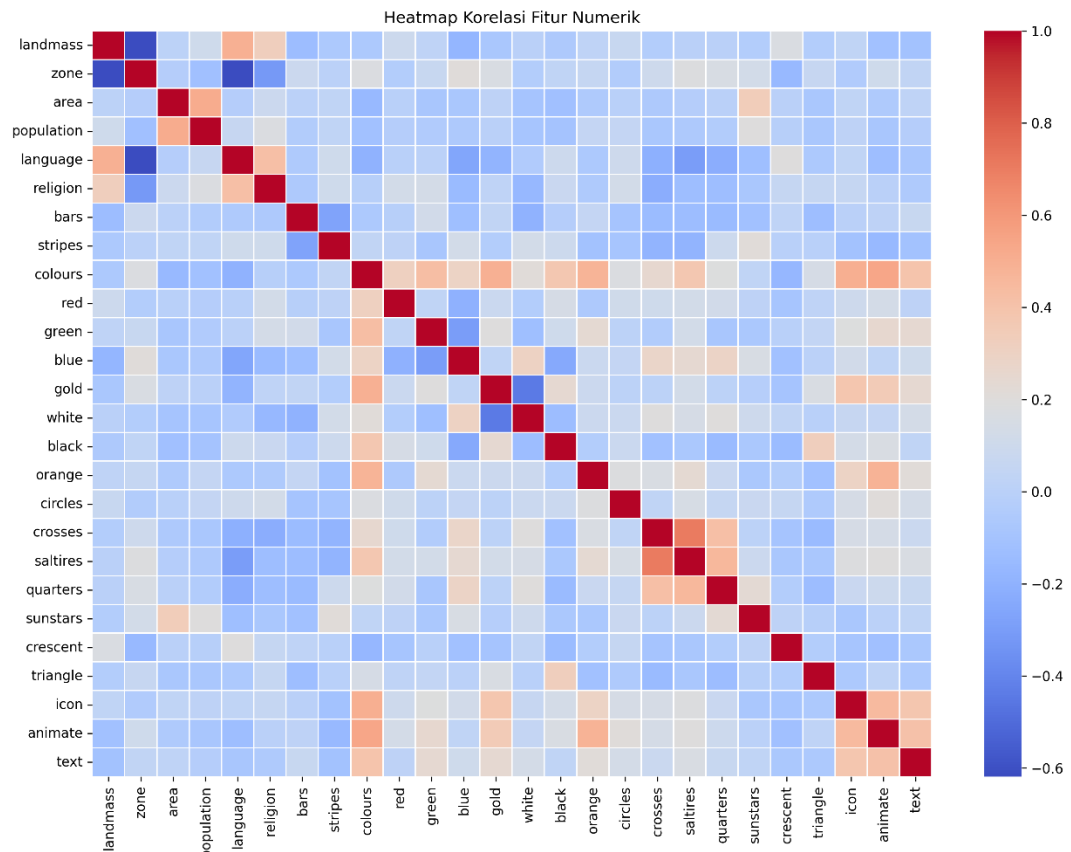
Grafik batang diatas ini menunjukkan bahwa distribusi kelas agama tidak merata, dimana beberapa kelas memiliki jumlah data lebih banyak dibandingkan kelas lainnya. Kondisi ini menunjukkan adanya ketidakseimbangan data (imbalanced data).

2. Visualisasi 2 :



Gambar diatas merupakan boxplot yang menunjukkan nilai ekstrem pada fitur area dan population. Nilai ini memang bersifat valid secara dunia nyata dan mencerminkan perbedaan karakteristik negara, sehingga tidak dihapus dari dataset.

3. Visualisasi 3 :



Heatmap korelasi diatas ini menunjukkan bahwa sebagian besar fitur tidak memiliki korelasi linier yang kuat. Sehingga menunjukkan bahwa hubungan antar fitur bersifat kompleks dan mendukung penggunaan model non-linier seperti random forest dan MPL.

5. Data Preparation

5.1 Data Cleaning

1. Handling Missing Values

Seluruh fitur dalam dataset tidak terdapat missing values (0%). Sehingga tidak dilakukan proses imputasi baik menggunakan mean, median, maupun metode lainnya.

2. Removing Duplicate Data

Hasil pemeriksaan tidak terdapat data duplikat (0 baris duplikat). Maka tidak diperlukan penghapusan data dan seluruh data tetap digunakan dalam analisis.

3. Handling Outliers

Dengan menggunakan metode Interquartile range (IQR) pada fitur numerik. Maka hasil nya adalah beberapa fitur seperti area, population, bars, colours, dan sunstars mengalami outlier. Outlier ini tidak akan dihapus dikarenakan nilai ini merepresentasikan karakteristik nyata antar negara seperti perbedaan luas wilayah dan kompleksitas simbol bendera.

4. Data Type Conversion dan Preprocessing

Kolom name dihapus dikarenakan hanya berfungsi sebagai identitas dan tidak memberikan kontribusi terhadap prediksi. Variabel religion ditetapkan sebagai target, sedangkan fitur lainnya digunakan sebagai input model. Fitur kategorikal

dilakukan proses encoding, dan fitur numerik dilakukan standarisasi, sehingga data memiliki skala yang seragam.

5.2 Feature Engineering

1. Creating New Features

Tidak dilakukan pembuatan fitur baru dikarenakan dataset Flags telah memiliki fitur yang cukup lengkap dan informatif, seperti warna, simbol, dan elemen visual bendera, yang secara langsung berkaitan dengan target klasifikasi religion.

2. Features Extraction

Proses Encoding untuk fitur kategorikal yaitu mainhue, topleft, dan botright. Proses ini mengubah nilai kategorikal menjadi representatif numerik agar dapat diproses oleh algoritma ML dan DL. Dengan encoding untuk informasi warna dominan dan posisi warna pada bendera dapat dimanfaatkan dengan baik oleh model.

3. Feature Selection

Ini dilakukan secara implisit dengan menghapus fitur name, karena fitur ini hanya berfungsi sebagai identitas negara tidak memiliki hubungan langsung dengan target prediksi.

4. Dimensionality Reduction

Tidak menerapkan metode dimensionality reduction seperti PCA. Dikarenakan jumlah fitur setelah preprocessing masih tergolong kecil (<50) dan dapat ditangani dengan baik.

5.3 Data Transformation

Disini untuk tahap preprocessing data tabular dilakukan dengan proses encoding dan scaling. Fitur kategorikal yaitu mainhue, topleft, dan botright diproses menggunakan One-Hot Encoding. Metode ini dipilih karena fitur-fitur tersebut bersifat nominal dan tidak memiliki hubungan urutan, sehingga One-Hot Encoding dapat menampilkan data tanpa memunculkan kesan adanya urutan atau tingkatan antar kategori tertentu. Kemudian fitur numerik dilakukan proses standarisasi menggunakan StandardScaler yang mengubah data agar memiliki mean 0 dan standar deviasi 1. Standarisasi dipilih karena dataset memiliki perbedaan skala yang signifikan antar fitur serta mengandung outliers yang dipertahankan.

5.4 Data Splitting

Dataset dibagi menggunakan metode stratified train-test split untuk menjaga proporsi kelas pada variabel target religion tetap seimbang antara data latih dan data uji. Strategi ini penting dikarenakan dataset memiliki jumlah sampel yang relatif terbatas dan bersifat multiclass. Pembagian data dilakukan dengan proporsi sebagai berikut :

1. Training set : 80% (155 sampel)
2. Test set : 20% (39 sampel)

Dataset tidak dibagi menjadi validation set terpisah, proses validasi dilakukan dengan menggunakan validation split internal atau cross-validation pada data training selama proses pelatihan model. Nilai random state = 42.

5.5 Data Balancing (opsional)

Dikarenakan dataset berukuran kecil dan mengandung banyak fitur kategorikal hasil One-Hot Encoding, sehingga penggunaan SMOTE berpotensi menambah noise dan menurunkan generalisasi model.

5.6 Ringkasan Data Preparation

1. Pemeriksaan struktur data dilakukan dengan pengecekan jumlah baris dan kolom pada dataset Flags, langkah ini bertujuan untuk memastikan jumlah data sesuai dengan dokumentasi dataset serta mengetahui skala data yang akan diproses. Dataset dibaca menggunakan pandas dan diperiksa menggunakan fungsi shape, yang menunjukkan dataset memiliki 194 baris dan 30 kolom.
2. Handling Missing Values dilakukan pemeriksaan nilai kosong (missing values) pada seluruh fitur, missing values dapat menyebabkan error pada proses training model dan menurunkan kualitas hasil prediksi. Pemeriksaan dilakukan menggunakan fungsi `isnull().sum()`. Hasilnya menunjukkan tidak terdapat missing values (0%), sehingga tidak diperlukan proses imputasi.
3. Removing duplicate data, ini dilakukan untuk mengecek apakah ada data yang duplikat di datasetnya, data duplikat dapat menyebabkan bias pada model karena satu observasi direpresentasikan lebih dari satu kali. Pengecekan dilakukan menggunakan fungsi `duplicated().sum()`, hasil menunjukkan tidak terdapat data duplikat sehingga seluruh data dipertahankan.
4. Analisis outliers dilakukan pada fitur numerik menggunakan metode Interquartile range (IQR), outliers dapat mempengaruhi performa model terutama pada algoritma berbasis jarak dan statistik. Outliers diidentifikasi menggunakan batas bawah dan atas IQR, meskipun ditemukan outliers pada beberapa fitur seperti area, population, bars, dan sunstarts, data tersebut tidak dihapus karena mempresentasikan kondisi nyata antar negara dan masih relevan secara dominan.
5. Penghapusan Fitur Tidak Relevan yaitu kolom name dihapus dari dataset, kolom tersebut hanya berfungsi sebagai identitas negara dan tidak memiliki kontribusi prediktif terhadap target. Kolom dihapus menggunakan fungsi `drop()` sebelum proses modeling.
6. Penentuan target dan fitur, variabel religion ditetapkan sebagai target sedangkan fitur lainnya digunakan sebagai input model. Penutupan target diperlukan untuk membedakan data input dan output pada proses supervised learning, dataset dipisahkan menjadi variabel fitur (X) dan target (y) sebelum preprocessing lanjutan.
7. Encoding fitur kategorikal yaitu fitur mainhue, topleft, dan botright. Model machine learning tidak dapat memproses data kategorikal dalam bentuk teks secara langsung, encoding dilakukan menggunakan One-Hot Encoding melalui `OneHotEncoder`, sehingga setiap kategorikal dipresentasikan dalam bentuk vektor biner.
8. Scaling untuk fitur numerik akan distandarisasi menggunakan `StandardScaler`. Scaling diperlukan karena fitur memiliki skala yang berbeda-beda, misalnya antara area dan bars, yang dapat mempengaruhi proses training model. Standarisasi dilakukan dengan mengubah data agar memiliki mean 0 dan standar deviasi 1 menggunakan `StandardScaler`.
9. Penerapan Pipeline Preprocessing untuk gabungan semua proses encoding dan scaling, pipeline meningkatkan konsistensi, menghindari data leakage, dan memudahkan reproduktibilitas eksperimen. Menggunakan `ColumnTransformer`

untuk menerapkan preprocessing pada fitur numerik dan kategorikal secara bersamaan. Hasil akhir menghasilkan dataset dengan 48 fitur yang siap digunakan untuk pemodelan.

10. Kesiapan data untuk modeling, dataset akhir digunakan sebagai input untuk model baseline, advanced, dan deep learning. Data yang telah diproses dengan baik meningkatkan stabilitas dan performa model. Hasil preprocessing disimpan dalam variabel `X_processed` dan digunakan pada tahap pembagian data serta training model.

6. Modeling

6.1 Model 1 Baseline

1. Deskripsi Model

Naive Bayes merupakan algoritma klasifikasi probabiliistik yang bekerja berdasarkan Teorema Bayes dengan asumsi bahwa setiap fitur bersifat independent satu sama lain terhadap kelas target. Model ini menghitung probabilitas suatu data termasuk kedalam kelas tertentu berdasarkan distribusi fitur-fiturnya, kemudian memilih kelas dengan probabilitas tertinggi sebagai hasil prediksi. Naive Bayes dipilih sebagai model baseline karena dataset Flags memiliki banyak fitur biner dan kategorikal seperti warna, simbol, dan elemen visual bendera yang sesuai dengan karakteristik model ini, selain itu Naive Bayes sederhana, cepat, dan efektif pada dataset berukuran kecil dan multivariat, sehingga cocok digunakan sebagai pembanding awal untuk menilai peningkatan performa model yang lebih kompleks.

2. Hyperparameter

Model Naive Bayes yang digunakan adalah gaussian Naive Bayes, dengan parameter sebagai berikut :

- `Var_smoothing`: `1e-9` (default), digunakan untuk menstabilkan perhitungan varians dan menghindari pembagian dengan nol.

Tidak dilakukan tuning hyperparameter secara khusus karena tujuan model ini adalah baseline

3. Implementasi

```
# Inisialisasi model Naive Bayes
model_baseline = GaussianNB()
# Training model
model_baseline.fit(X_train, y_train)
# Prediksi data test
y_pred_baseline = model_baseline.predict(X_test)
# Evaluasi awal (akurasi)
accuracy_baseline = accuracy_score(y_test, y_pred_baseline)
print("Akurasi Naive Bayes (Baseline):", accuracy_baseline)
```

4. Hasil Awal

Model Naive Bayes memperoleh nilai akurasi sebesar 30,77%. Menunjukkan bahwa model mampu menangkap pola dasar pada dataset Flags, namun performanya masih terbatas. Hal ini dapat disebabkan oleh asumsi independensi antar fitur yang dimiliki Naive Bayes, sementara pada dataset ini beberapa fitur seperti warna, simbol, dan pola bendera kemungkinan saling berkorelasi.

6.2 Model 2 ML / Advanced

1. Deskripsi model

Random Forest merupakan algoritma ensemble learning yang membangun banyak model Decision Tree secara acak dan menggabungkan hasil prediksinya melalui mekanisme voting. Setiap pohon dilatih menggunakan subset data subset fitur yang berbeda, sehingga model mampu mengurangi varians dan meningkatkan kemampuan generalisasi dibandingkan satu decision tree tunggal. Random forest dipilih sebagai mode advanced karena mampu menangani data tabular dengan baik, termasuk data dengan fitur numerik dan kategorikal hasil encoding. Model ini relatif robust terhadap outlier dan noise, serta mampu menangkap hubungan non-linier antar fitur yang tidak dapat ditangkap oleh model baseline seperti Naive Bayes.

Keunggulan :

- Mampu menangani hubungan non-linear
- Lebih stabil dan akurat dibanding single decision tree
- Tidak sensitif terhadap outliers ringan
- Cocok untuk dataset tabular berukuran kecil hingga menengah

Kelemahan :

- Interpretasi model lebih sulit dibandingkan model sederhana
- Waktu training lebih lama dibandingkan baseline
- Ukuran model relatif besar

2. Hyperparameter

- `n_estimators` 100 yaitu jumlah pohon keputusan dalam hutan
- `max_depth` yaitu kedalaman maksimum setiap pohon
- `random_state`:42 digunakan untuk memastikan hasil eksperimen dapat direproduksi

Tidak dilakukan hyperparameter tuning secara khusus (seperti Grid search atau random search).

3. Implementasi

```
# Inisialisasi model Random Forest
model_advanced = RandomForestClassifier(
    n_estimators=100,
    max_depth=10,
    random_state=42
)
# Training model
model_advanced.fit(X_train, y_train)
# Prediksi data test
y_pred_advanced = model_advanced.predict(X_test)
# Evaluasi awal (akurasi)
accuracy_advanced = accuracy_score(y_test, y_pred_advanced)
print("Akurasi Random Forest:", accuracy_advanced)
```

4. Hasil model

Menghasilkan nilai akurasi 64.10% yang menunjukkan bahwa model mampu mengklasifikasikan sekitar 64% data uji dengan benar. Random forest cukup efektif dalam menangkap pola kompleks pada dataset Flags yang terdiri dari fitur numerik, biner, dan kategorikal hasil encoding.

6.3 Model 3 Deep Learning

1. Deskripsi Model

Nama Model : MLP

☑Multilayer Perceptron (MLP) - untuk tabular

MLP dipilih karena dataset Flags merupakan data tabular yang telah melalui proses encoding dan scaling. MLP mampu mempelajari hubungan non-linier antar fitur numerik dan kategorikal hasil One-Hot Encoding.

2. Arsitektur model

No	Layer	Jumlah Neuron	Aktivasi	Keterangan
1	Dense	(None, 128)	6,272	ReLU hidden layer pertama
2	Dropout	(None, 128)	0	Dropout 0.3 untuk regularisasi
3	Dense	(None, 64)	8,256	ReLU, hidden layer kedua
4	Dropout	(None, 64)	0	Dropout 0.3
5	Dense	(None, 8)	520	Softmax, output layer (8 kelas agama)

Total parameters : 45,146

Trainable parameters : 15,048

Non-trainable parameters : 0

3. Input dan preprocessing khusus

Input shape : (input_dim) sesuai dengan jumlah fitur setelah one-hot encoding (48fitur)

Preprocessing khusus:

- Scalling numerik menggunakan StandarScaler
- One-hot encoding fitur kategorikal (mainhue, topleft, botright)
- Drop kolom identifier (name)

Tidak ada augmentasi karena dataset tabular

4. Hyperparameter

- Optimizer : Adam
- Learning rate : default 0.001
- Loss Function : categorical_crossentropy
- Metrics : accuracy
- Batch size : 32
- Epoch 50
- Validation split : 0.2 (20% data training digunakan sebagai validation)
- Callback : EarlyStopping (monitor val_loss, patience=10)

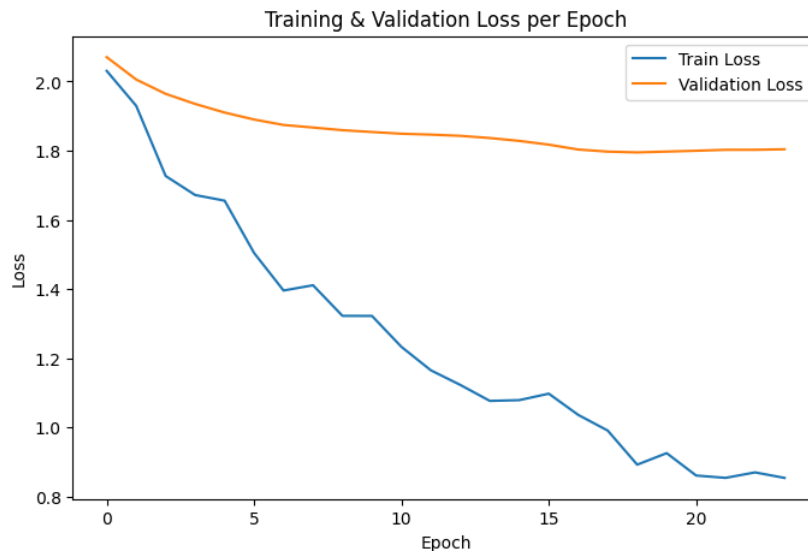
5. Implementasi

6. Training process

Training Time : 8 – 10 menit

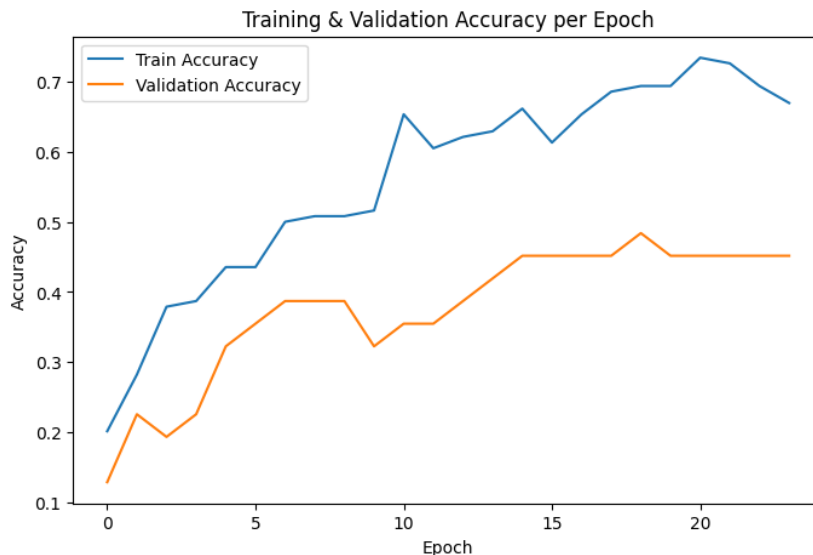
Computational Resource : google colab

Training history visualization :



Train Loss : menurun sangat baik hingga dibawah 1.0

Validation Loss : menurun hingga epoch 6 (1.4), namun setelah itu grafiknya mulai naik kembali. Ini adalah tanda bahaya overfitting.



Train accuracy : meningkat tajam hingga mencapai > 0.7 (70%)

Validation Accuracy : mengikuti kenaikan namun tertahan di angka 0.58 (58%) dan mulai tidak stabil di epoch akhir.

Analisa training :

Model mengalami overfitting, gap antara akurasi traing (70%) dan validasi (58%) cukup besar. Validation loss yang naik kembali di akhir epoch menunjukkan model sudah terlalu menyesuaikan diri dengan data traing sehingga performanya memburuk pada data baru.

Model tidak convergae ke titik yang stabil. Sebaliknya model justru melewati titik optimalnya (yang seharusnya berada di sekitar epoch 6) dan mulai mengalami divergensi (peningkatan error validasi).

Menambahkan epoch akan membuat validation loss semakin naik dan akurasi validasi kemungkinan akan turun lebih jauh. Strategi yang benar adalah berhenti di epoch 6 atau 7 menggunakan early stopping.

7. Model summary

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 128)	6,272
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 64)	8,256
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 8)	520

Total params: 45,146 (176.36 KB)

Trainable params: 15,048 (58.78 KB)

Non-trainable params: 0 (0.00 B)

Optimizer params: 30,098 (117.57 KB)

Arsitektur model menggunakan Sequential yang terdiri dari tiga lapisan utama (Dense) dengan total 15,048 parameter terlatih. Model dimulai dengan lapisan dense_3 yang memiliki 128 unit, diikuti oleh lapisan dropout_2 untuk regulasi, kemudian masuk ke lapisan dense_4 dengan 64 unit dan lapisan dropout_3 untuk mencegah overfitting. Sebagai tahap akhir model ditutup dengan lapisan dense_5 yang memiliki 8 unit, yang mengindikasikan bahwa model ini dirancang untuk tugas klasifikasi dengan 8 kelas target.

7. Evaluation

7.1 Metrik Evaluasi

Berdasarkan dataset yaitu untuk klasifikasi maka metrik evaluasi yang dipakai adalah Accuracy, Precision, Recall, F1-Score (macro), dan confusion Matrix. Accuracy mengukur proporsi prediksi yang benar terhadap total prediksi, memberikan gambaran umum performa model. Precision (macro) mengukur kualitas prediksi positif model untuk setiap kelas tanpa terpengaruh jumlah sampel tiap kelas, sedangkan Recall (macro) menilai kemampuan model menemukan semua instance dari setiap kelas. F1-Score (macro), sebagai harmonik mean dari precision dan recall, memberikan trade-off yang seimbang antara kedua metrik tersebut, cocok untuk multiclass dan dataset yang berpotensi tidak seimbang. Confusion Matrix menampilkan jumlah prediksi benar dan salah per kelas, sehingga memberi insight detail mengenai kelas yang sering salah diklasifikasikan. Kombinasi metrik ini tepat untuk dataset multiclass Flags.

7.2 Hasil Evaluasi Model

1. Model 1 (Baseline)

Metrik :

- Accuracy : 0.3077
- Precision : 0.2321
- Recall : 0.2231
- F1-Score : 0.196

Test Set Prediction :

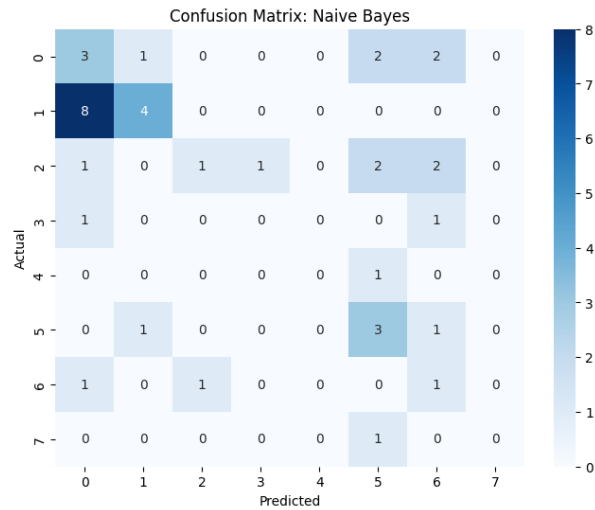
Actual : 3, Prediction : 0

Actual : 1, Prediction : 0

Actual : 0, Prediction : 5

Actual : 1, Prediction :0

Actual : 5, Prediction :1



2. Model 2 (Advanced /ML)

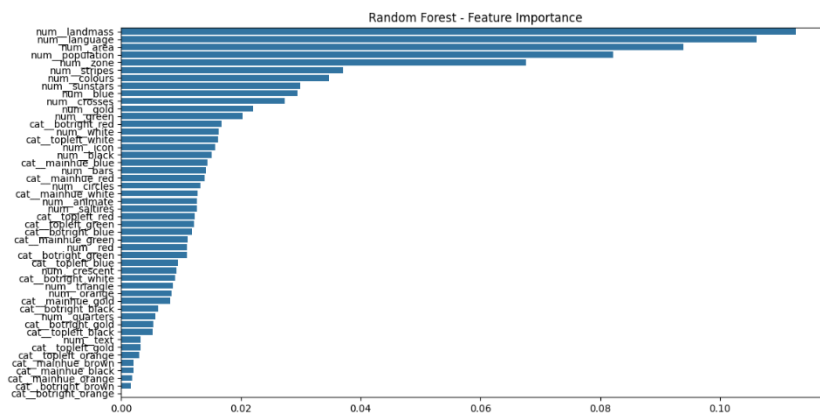
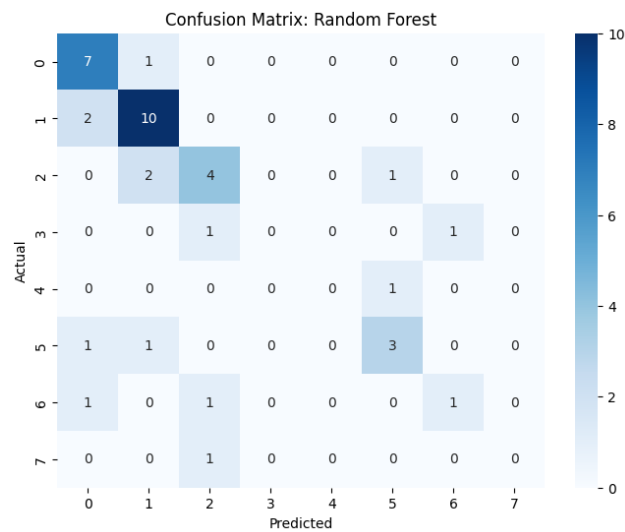
Metrik :

Accuracy : 0.6410

Precision : 0.3778

Recall :0.4016

F1-Score : 0.3847



Feature Importance : Fitur yang paling berpengaruh dalam model ini adalah num__language, diikuti oleh num__landmass, num__area, dan num__population.

Test Set Prediction :

Actual : 3, Prediction : 2

Actual : 1, Prediction : 1

Actual : 0, Prediction : 1

Actual : 1, Prediction : 1

Actual : 5, Prediction : 1

3. Model 3 (MLP – Deep Learning)

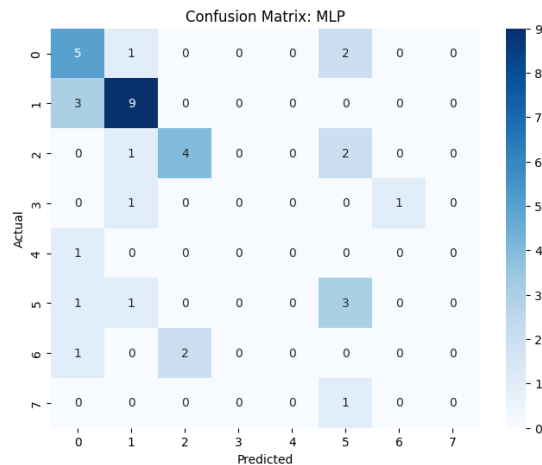
Metrik :

Accuracy : 0.5385

Precision : 0.2736

Recall : 0.3183

F1-Score : 0.2904



Training History :

Loss menunjukkan tren penurunan pada Train Loss hingga dibawah 1.0 namun validation loss mulai meningkat setelah epoch 6, train accuracy mencapai diatas 0.7, sementara validation accuracy berfluktuasi di kisaran 0.5 – 0.6.

Test Set Predictions :

Actual : 3, Prediction : 1

Actual : 1, Prediction : 1

Actual : 0, Prediction : 5

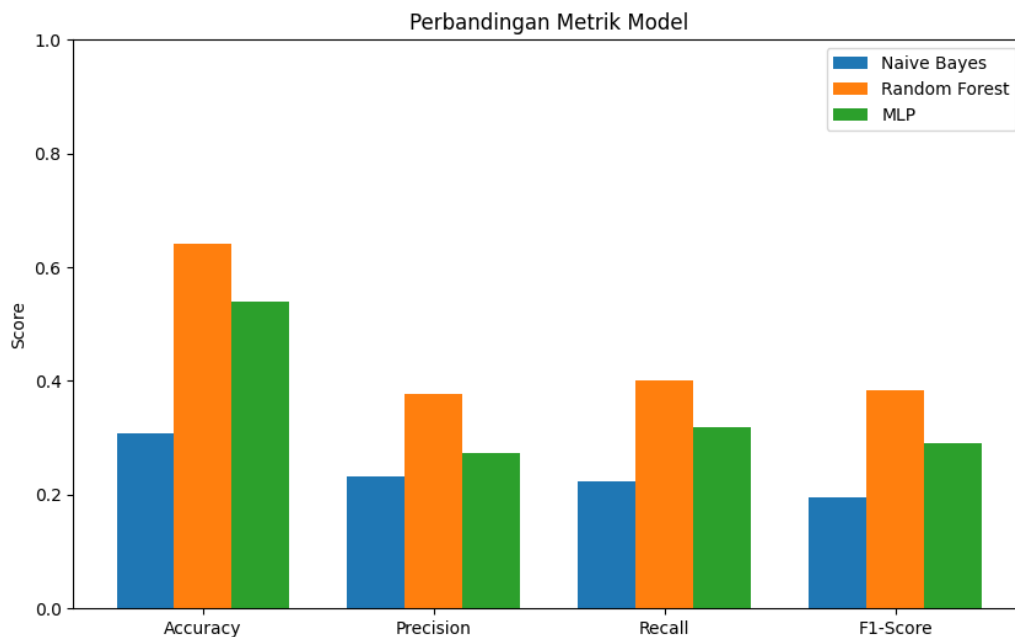
Actual : 1, Prediction : 1

Actual : 5, Prediction : 1

7.3 Perbandingan Ketiga Model

Model	Accuracy	Precision	Recall	F1-Score	Training Time	Inference Time
Baseline (Naive Bayes)	0.308	0.2321	0.2231	0.1960	<1s	<0.01s
Advance (Random Forest)	0.6410	0.3778	0.4016	0.3847	30s	0.05s

Deep Learning (MLP)	0.5385	0.2736	0.3183	0.2904	8-10min	0.1s
---------------------	--------	--------	--------	--------	---------	------



7.4 Analisis Hasil

1. Model terbaik: Random Forest (Advanced/ML)

Meskipun MLP memiliki potensi menangkap hubungan non-linear, pada dataset kecil seperti Flags, MLP menunjukkan overfitting (gap akurasi train vs validation cukup besar) dan performa test set lebih rendah dibanding Random Forest. Random Forest mencapai akurasi 64,10% dengan F1-score 0.3847, lebih baik dari Naive Bayes (30,77%) dan MLP (53,85%). Random Forest mampu menangkap interaksi antar fitur numerik dan kategorikal hasil encoding dengan lebih stabil.

2. Perbandingan dengan baseline:

Naive Bayes : Akurasi rendah karena asumsi independensi antar fitur tidak sesuai dengan dataset Flags (beberapa fitur bendera saling berkorelasi).

Random Forest : Peningkatan performa signifikan untuk Accuracy naik dari 30,77% → 64,10% dan F1-score naik dari 0.196 → 0.3847 Model mampu menangkap hubungan non-linear dan fitur penting (misal language, landmass, area, population) yang tidak bisa diatasi Naive Bayes

MLP : Meskipun potensial untuk hubungan non-linear, dataset kecil + overfitting membuat performa test set lebih rendah dibanding Random Forest.

3. Trade-off:

Naive Bayes: sangat sederhana, cepat, mudah diinterpretasi sehingga performa rendah. Dalam performa dengan waktu training hampir instan (<1s)

Random Forest: kompleksitas sedang, mampu menangkap pola non-linear, robust maka performa cukup baik. Dalam performa dengan waktu training sedang (~30s)

MLP: kompleksitas tinggi, banyak parameter membuat rawan overfitting pada dataset kecil. Dalam performa dengan waktu training lama (8–10 menit di Colab)

4. Error Analysis:

- Kesalahan Umum :
Kesalahan sering terjadi pada kelas dengan distribusi sedikit (minority classes), misal kelas agama dengan jumlah negara rendah. Fitur bendera yang mirip antar negara menyebabkan model sulit membedakan kelas tertentu.
- Kasus sulit diprediksi:
Negara dengan simbol/pola bendera yang tidak unik atau mirip dengan negara lain sering salah klasifikasi.

5. Overfitting/Underfitting:

- Naive Bayes: underfitting karena model terlalu sederhana, tidak mampu menangkap kompleksitas dataset.
- Random Forest: hampir ideal sehingga tidak terlalu overfitting, performa test set relatif mendekati performa train set.
- MLP: overfitting dikarenakan gap besar antara train accuracy ($>70\%$) dan validation accuracy ($\sim 58\%$), validation loss meningkat setelah epoch 6, menunjukkan model mulai menyesuaikan data training secara berlebihan

8. Conclusion

8.1 Kesimpulan Utama

Model terbaik : Random Forest

Alasan :

- Random Forest memberikan keseimbangan terbaik antara performa, stabilitas, dan efisiensi training dibandingkan model lain.
- Performanya unggul dibanding Naive Bayes (baseline) dan MLP (deep learning) untuk dataset Flags, Akurasi test set 64,10% (lebih tinggi daripada Naive Bayes 30,77% dan MLP 53,85%) dan F1-Score: 0.3847, lebih baik dari baseline dan MLP
- Mampu menangkap hubungan non-linier antar fitur numerik dan kategorikal yang sulit ditangani Naive Bayes.
- Tidak mengalami overfitting berlebihan seperti MLP, sehingga performa test set relatif stabil

Goals : Telah tercapai dengan section 3,2

8.2 Key Insights

Insight dari Data :

1. Distribusi fitur unik per kategori

Misal: Banyak bendera memiliki warna merah, putih, dan biru, sehingga fitur red, white, blue sering muncul. Hal ini bisa memengaruhi klasifikasi agama tertentu yang dominan pada negara-negara dengan pola warna tertentu.

2. Beberapa fitur numerik memiliki outlier signifikan

Fitur area dan population memiliki outlier ekstrem (misal Rusia, China), tapi outlier dipertahankan karena merepresentasikan kondisi nyata negara. Ini bisa membuat model lebih sensitif terhadap fitur-fitur skala besar.

3. Fitur kategorikal terbatas tapi penting

Fitur seperti mainhue, topleft, botright memiliki distribusi terbatas namun sangat membantu model membedakan kelas agama tertentu.

Insight dari Model :

1. Model Random Forest lebih efektif menangkap pola non-linear. Performanya lebih baik dibanding Naive Bayes dan MLP karena mampu memanfaatkan interaksi antar fitur numerik dan kategorikal.
2. Deep Learning (MLP) mengalami overfitting meskipun dataset relatif kecil. Train accuracy tinggi (>70%) tetapi validation accuracy stabil di 50–60%. Ini menunjukkan MLP terlalu kompleks untuk dataset tabular kecil seperti Flags.

8.3 Kontribusi proyek

1. Manfaat praktis

Proyek ini dapat digunakan untuk menganalisis korelasi antara desain bendera dan agama dominan suatu negara, menjadi alat edukasi tentang simbolisme bendera, sekaligus contoh penerapan AI pada data tabular multikelas.

2. Pembelajaran yang didapat

Dari proyek ini dipelajari pentingnya preprocessing data, pemilihan model sesuai karakteristik dataset, evaluasi model dengan berbagai metrik, serta strategi mencegah overfitting pada model deep learning.

9. Future Work

Data:

- ☒ Mengumpulkan lebih banyak data
- ☐ Menambah variasi data
- ☒ Feature engineering lebih lanjut

Model:

- ☒ Mencoba arsitektur DL yang lebih kompleks
- ☒ Hyperparameter tuning lebih ekstensif
- ☒ Ensemble methods (combining models)

Transfer learning dengan model yang lebih besar

Deployment:

- ☐ Membuat API (Flask/FastAPI)
- ☐ Membuat web application (Streamlit/Gradio)
- ☐ Containerization dengan Docker
- ☐ Deploy ke cloud (Heroku, GCP, AWS)

Optimization:

- ☐ Model compression (pruning, quantization)
- ☐ Improving inference speed
- ☐ Reducing model size

10. Reproducibility

10.1 Github repository : <https://github.com/arentapm/Proyek-Machine-Learning.git>

10.2 Environment & Dependencies :

Python version : 3.12.12

Library Versions:

Pandas : 2.2.2

numpy	: 2.0.2
matplotlib	: 3.10.0
seaborn	: 0.13.2
scikit-learn	: 1.6.1
tensorflow	: 2.19.0
keras	: 3.10.0

- Jafarigol, E., Keely, W., Hortag, T., Welborn, T., Hekmatpour, P., & Trafalis, T. B. (2023). Religious Affiliation in the Twenty-First Century: A Machine Learning Perspective on the World Value Survey. *Society*, 60(5), 733–749. <https://doi.org/10.1007/s12115-023-00887-0>
- Kalampokas, T., Mentizis, D., Vrochidou, E., & Papakostas, G. A. (2023). Connecting national flags – a deep learning approach. *Multimedia Tools and Applications*, 82(25), 39435–39457. <https://doi.org/10.1007/s11042-023-15056-y>