NLP Project

Parsa Kangavari

July 11, 2023

1 Phase 1: Crawling And Cleaning Data

Github Repository Link: Click here.

1.1 Step 1: Data Crawling

At first we should find websites that include match reports and player's ratings of a match in a single pages. The best website for this is SkySport. Then we should find and store urls of that pages. This urls has been crawled from a webpage that contains all matchs. For crawling we use Scrapy framework. At first we should crawl all URLs from SkySport. So we should crawl urls from a page contains all urls. In this page there will be all urls of all matches. We should choose seasions that we want. In this project we have chosen Primier league, Champions league and Fifa World Cup.

All matches page link: Click here.

Then we have been crawled urls of pages contains reports from this page and save them as a csv file. We have developed a spider that do this. you can see it in MatchURLSpider.py.

Results:

	Unnamed: 0	url
0	0	https://www.skysports.com/football/burnley-vs-bournemouth/373467
1	1	https://www.skysports.com/football/crystal-palace-vs-west-bromwich-albion/373468
2	2	https://www.skysports.com/football/huddersfield-town-vs-arsenal/373469
3	3	https://www.skysports.com/football/liverpool-vs-brighton-and-hove-albion/373470
4	4	https://www.skysports.com/football/manchester-united-vs-watford/373471
5	5	https://www.skysports.com/football/newcastle-united-vs-chelsea/373472
6	6	https://www.skysports.com/football/southampton-vs-manchester-city/373473
7	7	https://www.skysports.com/football/swansea-city-vs-stoke-city/373474
8	8	https://www.skysports.com/football/tottenham-hotspur-vs-leicester-city/373475
9	9	https://www.skysports.com/football/west-ham-united-vs-everton/373476

After that we have crawld reports and player ratings from each urls in dataset above. This informations have been crawld by a spider that you can see in Soc-

cerSpider.py. In this spider we read urls CSV file and then crawl all informations from each url.

Results:

	Unnamed: 0	report	ratings
0	0		
1	1		
2	2		
3	3		
4	4		
5	5		
6	6		
7	7		
8	8		
9	9		

1.2 Setp 2: Cleaning Data

For this section we have to clean datas that we crawled before. We should tokenize reports of matchs by sentences and words. For first one we split report by dots and for second one we split them by spaces. Then we should clean player ratings. player ratings are in range 0 to 10. We should set ratings true if they are bigger than 6 and set false otherwise.

tokenized by sentences:

	sent132	sent133	sent134	sent135	sent136	player0	rating0
0	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	McCarthy	True
1	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	Pope	True
2	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	Hennessey	False
3	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	Karius	True
4	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	Romero	False
5	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	Lossl	True
6	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	Fabianski	True
7	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	Dubravka	True
8	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	Lloris	True
9	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	Adrian	True
10	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	Fabianski	False

You can also see it on : Click here

tokenized by words:

	word2249	word2250	word2251	word2252	word2253	player0	rating0
0	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	McCarthy	True
1	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	Pope	True
2	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	Hennessey	False
3	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	Karius	True
4	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	Romero	False
5	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	Lossl	True
6	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	Fabianski	True
7	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	Dubravka	True
8	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	Lloris	True
9	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	Adrian	True
10	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	¡PAD¿	Fabianski	False

1.3 Step 3: Metrics

For this section we should get most regular words, unique words of each reports and ... . At first we should create new datasets that contains of sentences that belongs to positive and negative players. We have done it in PandNSeperation.py. In this script we seperate sentences belong to negative players and positive players and then save them in 2 seperated CSV file; negative and positives.csv. In this datasets, there are seperated sentences.

Results for negative:

	Unnamed: 0	sents
7	7	Solanke clincally finished a scintillating counter involving Salah and Firmino to finally br

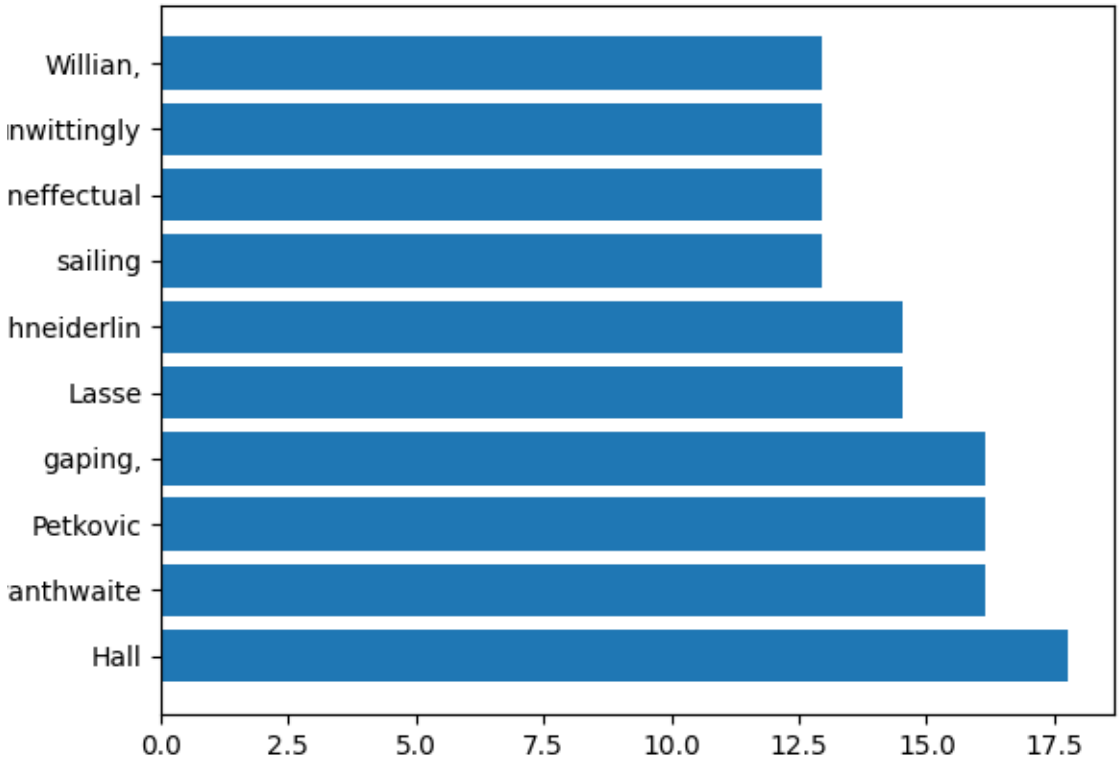
Results for positives:

	Unnamed: 0	sents
7	7	The contest did not really come alive until the 20th minute when England hopeful Nick F

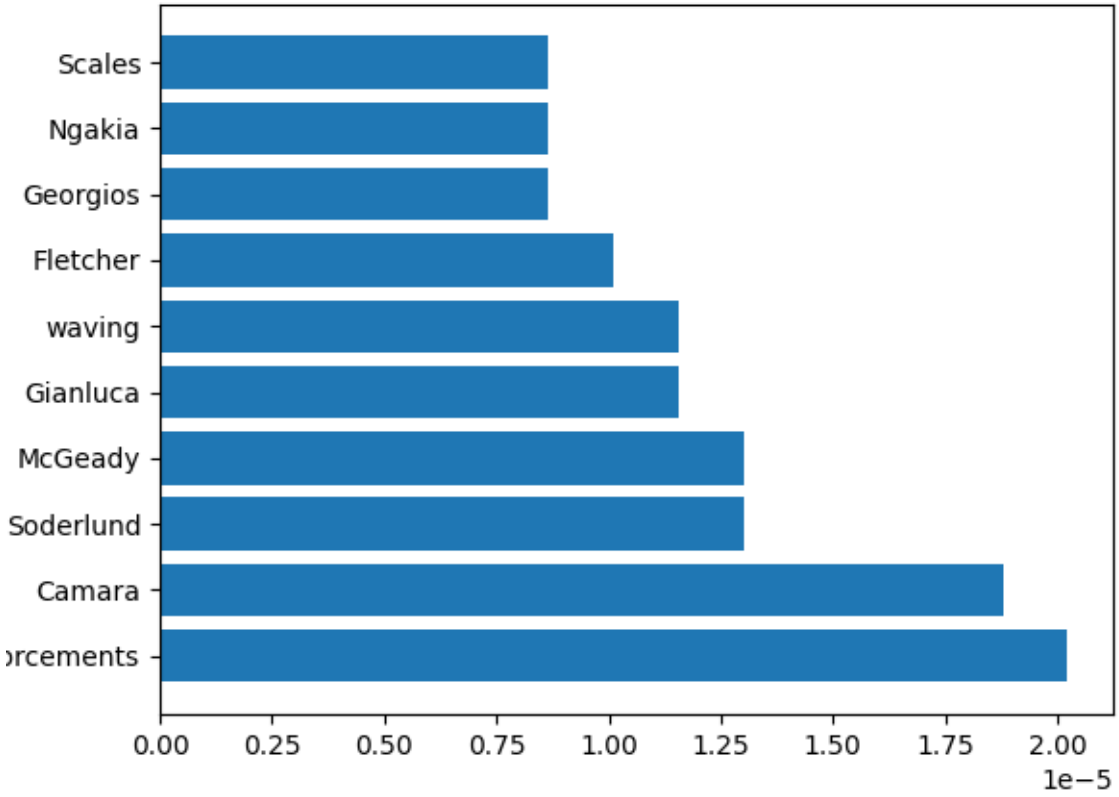
Then we should get metrics from that datas. So we write a script thet get all metrics. In SeperatedDataGetMetrics.py all metrics of seperated datas has been gotten. You can see all metrics as follow:

	Unnamed: 0	keys	values
0	0	Number of positive sents	24616
1	1	Number of negative sents	14597
2	2	Number of all sents	39213
3	3	Number of positive words	780056
4	4	Number of negative words	479749
5	5	Number of all words	1259805
6	6	Number of unique positive words	20253
7	7	Number of unique negative words	16918
8	8	Number of unique all words	22778
9	9	Number of only positive words	5860

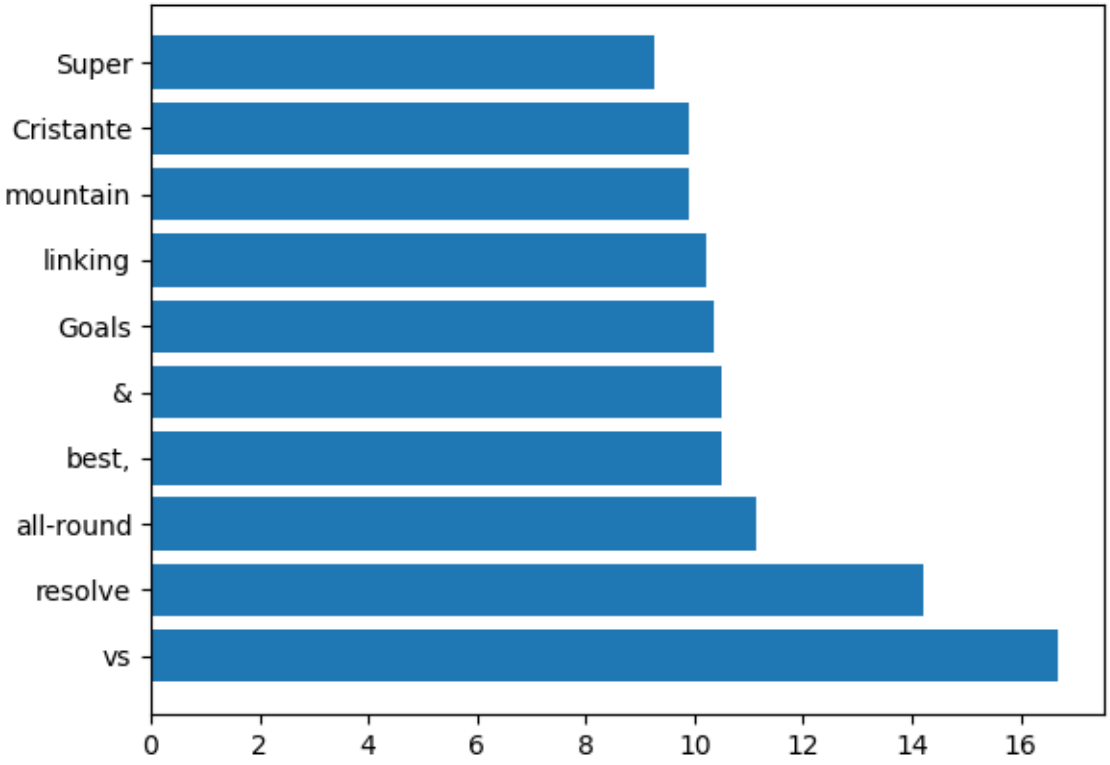
top10CommonNegativeWordsFrequency:



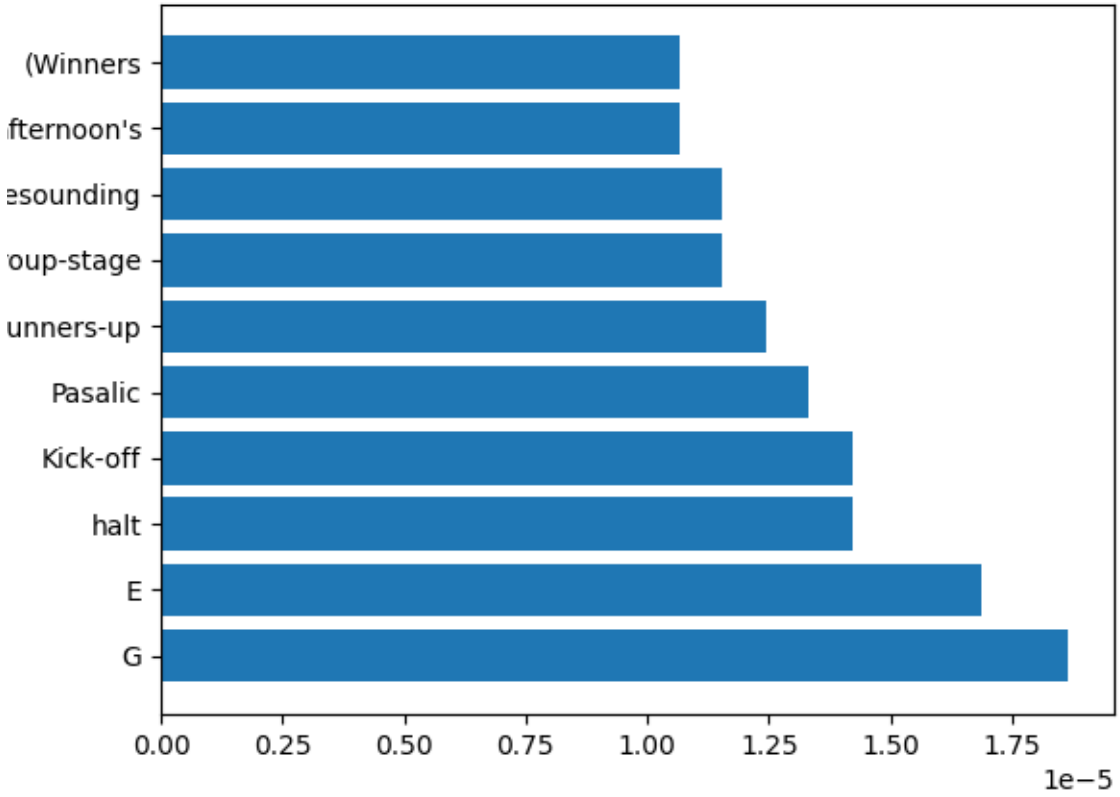
top10CommonNegativeWordsTFIDF:



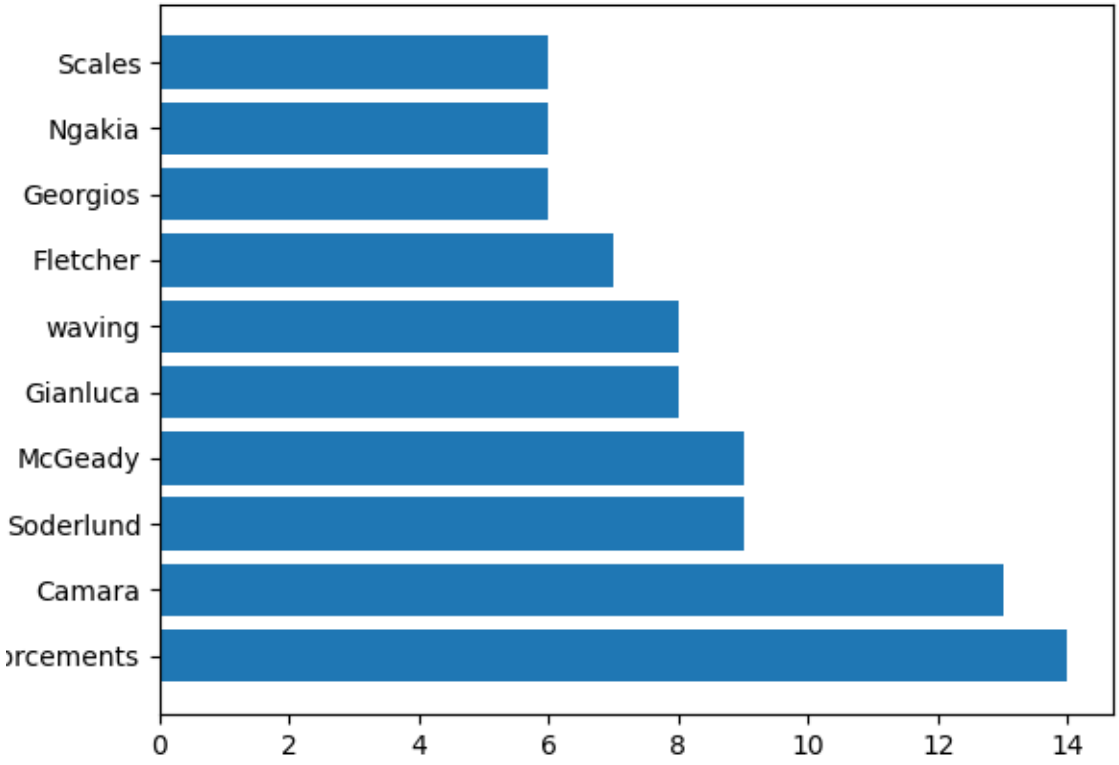
top10CommonPositiveWordsFrequency:



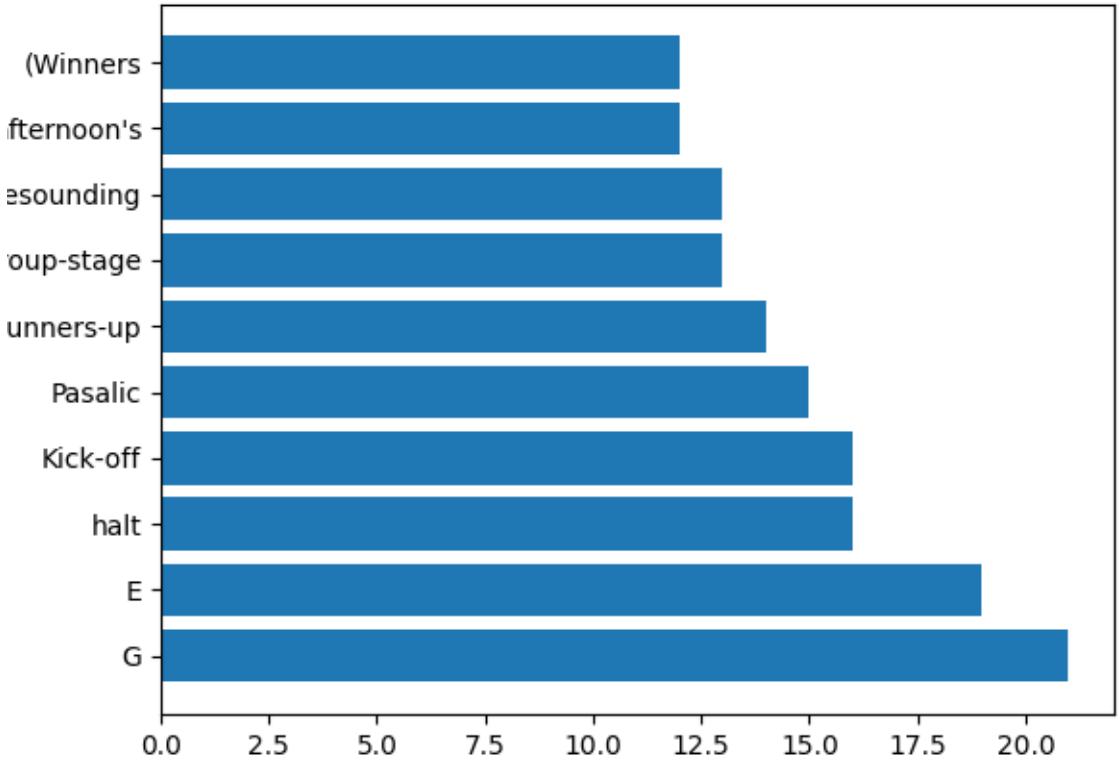
top10CommonPositiveWordsTFIDF



top10NegativeWordsCount



top10PositiveWordsCount



After that we should get metrics from new datasets.

2 Phase 2: Model Implimentation

Note: This part has been done in google colab environment at the link below.
https://colab.research.google.com/drive/1Sg-9LCH0BUBljELWQ9AeB7iqsqaRaZCH
All results of the project are in this workstation.

2.1 Installing and Importing Dependencies

At the beginning we should install and import some libraries:

- Sentence Transformer: Sentence Transformers is a Python library that provides pre-trained models for generating high-quality sentence embeddings. It enables you to convert sentences or text snippets into fixed-length

numerical representations called embeddings, which capture the semantic meaning and contextual information of the input text.

- Sklearn: Scikit-learn (or sklearn for short) is a popular open-source machine learning library in Python. It provides a wide range of tools and algorithms for data preprocessing, feature selection, model training, evaluation, and prediction.

- Pandas: Pandas is a popular open-source data manipulation and analysis library for Python. It provides easy-to-use data structures and data analysis tools, making it useful for tasks such as data cleaning, transformation, and analysis.

- Numpy: NumPy is a powerful Python library for scientific computing and data manipulation. It stands for "Numerical Python." NumPy provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently.

2.2 Preprocessing Sentence broken dataset

Then we should load sentence broken dataset that we have built. Then we should make that dataset in form that be able to feed to our models. For this goal, at first we seperate sentences about each player in each match. After that we have achived a dataset about players with label True and False. Then we change lables from True and False into 0 and 1. In addition, we get sentence embeddings of sentences about each players. After all, we have a dataset contains players'name, sentence embeddings of the player and his rating in the match.

2.3 Feature Extraction

Feature selection by clustering is a technique used to select a subset of relevant features from a larger set of available features in a dataset. It combines the principles of feature selection and clustering algorithms to identify the most informative features for a given task.

The process typically involves the following steps:

- Clustering: Initially, a clustering algorithm like K-means, hierarchical clustering, or DBSCAN is applied to the dataset using all the available features. This step groups similar instances or data points together based on their feature values.

- Feature Importance Calculation: After clustering, a measure of feature importance is computed within each cluster. This measure can be based on various criteria such as intra-cluster variance, inter-cluster variance, or other distance measures. The goal is to determine which features contribute the most to the differences between clusters and have a high impact on the clustering result.

- Feature Ranking: Once the feature importance scores are calculated, the features are ranked based on their scores. The higher the score, the more important the feature is considered to be in distinguishing between clusters.

- Feature Selection: Finally, a subset of the top-ranked features is selected as the final set of features to be used for subsequent analysis or modeling tasks. Depending on the specific requirements, a threshold may be defined to determine the number or percentage of features to be selected.

The advantage of feature selection by clustering is that it takes into account the underlying patterns and relationships within the data to guide the selection process. By considering the information provided by the clustering algorithm, it aims to retain the most discriminative features while eliminating redundant or irrelevant ones. This can lead to improved efficiency, interpretability, and generalization performance of machine learning models.

For this goal at first, we have clustered all Sentence embeddings in the dataset into 50 clusters. Each cluster should be a feature of each player; And then feed sentence embeddings of each player into the kmeans model and get 50 features of him. Finally we have 50 features of each player. Append these 50 features into the dataset and getting ready for feeding this dataset into a model for classification.

2.4 Classification of players

Sklearn, short for scikit-learn, is a popular machine learning library in Python that provides various algorithms and tools for classification, regression, clustering, and more. To perform classification using sklearn, you typically follow these steps:

- Data Preparation: Firstly, you need to prepare your dataset by splitting it into input features X and target variable y. X represents the independent variables/features, while y represents the dependent variable/class labels you want to predict.

- Train and Test Split: It is common practice to split your dataset into training and testing subsets. The training set is used to train the classifier, while the testing set is used to evaluate its performance on unseen data.

- Choose a Classifier: Select an appropriate classifier algorithm from sklearn based on your problem requirements. Sklearn offers various classification algorithms such as Logistic Regression, Support Vector Machines, Random Forest, etc.

- Instantiate and Train: Create an instance of the chosen classifier and then fit/train it on the training data using the fit method. This step involves learning the patterns and relationships between the features and class labels.

- Prediction: Once the classifier is trained, you can use it to make predictions on new, unseen data. Use the predict method or related methods of the classifier to generate predictions based on the learned model.
- Evaluate Performance: Compare the predicted class labels with the actual class labels of the test set to evaluate the performance of your classifier.

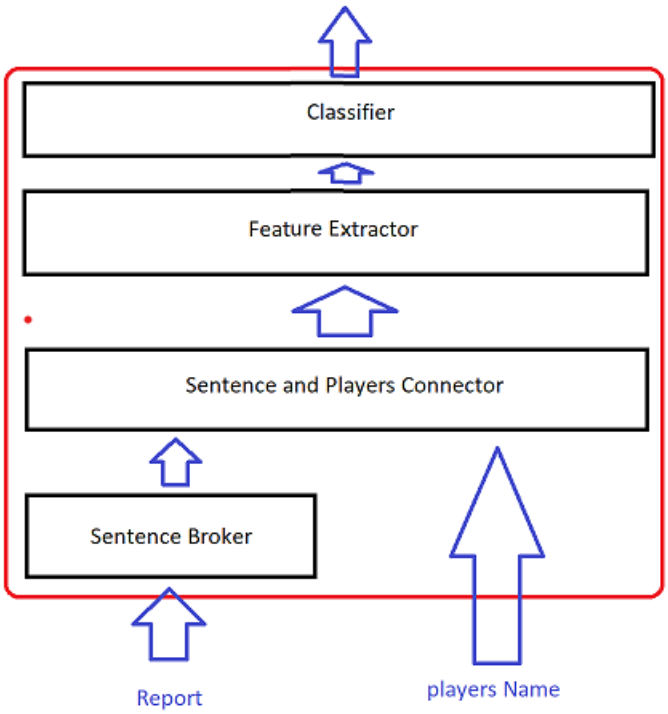
We should classify all players by their 50 features in 2 classes. We use sklearn linear model to do this. First 1900 entries are datas for train and last 100 datas are datas for test. All 50 features are X and rating is Y.

2.5 Finalizing Project

Finally we have 3 trained models:

- Sentence Transformer: This model gets a list of sentences and give the list of embeddings of that sentences.
- Clustering Model: This model gets a list of Sentence embeddings and give the cluster of that sentences. This model is being used for feature Extraction of players.
- Classification Model: This model gets features of a player and classify that player.

Here is Final Model:



This model gets report and 22 players name of a match. Then it breaks sentences for each player and then gets their embeddings. After that it Extracts 50 features of each players. Finally it classify players by their features.