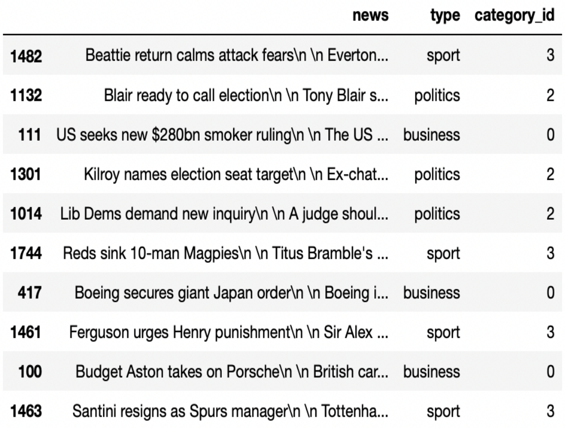
Part 2

Abstract:

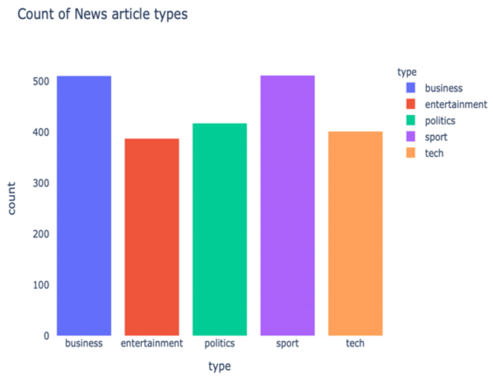
Because of the overwhelmingly increasing volume of news corpus on the internet, news article classification has become a major area of concern in the realm of text classification problems. These news stories were chosen from a dataset of 2225 British Broadcasting Corporation (BBC) news records for categorical research. The dataset includes 2225 news articles divided into five topical categories: politics (417), sport (511), entertainment (386), tech (401), and business (510). In this part, removal of stop-words, stemming, Lemmatization and word frequency were performed. For the feature engineering and selection, I have used chi2 test. Logistic Regression is used for the prediction.

Description:

The dataset contains 2225 articles in the form of .txt files and these articles are divided into five categories, which are politics, sport, business, tech, and entertainment. The first task for this part is to append all .txt file articles into one CSV file. With the use of os and pandas library and for loop, I have firstly specified the path of the dataset and then appended it into one data frame named df, where mainly two variables are there. One is news, which is basically an article and the other is type, which specifies the category of the article. The CSV dataset has been exported to local storage for further use. Then the new variable category_id has been added, which specifies numeric transformation of type variable. After that, for pre-processing, I have used stop-words, word stemming, lemmatization and word frequency. Due to that, the data is cleaner, and the dimensions are reduced. I have used the standardised TF-IDF technique for feature engineering. Here, sublinear_df is set to True by using a frequency logarithm to decrease returns as a word's frequency rises. Min_df is the minimum number of documents in which a term must be held, and we set it to 5. This is to prevent unusual terms that greatly raise the scale and overfitting of our features. To ensure a euclidian norm of 1 in all our functional vectors is set to l2. The latin-1 encoding that our input text uses is set for. The ngram range is set to (1, 2) to indicate that both unigrams and bigrams are to be considered. To delete all traditional pronouns ("a," "the",...) and further decrease the number of noisy elements, stop word is set to "English." We have also used the chi square test for feature selection, which includes a bag of words, unigram and bigram. Using chi-square analysis to find corelation between news (importance of words) and ids (news category). Here we are going to look for top-3 categories as n=3. First for each category, find words that are highly corelated to it. The next step of feature selection in chi square is Converts indices to feature names in increasing order of chi-squared stat values. Unigrams are List of single word features in increasing order of chi-squared stat values. Bigrams are for List for two-word features in increasing order of chi-squared stat values. Here we Print 3 unigrams and bigrams with highest Chi squared stat. Using the sklearn library, the data was divided into trains and tests in a 70:30 ratio. For the model selection, Logistic Regression has been considered for prediction, which gives 97.75% accuracy and an RMSE value of 0.36.



	news	type	category_id
1482	Beattie return calms attack fears\n \n Everton...	sport	3
1132	Blair ready to call election\n \n Tony Blair s...	politics	2
111	US seeks new $280bn smoker ruling\n \n The US ...	business	0
1301	Kilroy names election seat target\n \n Ex-chat...	politics	2
1014	Lib Dems demand new inquiry\n \n A judge shoul...	politics	2
1744	Reds sink 10-man Magpies\n \n Titus Bramble's ...	sport	3
417	Boeing secures giant Japan order\n \n Boeing i...	business	0
1461	Ferguson urges Henry punishment\n \n Sir Alex ...	sport	3
100	Budget Aston takes on Porsche\n \n British car...	business	0
1463	Santini resigns as Spurs manager\n \n Tottenha...	sport	3



Pre-Processing:

The pre-processing part contains four main parts, which are following.

Stop-words:

The BBC News stop word lists used to sort out stop words that often happen but don't summarise the important facts in the news classification process.

Word Stemming:

A word filtering procedure was carried out until the final classification process eliminated the non-descriptive tokens of less than three characters. Stemming is the basic/system/root word of words heuristic to the effect that the antifixes are removed and inflective forms decreased. Due to its greater utility than other basic methods, such as Paice/Husk, S and Lovins, Porters Stiemmer was used.

Lemmatization:

Contrary to Stemming, lemmatization eliminates the words inflected to make sure that the root word is in the vocabulary. The term source is Lemma in lemmatization. A lemma is a canonical form, a dictionary form or a citation form of a group of terms (plural lemmas or lemmata).

Word Frequency:

To count the words repetition and reduce the dimensions by removing least repeating words.

Feature Engineering and Selection:

Feature extraction is an important measure to reduce the dimensionality to ensure classification accuracy and increased time efficiency (Fagbola et al., 2012; Fagbola et al., 2017). Relevant features were obtained using a standardised TF-IDF technique from the words returned during pre-processing. TF-IDF is a space vector strategy that conveniently numerically makes the weight of terms (Joho and Sanderson, 2007). It has been embraced because of its very precise results in comparison to most other mathematical methods.

For each of our articles, we are using TfidfVectorizer to measure a tf-idf vector. sklearn.feature.text.device. To measure the tf-idf vector for each of our papers, TfidfVectorizer will be used.

Other features are also used for feature engineering, which are following:

Bag of Words:

The use of this model is a popular method to derive text-features: a model in which the existence (and usual frequencies) of terms (and the order in which the word occurs are ignored) are considered for each document.

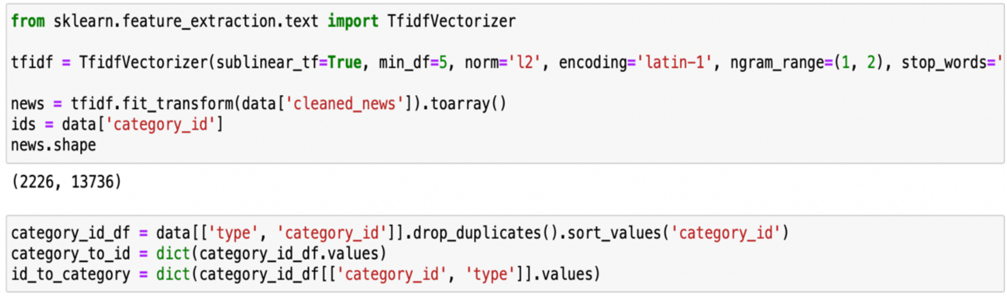
Unigram:

We presume in Unigram that any word occurs independently of the previous word. Therefore, here every word becomes a gramme.

Bigram:

The single words(bigram) or word series and its frequency are returned in this function. In Bigram, any occurrence depends only on the preceding word of each word. Therefore, here two words are counted as one gramme.

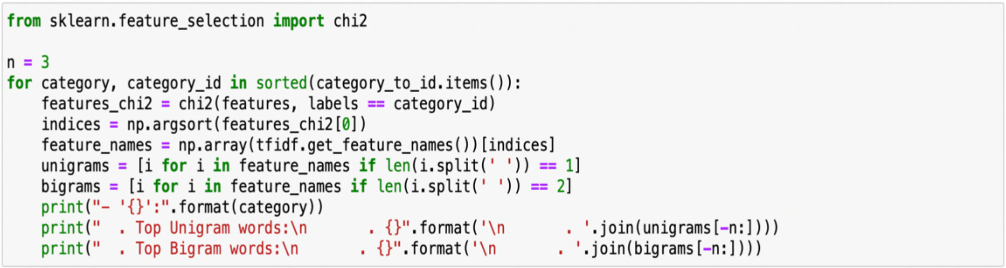
Using chi-square analysis to find corelation between news (importance of words) and ids (news category). Here we are going to look for top-3 categories as n=3.



```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer(sublinear_tf=True, min_df=5, norm='l2', encoding='latin-1', ngram_range=(1, 2), stop_words='

news = tfidf.fit_transform(data['cleaned_news']).toarray()
ids = data['category_id']
news.shape
```

```
(2226, 13736)
```

```
category_id_df = data[['type', 'category_id']].drop_duplicates().sort_values('category_id')
category_to_id = dict(category_id_df.values)
id_to_category = dict(category_id_df[['category_id', 'type']].values)
```



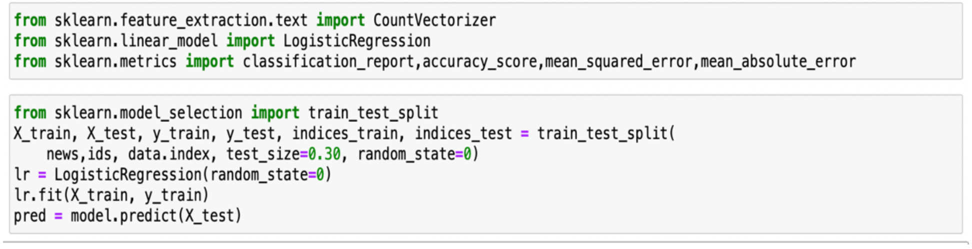
```
from sklearn.feature_selection import chi2

n = 3
for category, category_id in sorted(category_to_id.items()):
    features_chi2 = chi2(features, labels == category_id)
    indices = np.argsort(features_chi2[0])
    feature_names = np.array(tfidf.get_feature_names())[indices]
    unigrams = [i for i in feature_names if len(i.split(' ')) == 1]
    bigrams = [i for i in feature_names if len(i.split(' ')) == 2]
    print("- '{}':".format(category))
    print("  . Top Unigram words:\n       . {}".format('\n       . '.join(unigrams[-n:])))
    print("  . Top Bigram words:\n       . {}".format('\n       . '.join(bigrams[-n:])))
```

Train and Test:

To split the data into train and test, I have used train_test_split from sklearn model selection library. The train and test size ratio in for the model is 70:30, while random state is 0 during the splitting. Train and Test data taken from news and ids variables, which are basically independent and dependent variables.

Model Selection:

With the use of sklearn library I have imported logistic regression model and created it as **lr** in the code with random_state = 0. Earlier created train and test datasets fitted into the logistic regression model for the predictions. Then using model.fit function the values have been predicted.



```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report,accuracy_score,mean_squared_error,mean_absolute_error
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test, indices_train, indices_test = train_test_split(
    news,ids, data.index, test_size=0.30, random_state=0)
lr = LogisticRegression(random_state=0)
lr.fit(X_train, y_train)
pred = model.predict(X_test)
```

Model Evolution:

As we can see into the code, the accuracy of the model is 97.45% while root squared mean error is 0.36.

Accuracy - Accuracy - The number of accurate predictions made in a ratio of all predictions is the accuracy of classifications.

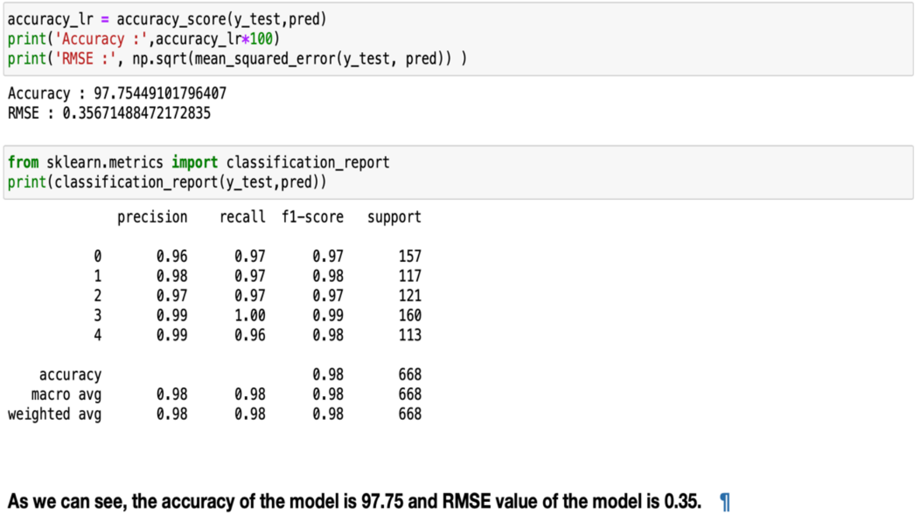
Precision - Precision is the fraction of valid instances among the retrieved instances (also known as positive predictive value).

F1_score - Consider both the accuracy and recall calculating the score.

Recall – Recall (also known as sensitivity) is a percentage of the related instances recovered over the total number of instances in question.

Why are these metrics being used? - For testing our model, we used Accuracy, Precision, F1 Score, and Recall as parameters because accuracy would provide an approximation of accurate prediction. Precision will provide us with an estimation of the positive category expected significance, i.e. how well our model is producing relevant results. The F1 Score provides a summed approximation of accuracy and recall. The related positive category forecast to the false negative and true positive category recognition results will be given by recall.

I have also included classification report, which shows macro-precision, macro-recall, macro-f1-score.



```
accuracy_lr = accuracy_score(y_test,pred)
print('Accuracy :',accuracy_lr*100)
print('RMSE :', np.sqrt(mean_squared_error(y_test, pred)) )
```

```
Accuracy : 97.75449101796407
RMSE : 0.35671488472172835
```

```
from sklearn.metrics import classification_report
print(classification_report(y_test,pred))
```

```
              precision    recall  f1-score   support

           0       0.96      0.97      0.97       157
           1       0.98      0.97      0.98       117
           2       0.97      0.97      0.97       121
           3       0.99      1.00      0.99       160
           4       0.99      0.96      0.98       113

    accuracy                           0.98       668
   macro avg       0.98      0.98      0.98       668
weighted avg       0.98      0.98      0.98       668
```

As we can see, the accuracy of the model is 97.75 and RMSE value of the model is 0.35. ¶

Literature Review

Text Classification Problem:

A multi-label text classification dilemma is the classification of news articles. Assuming a hyperplane function: O P(C) and an undefined target function: O p(C), text classification is demonstrated by estimating using, such that each instance x O is mapped to 0 or more predetermined classes C = c1,..., c|C| (Skjennum, 2016). Given an arbitrary problem with function: X C, where X is the feature space and C = C1, C2,..., Cn is a special distribution of classes with n as the number of unique classes, the class, Ci, I n, of an undefined test input X is calculated by $\boldsymbol{\gamma}$.

An observation Z may be described in the form (x, y), where x is the input into the learning protocol and y is the expected output using y = f(x). In other words, with a training dataset TD described as:

$T = \{z = (x,y)\}^{N}$ (1) $_{D\ n\ n\ n\ n=1}$

A increasing trend of new, user-aware, big data analytic terms with labels like “user-assisted classification”, “interactive classification”, “user-aware classification”, and “user-centred classification” has recently emerged. Incorporate customer comments, recommendations, scores, and personalised viewpoints into their classification process to improve the accuracy of automated/semi-automated classification decisions (Donkers et al., 2018). This strategy, for example, incorporates customised and sentiment-enhanced recommender services (Yibo et al., 2018). However, this approach is best suited for unstructured data analysis (Donkers et al., 2018).

where N = $|T_D|$, that is, the size of T_D, the expected error rate of a hypothesis h, is defined as (Van Meeuwen, 2013):

$\pi(\mathrm{h}) = {}^{E}\, [\![\mathrm{h}(x) \neq f(x)]\!]$ (2) $_{x \sim P_x}$

where the Boolean evaluation $[\![\mathrm{h}(x) \neq f(x)]\!]$ is 1 if the relation is true and 0 otherwise, while P_x depicts an unknown probability distribution x. The primary aim of learning algorithms is to pick a h with a low π(h) in order to improve the accuracy of a classification method. As a result, a learning algorithm is used to find a h with the fewest errors in the training results, TD, in such a way that (Li & Mostafa, 2006):

$e_D(\mathrm{h}) = \sum^{N}{}_{n=1} [\![\mathrm{h}(x_n) \neq y_n]\!]$, (3) where e_D(h) is the hypothesis with the least number of learning errors.

(Zhang et al., 2016) proposed a character-level coding convolutional network for classifying news datasets using "small and large convolution," "bag of means," "bag of sentences," and "small and large complete convolution." Gurmeet and Karan (2016) used ANN and a Support Vector Machine to classify a 1000-article subset of BBC news into categories such as industry, health, culture, and sports (SVM). The published paper, however, did not provide clear methodological specifics on measures such as the network topology used, data pre-processing, tokenization and stop-word elimination, and function selection. Among others who have contributed to this work are Chan et al. (2001) created a traditional online news categorizer scheme of personalization for Channel Asia's financial news classification by employing an SVM of ten distinct categories from the Reuters-21578 sets. Skjennum (2016) developed and introduced a versatile and scalable framework to identify 1.8 million news stories from a New York Times annotated corpus using an ensemble of feedforward multilayer perceptron networks, n-binary multinominal Nave Bayes, and a Long Short-Term Memory Recurrent Neural Network. However, the established multilingual classification system's dependability and efficiency could not be checked.