Team_Fake

News Authenticity Verification

Ml and DL strategies to fight fakes...

Highlights

- Problem statement
- Proposed strategy
- Achieved results

Note: These approaches are based on supervised learning approaches.

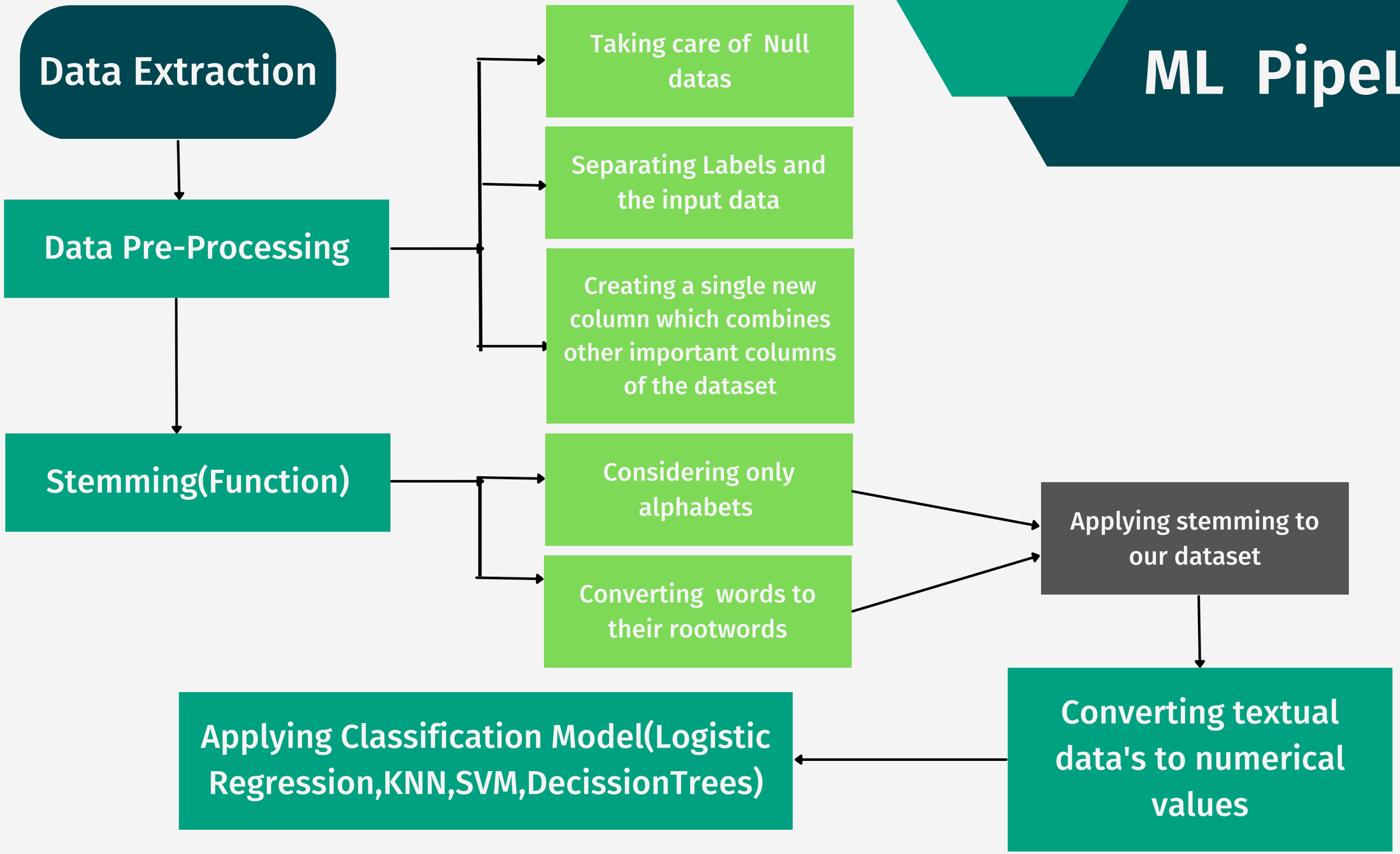
Problem statement:

To identify and differentiate between fake and original news. Develop a tool that aims at retrieving factual news on a certain topic. The tool shall take a news article and verify the facts, events and other relevant information

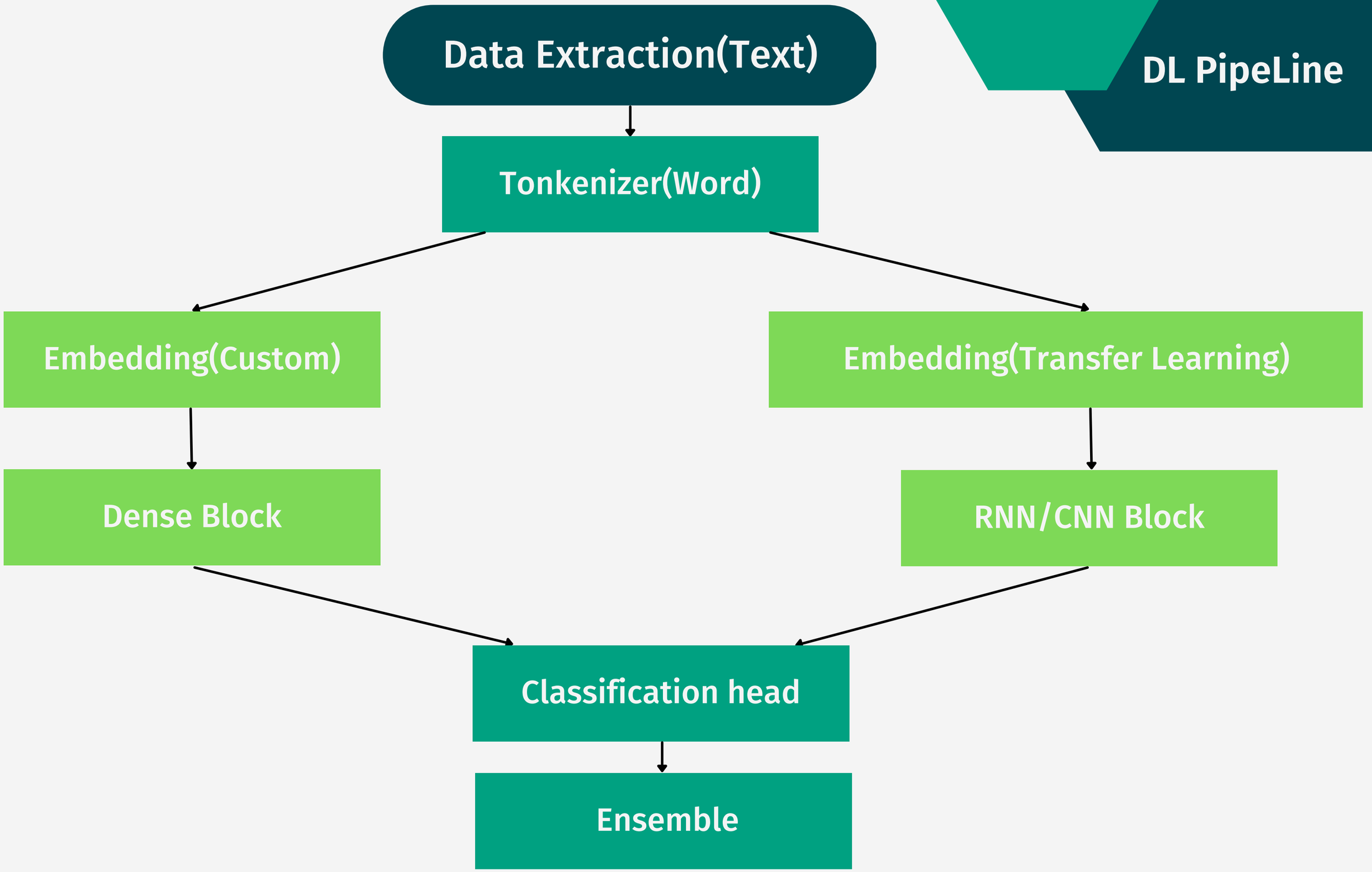
Dataset:

https://drive.google.com/file/d/1Ry11w6IizgwTC4OEeYbhYLH6Bq_-jnZT/view?usp=sharing

ML PipeLine

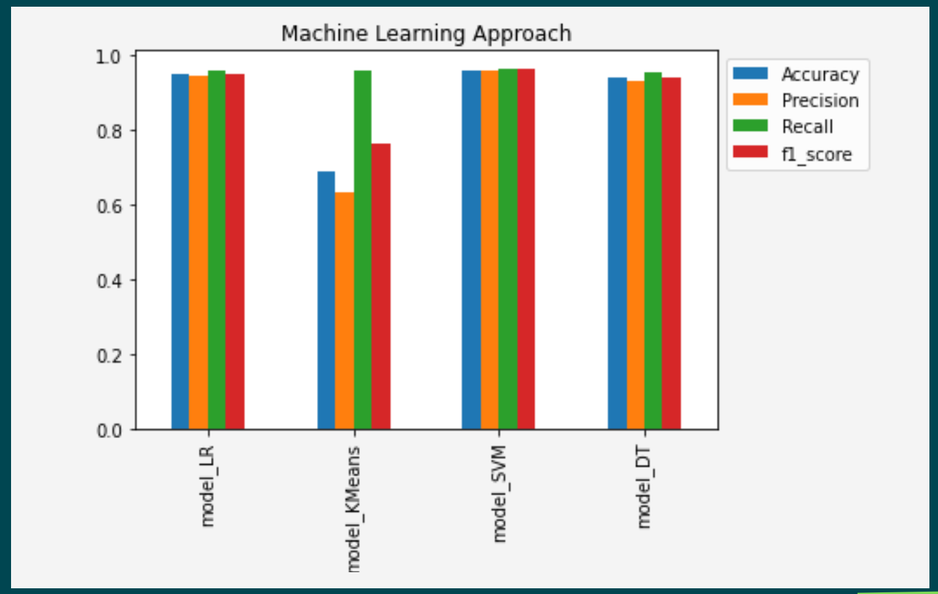

Data Extraction
Data Pre-Processing
Taking care of Null datas
Separating Labels and the input data
Creating a single new column which combines other important columns of the dataset
Stemming(Function)
Considering only alphabets
Converting words to their rootwords
Applying stemming to our dataset
Converting textual data's to numerical values
Applying Classification Model(Logistic Regression,KNN,SVM,DecissionTrees)

DL PipeLine


Data Extraction(Text)
Tonkenizer(Word)
Embedding(Custom)
Embedding(Transfer Learning)
Dense Block
RNN/CNN Block
Classification head
Ensemble

Models ML_strategy

	Accuracy	Precision	Recall	f1_score
model_LR	0.948638	0.943088	0.957957	0.950465
model_Kmeans	0.689471	0.63026	0.958766	0.760556
model_SVM	0.959174	0.956929	0.964021	0.960462
model_DT	0.938241	0.928365	0.953511	0.940771



- Logistic Regression.
- k-nearest neighbors.
- Support Vector Machine.
- Decision Trees.

on →

- Stemmed(Reducing word to its root word)

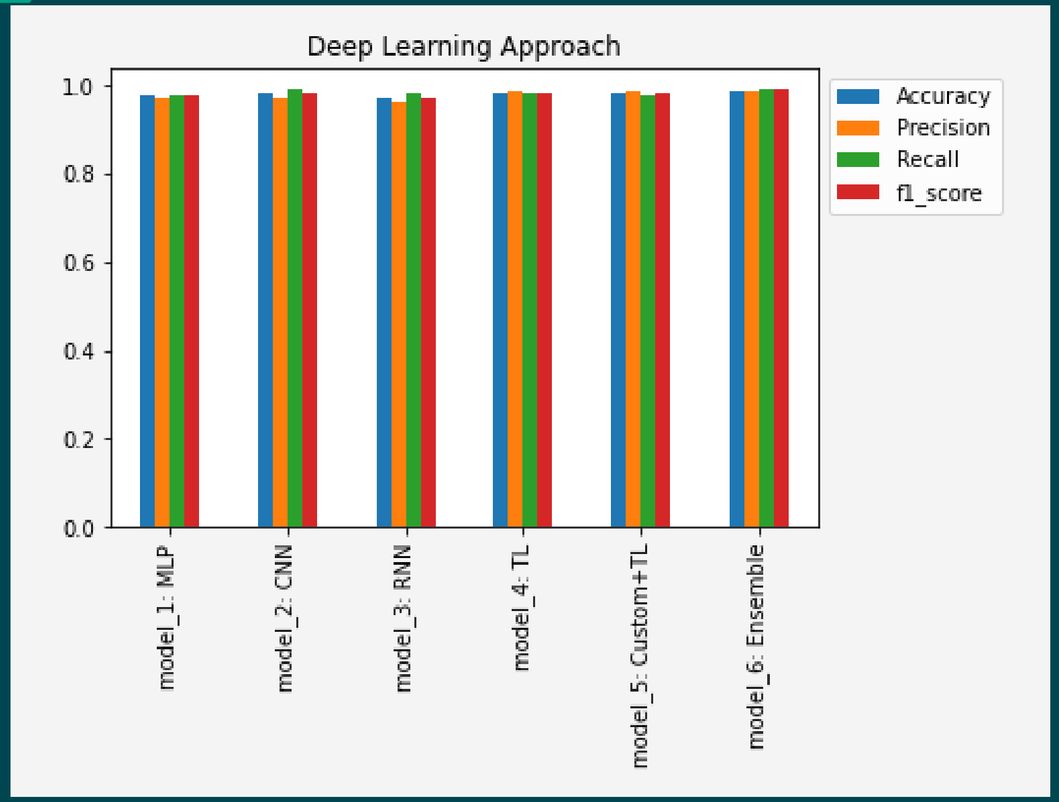
+

- TfidfVectorizer(Converting textual to numerical values)

Data's

Models DL_strategy

	Accuracy	Precision	Recall	f1_score
model_1_MLP	0.974049	0.981311	0.967982	0.974601
model_2_CNN	0.983697	0.983423	0.984907	0.984165
model_3_RNN	0.979651	0.977285	0.983291	0.980278
model_4_TL	0.984474	0.989658	0.980056	0.984834
model_5_Custom+TL	0.984585	0.988597	0.981351	0.984961
model_6_Ensemble	0.988688	0.988477	0.989543	0.989011



- **Multi-layer perceptron** with custom made embedding.
- **Convolutional deep network** with custom made embedding.
- **Recurrent Neural Networks.**

- Multi-layer perceptron with transfer embedding.
- Concatenation of custom and transferred embeddings.
- **Ensemble** using model 1&2

Colab_CodeLink:

Debunkathon Team Fake

Thank You