

Ujian Tengah Semester Praktikum Pembelajaran Mesin

Deskripsi singkat

Praktikan akan melatih sebuah model machine learning untuk mengklasifikasikan apakah suatu akun di platform Kaggle termasuk dalam kategori bot atau bukan. Model ini akan dilatih menggunakan data fitur akun seperti jumlah pengikut, jenis kelamin, jumlah dataset yang diunggah, dan berbagai atribut lainnya. Setelah model dilatih, model yang telah terlatih ini akan digunakan dalam aplikasi web sederhana yang dapat menerima inputan data pengguna dan memberikan prediksi apakah akun tersebut bot atau bukan.

Hal-hal yang akan dilakukan pada UTS ini

1. Melatih model machine learning.
2. Menyimpan model yang telah dilatih dan mengintegrasikannya dengan aplikasi web.
3. Memperbaiki dan melengkapi kode program agar website dapat berjalan dengan lancar.
4. Menguji dan memastikan website dapat berjalan dengan baik pada data input dan memberikan hasil prediksi.

Deskripsi mengenai dataset

Dataset Kaggle Bot Account Detection berisi data mengenai akun palsu atau bot di platform Kaggle. Dataset ini mencakup berbagai fitur yang dapat digunakan untuk menganalisis dan mengidentifikasi pola perilaku akun bot. Informasi lebih lanjut mengenai dataset ini dapat ditemukan di tautan berikut: <https://www.kaggle.com/datasets/shriyashjagtap/kaggle-bot-account-detection>.

Deskripsi mengenai proyek

Struktur dari proyek adalah seperti pada gambar berikut:

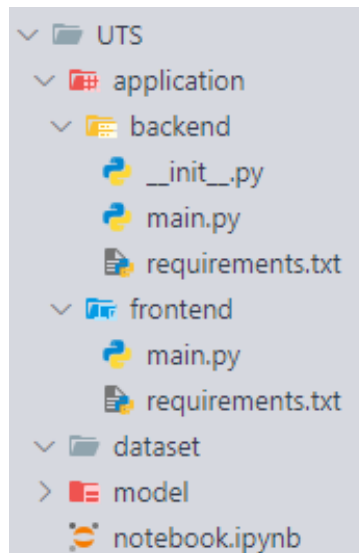


Figure 1: struktur proyek

- **application:** Direktori ini berisi seluruh kode yang diperlukan untuk menjalankan aplikasi. Di dalamnya terdapat sub-direktori **backend** dan **frontend**.

- **backend**: Menyimpan file `main.py` yang berfungsi untuk menjalankan API menggunakan FastAPI. Di sini juga terdapat file `requirements.txt` yang berisi semua dependensi yang diperlukan untuk menjalankan aplikasi backend. File `__init__.py` digunakan untuk menginisialisasi modul.
- **frontend**: Menyimpan file `main.py` yang berfungsi sebagai antarmuka pengguna menggunakan Streamlit. File `requirements.txt` di dalamnya berisi daftar pustaka yang diperlukan untuk aplikasi frontend.
- **dataset**: Berisi dataset yang digunakan untuk melatih model klasifikasi dalam proyek ini.
- **model**: Menyimpan model machine learning yang telah dilatih dan siap digunakan untuk prediksi.
- **notebook**: Notebook ini digunakan untuk melatih model machine learning. Di dalamnya, praktikan akan melakukan pemrosesan data, pelatihan model, dan evaluasi kinerja model. Notebook ini juga mencakup langkah-langkah eksperimen, seperti tuning hyperparameter dan analisis hasil.

Pada langkah pertama, praktikan akan melatih model machine learning yang digunakan untuk mengklasifikasikan apakah suatu akun Kaggle tergolong bot atau tidak. Pelatihan model ini akan dilakukan di dalam file `notebook.ipynb`, yang juga akan digunakan sebagai bahan penilaian. Praktikan diharapkan untuk menerapkan konsep-konsep yang telah dipelajari selama pertemuan sebelumnya, mulai dari melakukan **Exploratory Data Analysis (EDA)** untuk memahami karakteristik dataset, menghapus baris yang duplikat, hingga melakukan imputasi pada sampel yang memiliki nilai yang hilang. Selain itu, praktikan juga diharapkan untuk mengatasi masalah **imbalanced dataset** dan menghitung korelasi antar fitur untuk menentukan fitur yang relevan, serta memplot korelasi tersebut menggunakan heatmap. Selanjutnya, data akan diproses dengan melakukan **preprocessing**, termasuk encoding pada data kategorikal dan scaling pada data numerik. Praktikan juga akan melakukan **hyperparameter tuning** dan **cross-validation** untuk meningkatkan performa model, serta menerapkan teknik **ensemble learning** untuk mengoptimalkan hasil prediksi.

Seluruh proses pelatihan model harus dituliskan secara terurut dan dilengkapi dengan deskripsi yang jelas pada bagian markdown, agar setiap langkah dapat dipahami dengan baik.

Setelah melatih model, langkah selanjutnya adalah mengimplementasikan model ke dalam aplikasi. Proyek ini menggunakan **FastAPI** sebagai backend dan **Streamlit** sebagai frontend. Praktikan akan menggunakan FastAPI untuk menghubungkan model machine learning yang telah dilatih dan menggunakan Streamlit untuk meminta inputan serta menampilkan output. Di bawah ini merupakan gambar dari UI website.

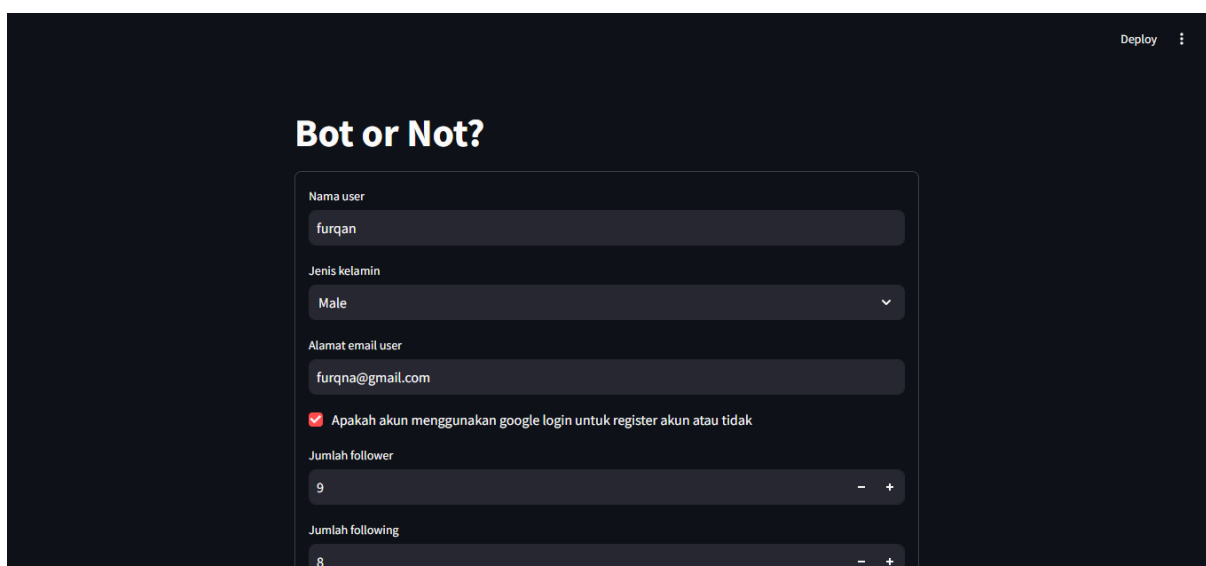


Figure 2: tampilan UI

Sebelum mengintegrasikan model ke dalam aplikasi, praktikan **diharuskan** untuk melengkapi kode program dan function-function yang sengaja dikosongkan dalam file `main.py` di bagian backend dan frontend. Hal ini bertujuan agar website dapat berjalan dengan lancar dan dapat memproses input dari pengguna dengan benar. Praktikan juga akan memastikan bahwa sistem backend dapat menerima data dari frontend, memprosesnya menggunakan model yang telah dilatih, dan mengembalikan hasil prediksi kepada pengguna. Pastikan semua kode sudah dilengkapi sehingga aplikasi berjalan dengan sesuai. Setelah proses ini selesai, praktikan dapat menguji aplikasi untuk memastikan semuanya berfungsi dengan baik dan siap digunakan di production.

Alur preprocessing yang dilakukan oleh praktikan pada saat pelatihan model harus diimplementasikan secara sama persis di pipeline backend. Hal ini bertujuan untuk memastikan bahwa model dapat melakukan klasifikasi pada data baru dengan cara yang konsisten dan sesuai dengan data yang telah digunakan untuk pelatihan sebelumnya. Praktikan harus memastikan bahwa setiap langkah preprocessing, seperti normalisasi, encoding, dan transformasi fitur lainnya, diterapkan dengan cara yang sama pada data yang diterima dari frontend sebelum diproses oleh model. Dengan demikian, hasil prediksi yang diberikan oleh model akan tetap valid dan akurat pada data baru yang diterima oleh aplikasi. Berikut function pipeline model.

```
def preprocess_pipeline():
    numeric_features = [...] # masukkan kolom-kolom numerik

    categorical_boolean_features = [...] # masukkan kolom kategorikal dan boolean

    numeric_transformer = Pipeline(steps=[
        ...
    ])

    categorical_boolean_transformer = Pipeline(steps=[
        ...
    ])

    preprocessor = ColumnTransformer(
        ...
    )

    return preprocessor
```

Figure 3: function pipeline

Selain itu, **fitur-fitur yang digunakan untuk memprediksi kelas juga harus disesuaikan agar sama dengan fitur-fitur yang digunakan pada saat pelatihan model.** Praktikan perlu memastikan bahwa data input yang diterima dari frontend mengandung fitur yang relevan dan sesuai dengan yang digunakan selama proses pelatihan. Setiap perubahan atau ketidaksesuaian pada fitur yang digunakan dapat mengakibatkan hasil prediksi yang tidak akurat atau bahkan gagal. Oleh karena itu, penting bagi praktikan untuk memeriksa dan menyesuaikan setiap fitur yang diterima dari pengguna, baik itu fitur numerik, kategorikal, maupun boolean, sehingga sesuai dengan format dan skala data yang telah diproses sebelumnya. Dengan demikian, proses klasifikasi yang dilakukan oleh model dapat berjalan dengan benar dan menghasilkan prediksi yang valid berdasarkan data yang telah diproses dengan cara yang konsisten.

Untuk menjalankan aplikasi backend dan frontend, langkah-langkah yang perlu dilakukan adalah sebagai berikut:

1. Backend:

- Masuk ke direktori `backend`.
- Pastikan semua dependensi yang diperlukan telah terinstal dengan menjalankan perintah `pip install -r requirements.txt`.
- Jalankan backend dengan perintah `uvicorn main:app --reload` untuk mengaktifkan server FastAPI.

- Setelah server berjalan, backend akan menerima request dari frontend dan memprosesnya.

2. Frontend:

- Masuk ke direktori `frontend`.
- Pastikan semua dependensi untuk Streamlit telah terinstal dengan menjalankan perintah `pip install -r requirements.txt`.
- Jalankan frontend dengan perintah `streamlit run main.py` untuk memulai aplikasi Streamlit.
- Setelah itu, aplikasi frontend akan berjalan dan dapat meminta input dari pengguna serta menampilkan output prediksi.

Pastikan bahwa backend dan frontend berjalan secara bersamaan dan dapat saling berkomunikasi dengan lancar. Jika ada kesalahan atau ketidaksesuaian dalam input atau output, lakukan debugging pada bagian yang relevan di backend atau frontend. Setelah kedua aplikasi berjalan dengan baik, pastikan untuk menguji alur kerja secara menyeluruh untuk memastikan semuanya berfungsi dengan benar.

Pada saat pengumpulan tugas UTS ini, praktikan diharuskan untuk mengumpulkan keseluruhan proyek, namun tanpa menyertakan folder `env`. Semua file proyek harus di-zip terlebih dahulu sebelum dikumpulkan. Praktikan harus memastikan bahwa seluruh kode yang digunakan untuk melatih model machine learning dan dokumentasi penjelasannya telah tertulis dengan jelas di dalam notebook yang telah disediakan. Selain itu, pastikan semua kode pada bagian front-end dan back-end yang belum lengkap telah diselesaikan dan berfungsi dengan baik. Sebelum mengumpulkan, pastikan website berjalan dengan sempurna di masing-masing laptop praktikan, dan jika praktikan lupa menghapus folder `env` sebelum mengumpulkan, maka pengurangan nilai akan diberlakukan.

Metrik Penilaian UTS

- **Praktikan menerapkan konsep EDA dan preprocessing yang lengkap, termasuk mencari korelasi antar fitur dan lainnya** (10 poin). Pada bagian ini, praktikan diharapkan melakukan eksplorasi data yang mendalam, melakukan pengolahan data seperti imputasi nilai yang hilang, mengatasi data yang tidak seimbang, serta menganalisis korelasi antar fitur untuk menentukan fitur yang relevan.
- **Praktikan menerapkan hyperparameter tuning dan cross-validation** (20 poin). Dalam hal ini, praktikan diharapkan melakukan pencarian hyperparameter yang optimal untuk model yang digunakan, serta melakukan cross validation untuk mengukur performa model secara menyeluruh dan menghindari overfitting.
- **Praktikan menerapkan ensemble learning** (20 poin). Praktikan diharapkan menggunakan teknik ensemble seperti Random Forest, Bagging, Boosting, atau Stacking untuk meningkatkan kinerja model dan memberikan hasil prediksi yang lebih akurat.
- **Praktikan telah melengkapi seluruh kode** (20 poin). Penilaian ini akan menilai kelengkapan dan keakuratan kode yang telah praktikan implementasikan pada bagian backend dan frontend, serta memastikan bahwa seluruh fungsionalitas yang diperlukan telah diselesaikan.
- **Program telah berjalan dengan semestinya** (30 poin). Pada bagian ini, evaluasi akan dilakukan terhadap keberhasilan implementasi aplikasi secara keseluruhan, termasuk integrasi antara backend dan frontend, serta memastikan bahwa aplikasi dapat menerima input, memproses data, dan menampilkan output dengan benar.