https://www.kaggle.com/competitions/oct312022moviegenres/rules

Это ссылка на соревнование, необходимо ознакомиться с кодом в google colab и правилами, модифицировать код в google colab для достижения наибольшего результата в leadboard (желательно получить больше 0,97). Начинать участие в соревнование не нужно (и загружать что-то), все черновые варианты необходимо отправлять мне и проверять через мой аккаунт.

Colab, который производит результат 0.904, в архиве:

In this competition, we predict the movie genre categories for observations in The Movie Database ([TMDB](https://www.themoviedb.org/?language=en-US)). This is a 20-class [multi-label prediction](https://scikit-learn.org/stable/modules/multiclass.html), where most movies have 3-4 labels, but a few can have 8 genres. Much of semantic information is encoded in text, so natural language processing (NLP) is useful for feature extraction. See baseline model.

See Rules tab for guidelines, grading, submission details, and the starter Colab notebook. You are encouraged to learn from other competitions in Kaggle or elsewhere, but keep your modeling constrained by rules of this competition.

## 𝓐𝓵𝓵𝓸𝔀𝓮𝓭 𝓜𝓸𝓭𝓮𝓵𝓼 𝓪𝓷𝓭 𝓔𝓷𝓰𝓲𝓷𝓮𝓮𝓻𝓲𝓷𝓰:

1. **Ok to use these SKL models:**
   1. Current week's topic: [decision trees](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.tree) and [ensembles](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.ensemble)
   2. Relevant: [multi-class/output](https://scikit-learn.org/stable/modules/multiclass.html), [multi-output](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.multioutput), [multi-class](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.multiclass), except SVC/SVM, [MLP](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.neural_network), and others that we have not covered yet
   3. Other (separately or in ensemble): [linear models](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.linear_model), [KNN](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.neighbors), [Naive Bayes](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.naive_bayes), [LDA/QDA](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.discriminant_analysis)
2. You can also use clustering algorithms and dimensions reduction algorithms (stats::pca() in R, PCA() in SKL, etc.)
3. Ok to engineer features, tune hyperparameters/optimizers, sub/oversample, regularize, [auto select features](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.feature_selection), [extract features](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.feature_extraction), reduce dimensions, use CPU/GPU/TPU.
4. Ok to use [these pre-trained SBERT embeddings](https://www.sbert.net/docs/pretrained_models.html).

The evaluation metric for this competition is [Mean F1-Score](https://en.wikipedia.org/wiki/F-score) ([SKL's description](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html#sklearn.metrics.f1_score)). The F1 score, commonly used in information retrieval, measures accuracy using the statistics precision ( \text{p} ) and recall ( \text{r} ).

Precision is the ratio of true positives (\text{tp}) to all predicted positives (\text{tp} + \text{fp}). Recall is the ratio of true positives (\text{tp}) to all actual positives (\text{tp} + \text{fn}). The F1 score is given by:

[ \text{F1} = 2\frac{\text{p} \cdot \text{r}}{\text{p}+\text{r}}\ \ \mathrm{where}\ \ \text{p} = \frac{\text{tp}}{\text{tp}+\text{fp}},\ \ \text{r} = \frac{\text{tp}}{\text{tp}+\text{fn}} ]

The F1 metric weights recall and precision equally, and a good retrieval algorithm will maximize both precision and recall simultaneously. Thus, moderately good performance on both will be favored over extremely good performance on one and poor performance on the other.