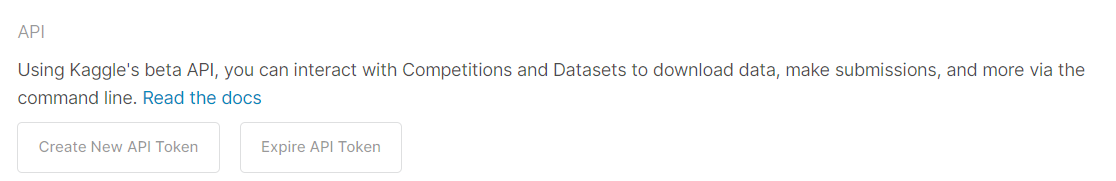
[**Kaggle**](https://www.kaggle.com/general/51898) **:**

**Generate kaggle.json → Login ke kaggle → klik avatar profile yang ada di sebelah kanan atas → masuk ke bagian account → scroll down ke menu API dan klik “Create New API Token”**

****

**Code untuk Google Colab → Download Datasets**

# upload kaggle.json

from google.colab import files

files.upload()

# make directory and change permission

!mkdir -p ~/.kaggle

!cp kaggle.json ~/.kaggle/

!chmod 600 ~/.kaggle/kaggle.json

!ls ~/.kaggle

# download dataset, choose 'copy api command' from kaggle dataset

!kaggle datasets download -d hgultekin/bbcnewsarchive

1. **Submission 1 : NLP**   
   Kriteria:

* Dataset yang akan dipakai bebas, namun minimal memiliki 1000 sampel.
* Harus menggunakan LSTM dalam arsitektur model.
* Harus menggunakan model sequential.
* Validation set sebesar 20% dari total dataset.
* Harus menggunakan Embedding.
* Harus menggunakan fungsi tokenizer.
* Akurasi dari model minimal 75%.
* **Bintang 4**: 2000 sampel data, akurasi pada training set dan validation set di atas 85%.
* **Bintang 5**: 2000 sampel data, minimal 3 kelas, akurasi pada training set dan validation set di atas 90%.

Referensi Dataset:

* [BBC News](https://raw.githubusercontent.com/susanli2016/PyCon-Canada-2019-NLP-Tutorial/master/bbc-text.csv)
* [BBC News Kaggle](https://www.kaggle.com/hgultekin/bbcnewsarchive)
* [Coronavirus Tweets](https://www.kaggle.com/datatattle/covid-19-nlp-text-classification)
* [Google QUEST Q&A Labeling](https://www.kaggle.com/c/google-quest-challenge/data)
* [movie genre data](https://www.kaggle.com/lokkagle/movie-genre-data)
* [Emotions dataset for NLP](https://www.kaggle.com/praveengovi/emotions-dataset-for-nlp)
* [Indonesian Abusive and Hate Speech Twitter Text](https://www.kaggle.com/ilhamfp31/indonesian-abusive-and-hate-speech-twitter-text?select=data.csv)

Tips:

* Untuk dataset dari kaggle, anda bisa download dataset pada colab menggunakan [Kaggle API](https://www.kaggle.com/general/51898)
* Untuk tokenizer.fit\_on\_text sebaiknya hanya digunakan untuk training data, sehingga di test data terdapat kosakata baru yang tidak ada di tokenizer, hal ini dapat menghindari peluang overfitting
* Penggunaan library stopwords dari NLTK atau sklearn, cocok untuk yang datanya mengandung kalimat atau teks yang panjang, kosakata umum dapat dihapus agar meningkatkan kualitas model dan menghindari overfitting
* Bisa melakukan remove full url, remove the punctuation, dan beberapa karakter yang tidak penting dengan regex.
* Untuk informasi terkait Text Processing bisa di check di [url ini](https://www.kdnuggets.com/2019/04/text-preprocessing-nlp-machine-learning.html)
* Untuk callback, sebaiknya yang di logs.get adalah val\_accuracy, banyak ternyata yang tidak menghiraukan val\_accuracy sehingga submissionnya kurang memuaskan
* Pastikan data tidak ada data atau bagian data yang NaN / null / N/A
* Kode yang tidak digunakan sebaiknya dihapus, utamakan penggunaan tag comment untuk menjelaskan bagian kode saja.
* Anda bisa terapkan penggunaan layer [Bidirectional LSTM](https://www.tensorflow.org/api_docs/python/tf/keras/layers/Bidirectional)
* Pelajari PengunaanCallbacks, [EarlyStopping](https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/), ModelCheckpoint-(optional)
* Menggunakan Dataset BBC cukup mainstream di kalangan tutorial ML, sehingga sudah dipastikan datasetnya bagus, contohnya [disini](https://towardsdatascience.com/multi-class-text-classification-with-lstm-using-tensorflow-2-0-d88627c10a35)

1. **Submission 2 : Time Series**  
   Kriteria:

* Dataset yang akan dipakai bebas, namun minimal memiliki 1000 sampel.
* Harus menggunakan LSTM dalam arsitektur model.
* Validation set sebesar 20% dari total dataset.
* Model harus menggunakan model sequential.
* Harus menggunakan Learning Rate pada Optimizer.
* MAE < 10% skala data.
* **Bintang 4:** dataset memiliki minimal 2000 sampel data, MAE dari model < 10% skala data.
* **Bintang 5**: dataset memiliki minimal 10000 sampel data, MAE dari model < 10% skala data.

Rekomendasi dataset:

* [Delhi Weather Data](https://www.kaggle.com/mahirkukreja/delhi-weather-data)
* [Bike Sharing Dataset](https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset)
* [Gold Price](https://www.kaggle.com/arashnic/learn-time-series-forecasting-from-gold-price)
* [Weather archive Jena](https://www.kaggle.com/pankrzysiu/weather-archive-jena)

Tips:

* Untuk dataset dari kaggle, anda bisa download dataset pada colab menggunakan [Kaggle API](https://www.kaggle.com/general/51898)
* Menggunakan teknik normalisasi data (MinMaxScaler, StandardScaler)
* Perhitungan kriteria\_mae = (data.max() - data.min()) \* 0.1
* Hyperparameter tuning pada model dan data
* Timestep hanya sebagai acuan saja, namun pastikan data sudah terurut berdasarkan timestep
* Cara menggunakan validasi data di time series:

train\_set = windowed\_dataset(train\_data, .......)

valid\_set = windowed\_dataset(valid\_data, .......)

……

model.fit(train\_set, epoch=100, validation\_data=valid\_set)

* Dibeberapa kasus menghapus Outlier akan dapat membantu pembelajaran mesin namun tidak memungkinkan akan sesuai dengan ekspektasi, maka lakukan pengujian dengan menghapus Outlier.
* Kode yang tidak digunakan sebaiknya dihapus, utamakan penggunaan tag comment untuk menjelaskan bagian kode saja.
* Anda bisa terapkan penggunaan layer [Bidirectional LSTM](https://www.tensorflow.org/api_docs/python/tf/keras/layers/Bidirectional)
* Pelajari Pengunaan Callbacks, [EarlyStopping](https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/), ModelCheckpoint-(optional)
* Cari dataset dengan range data (bukan jumlah data) yang sangat besar (misal data.min() nya 1, dan data.max() nya 3000), sehingga kemungkinan dapet mae kecil pada epoch awal-awal tinggi

1. **Submission 3 : Image Classification**  
   Kriteria:

* Dataset yang akan dipakai bebas, namun minimal memiliki **1000 buah gambar**.
* Dataset dibagi menjadi **80% train set** dan **20% test set**.
* Model harus menggunakan **model sequential**.
* Model harus menggunakan **Conv2D Maxpooling Layer**.
* Akurasi pada training dan validation set **minimal sebesar 80%.**
* Menggunakan **Callback**.
* Membuat **plot** terhadap akurasi dan loss model.
* Menulis kode untuk menyimpan model ke dalam format **TF-Lite**.
* **Bintang 4** : Dataset memiliki **minimal 2000** sampel gambar dan **minimal 3 kelas**. Serta akurasi pada training dan validation set **minimal 85%.**
* **Bintang 5** : Dataset memiliki **minimal 10000 gambar**, resolusi gambar pada dataset tidak seragam. Serta akurasi pada training set dan validation set **minimal 92%**.

Rekomendasi dataset:

* Rock Paper Scissor (Yang ada di submission pemula)
* [Animals 10](https://www.kaggle.com/madisona/translated-animals10/version/1) (Ambil beberapa class saja)
* [FruitsVegetables](https://drive.google.com/file/d/1ruaStccmRUdgpxlI5lD2LDWH9nxoc9VY/view?usp=sharing)
* [Disease](https://drive.google.com/file/d/1ONPBCD-wRaZp8yY4EplIZImzrm3GYjnC/view?usp=sharing)
* [ASL Alphabet](https://www.kaggle.com/grassknoted/asl-alphabet?select=asl_alphabet_train)

Tips:

* Untuk melakukan EDA (Exploratory Data Anlysis)
* Bisa melakukan Preprocessing Image
* Menggunakan Teknik Transfer Learning, lumayan lama waktu training tapi akurasi lumayan cukup tinggi

Contohnya :

* + ResNet152V2
  + DenseNet121
  + InceptionV3
  + MobileNetV2
  + VGG19
* Untuk dataset dari kaggle, anda bisa download dataset pada colab menggunakan [Kaggle API](https://www.kaggle.com/general/51898)
* Jangan lupa implementasi Callback, plot (accuracy and loss), dan TFLite converter
* Pelajari penggunaan [Augmentasi](https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/) gambar
* Pada saat melakukan prediksi pada image yang di input/”choose file” sebaiknya di normalisasi juga. Pada variable train\_datagen dan test\_datagen, sebelum gambar di feed ke model dinormalisasikan terlebih dahulu gambarnya. Sehingga sebaiknya dibagian prediksi juga perlu melakukan hal yang sama. Setelah mengubah gambar menjadi array. Array tersebut perlu di normalisasi dengan kode berikut : **x /= 255**
* Karena probabilitas salah satu kelas/label dari model dengan layer ber-activation softmax belum tentu 1, maka langkah terbaik adalah dengan mengecek mana probabilitas yang paling tinggi. Bisa dibuat menggunakan logika if else, atau menggunakan modul numpy.argmax

**prediksi = np.argmax(classes)**

pada variable prediksi akan terdapat angka/integer tertinggi yang merepresentasikan kelas/labelnya. Bisa di cek pada variable **train\_generator.class\_indices**

* Pelajari penggunaan [Dropout](https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/)
* Pelajari Pengunaan Callbacks, [EarlyStopping](https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/), ModelCheckpoint-(optional)