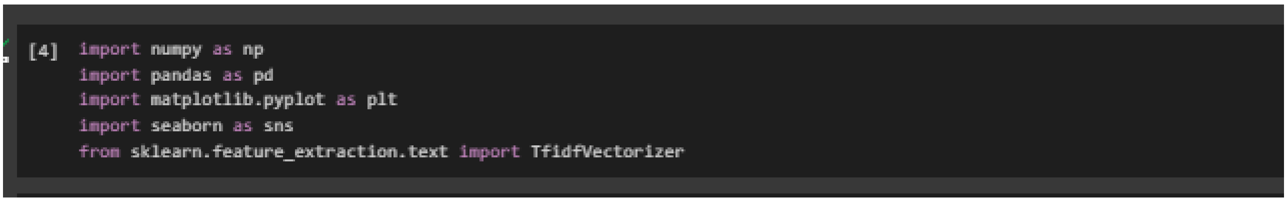
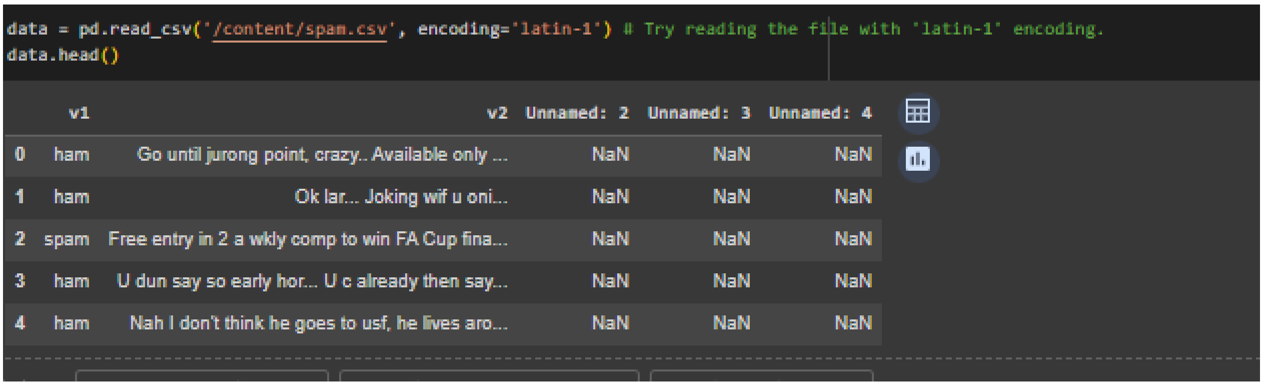
Project on Email spam detection using Machine Learning

- **Aim:-**To create a Data science Project, where we will be detection of Email spam. In this Project, use Python to build an email spam detector. Then, use machine learning to train the spam detector to recognize and classify emails into spam and non-spam.
- Steps to be taken in the project is sub-divided into the following sections. These are:
    - Importing the libraries such as ‘numpy’, ‘pandas’,‘sklearn model’ etc.
    - Loading Dataset as a CSV file for training & testing the models.
    - Splitting the data set into independent & dependent sets.
    - Checking if still any null values or any other data types other than float and integers are present into the dataset or not.
    - Importing the train_test_split model from sklearn.model for splitting data into train & test sets.
    - Applying the different kinds of ML Algorithms .which gives Best accuracy of model.
    - Also checking with new data set for predicting the values.
- Steps of creating ML model:-
- Importing numpy as np & pandas as pd for loading and reading the data-set & using matplotlib.pyplot and Seaborn for visualization of data.



```
[4] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    from sklearn.feature_extraction.text import TfidfVectorizer
```

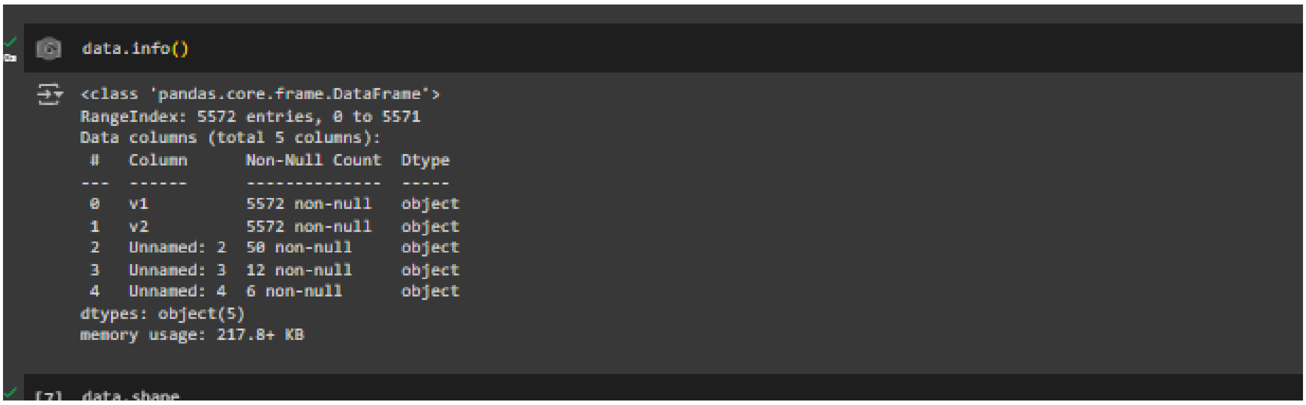
- Loading the csv-dataset in the variable name ‘data’ Then viewing the data with data.head()



```
data = pd.read_csv('/content/spam.csv', encoding='latin-1') # Try reading the file with 'latin-1' encoding.
data.head()
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

- Checking the data such as number of columns, rows and type of data(float,integer) with help data.info()

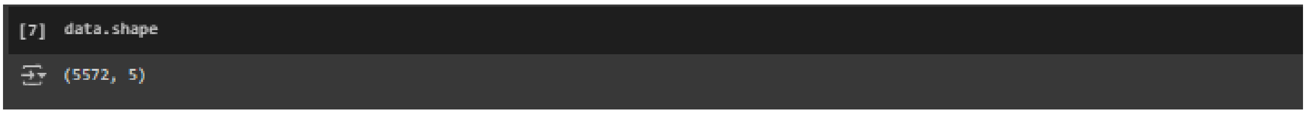


```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   v1          5572 non-null   object
 1   v2          5572 non-null   object
 2   Unnamed: 2  50 non-null     object
 3   Unnamed: 3  12 non-null     object
 4   Unnamed: 4  6 non-null      object
dtypes: object(5)
memory usage: 217.8+ KB
```

```
[7] data.shape
```

We observe that the above data have object.

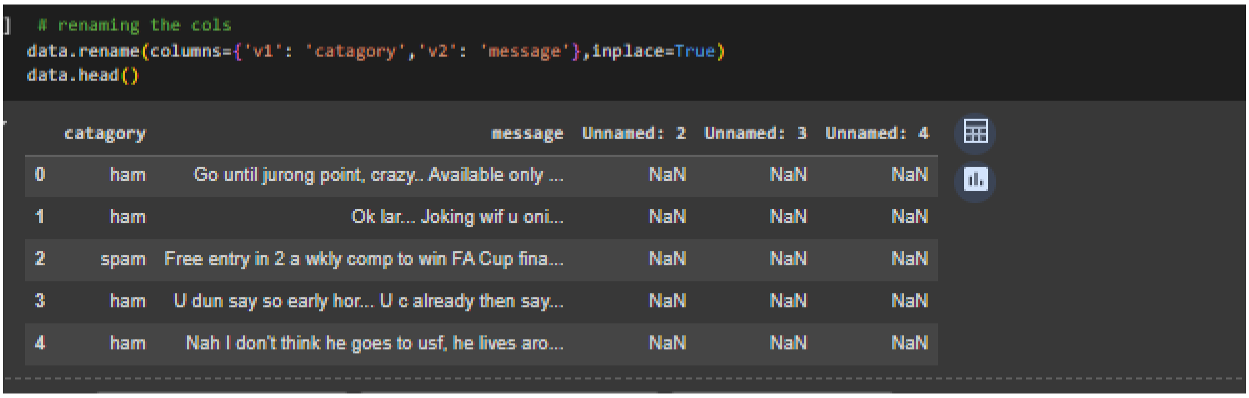


```
[7] data.shape
```

```
(5572, 5)
```

Train data have 5572 Rows and 5 columns

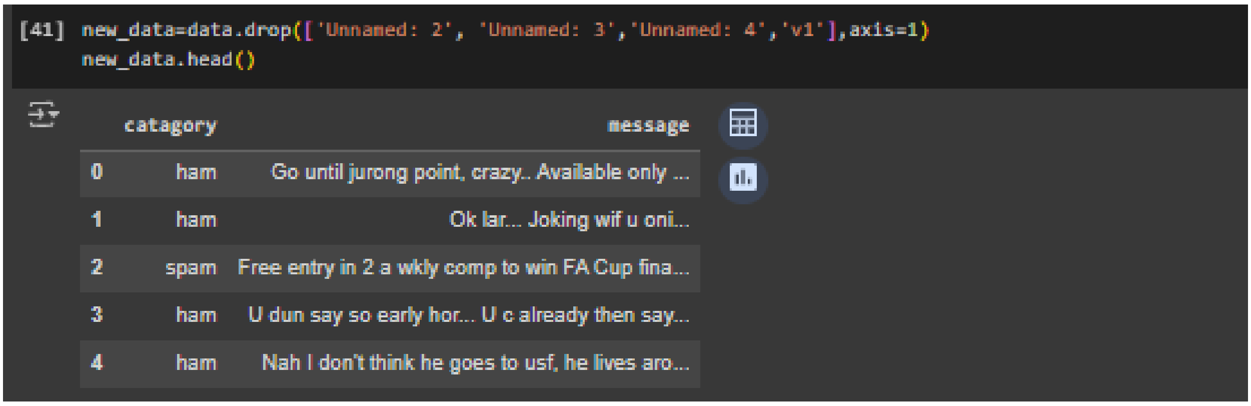
Rename the columns.



```
# renaming the cols
data.rename(columns={'v1': 'catagory','v2': 'message'},inplace=True)
data.head()
```

	catagory	message	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

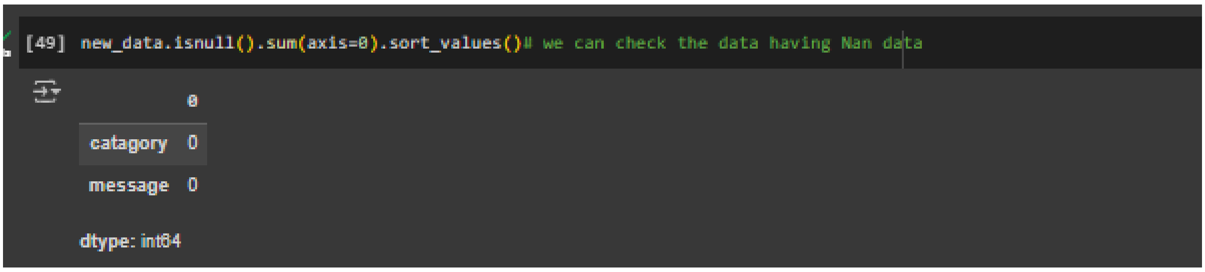
Remove the Nan value columns



```
[41] new_data=data.drop(['Unnamed: 2', 'Unnamed: 3','Unnamed: 4','v1'],axis=1)
     new_data.head()
```

	catagory	message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

- Now checking data have Nan value or not.



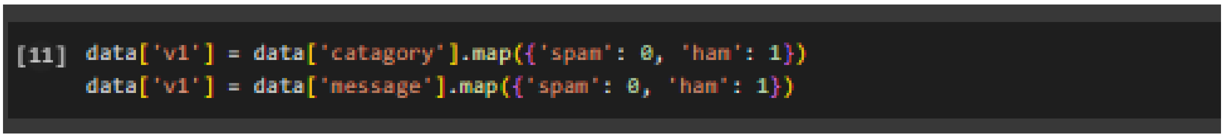
```
[49] new_data.isnull().sum(axis=0).sort_values()# we can check the data having Nan data

              0
    catagory  0
    message   0

   dtype: int64
```

We observe that the above data have not Nan value.

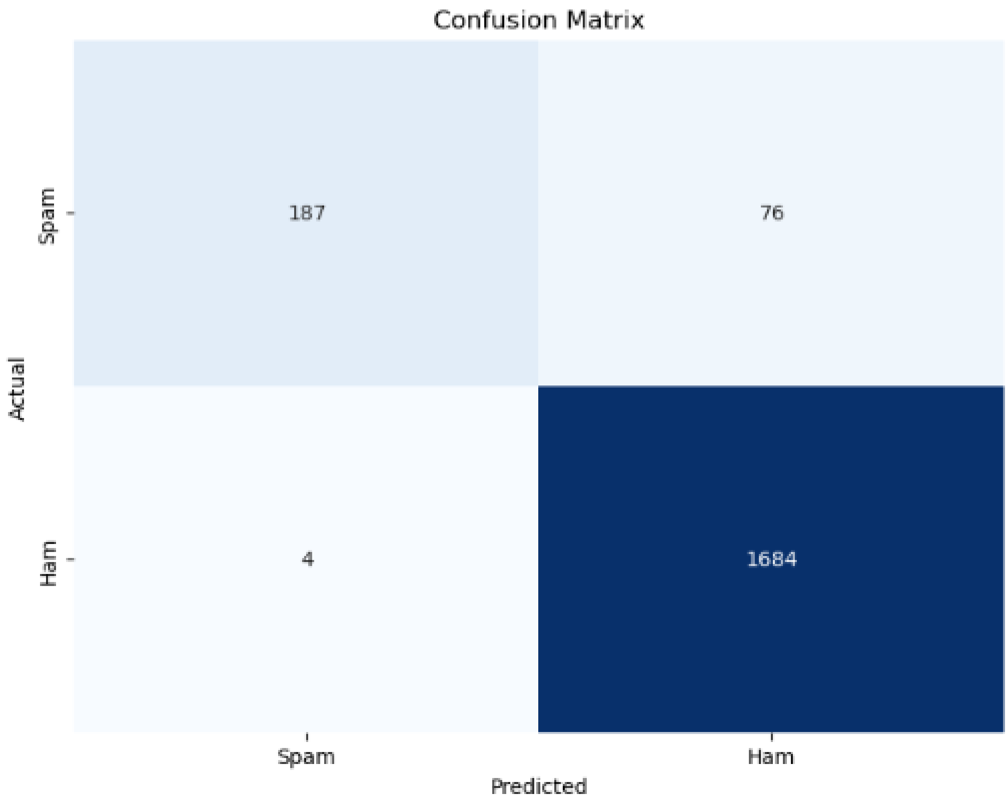
- Now,Main focus convert the categorical data into Numerical data with help of one hot encoding method.



```
[11] data['v1'] = data['catagory'].map({'spam': 0, 'ham': 1})
     data['v1'] = data['message'].map({'spam': 0, 'ham': 1})
```

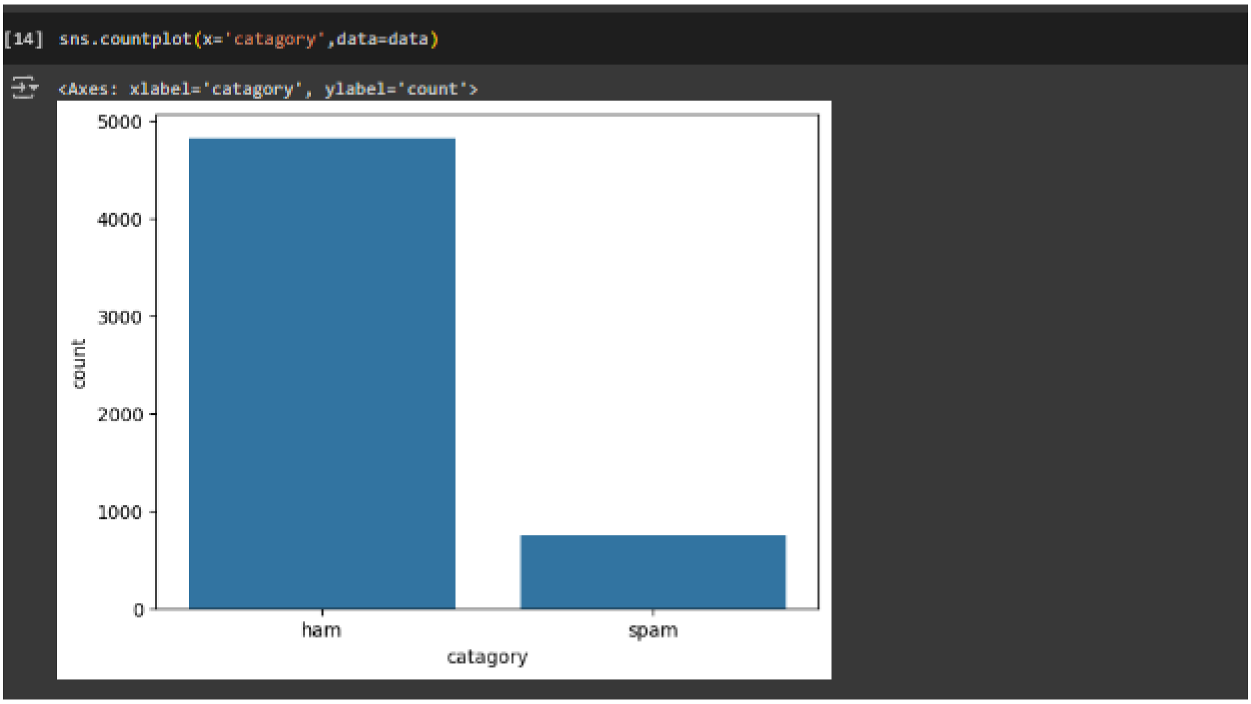
Finally we observe the data are fully cleaned.

- Now we check the data dependency.



We see that data dependent each other.

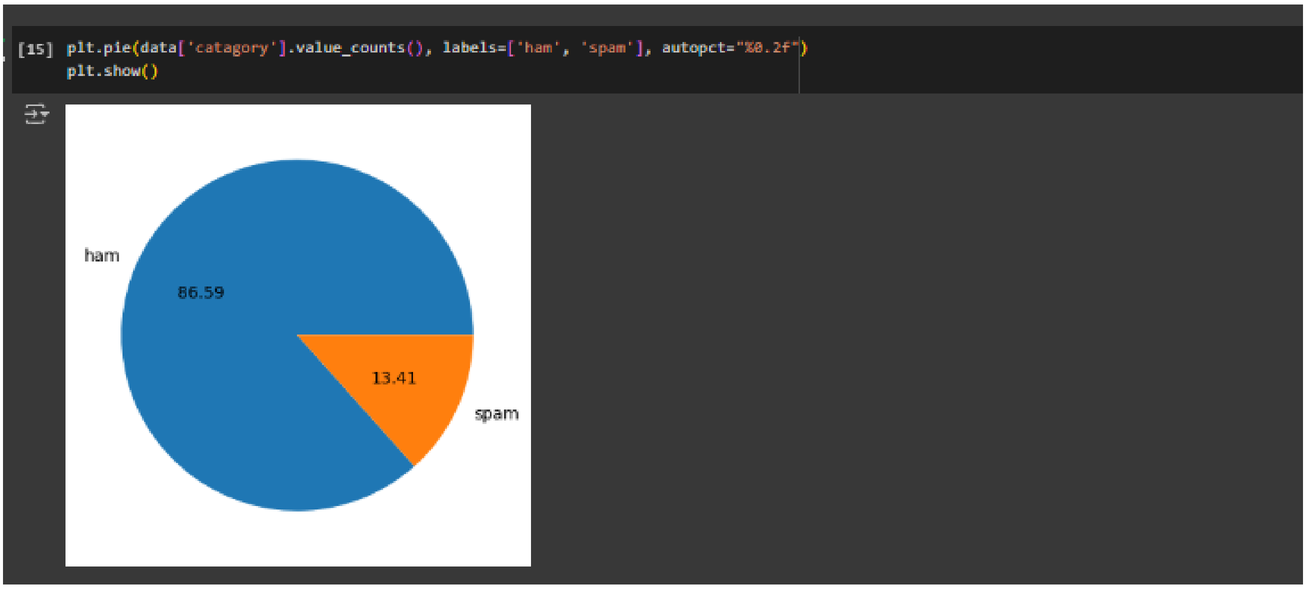
- Visualizing the the spam or ham



```
[14] sns.countplot(x='catagory',data=data)
```

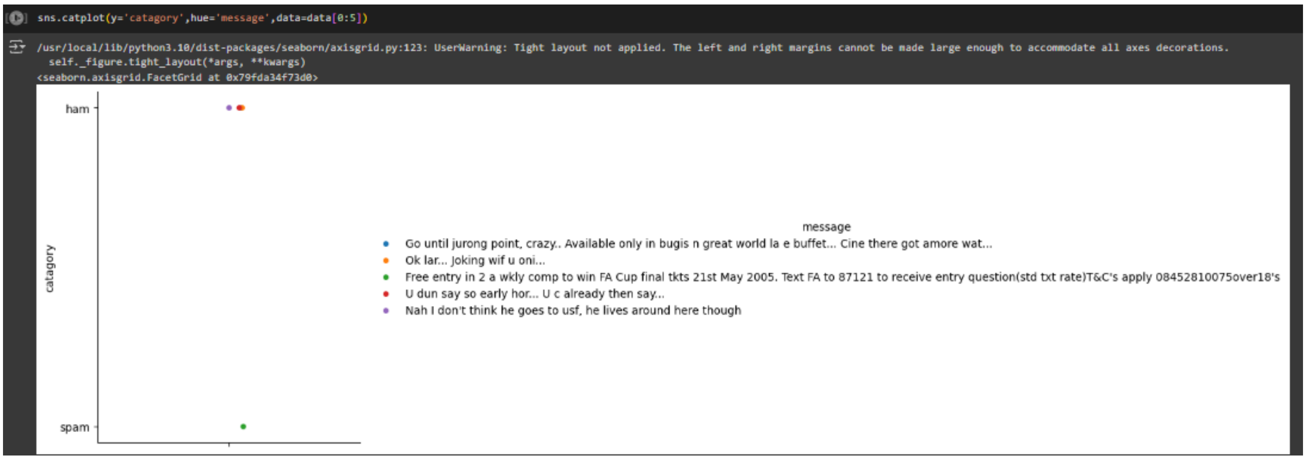
```
<Axes: xlabel='catagory', ylabel='count'>
```

As per Visualizing the above graph, Ham message is more than Spam message.



```
[15] plt.pie(data['catagory'].value_counts(), labels=['ham', 'spam'], autopct="%0.2f")
     plt.show()
```

As per Visualizing the above graph 13.4 % spam and 86.59 ham

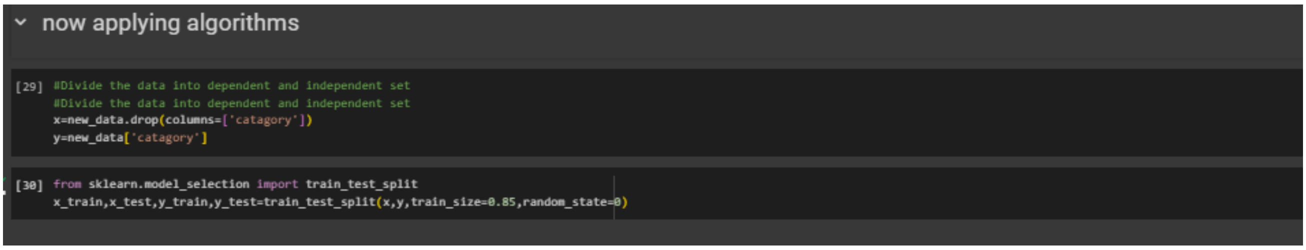


```
sns.catplot(y='catagory',hue='message',data=data[0:5])
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:123: UserWarning: Tight layout not applied. The left and right margins cannot be made large enough to accommodate all axes decorations.
  self._figure.tight_layout(*args, **kwargs)
<seaborn.axisgrid.FacetGrid at 0x79fda34f73d0>
```

- Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
- Ok lar... Joking wif u oni...
- Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
- U dun say so early hor... U c already then say...
- Nah I don't think he goes to usf, he lives around here though

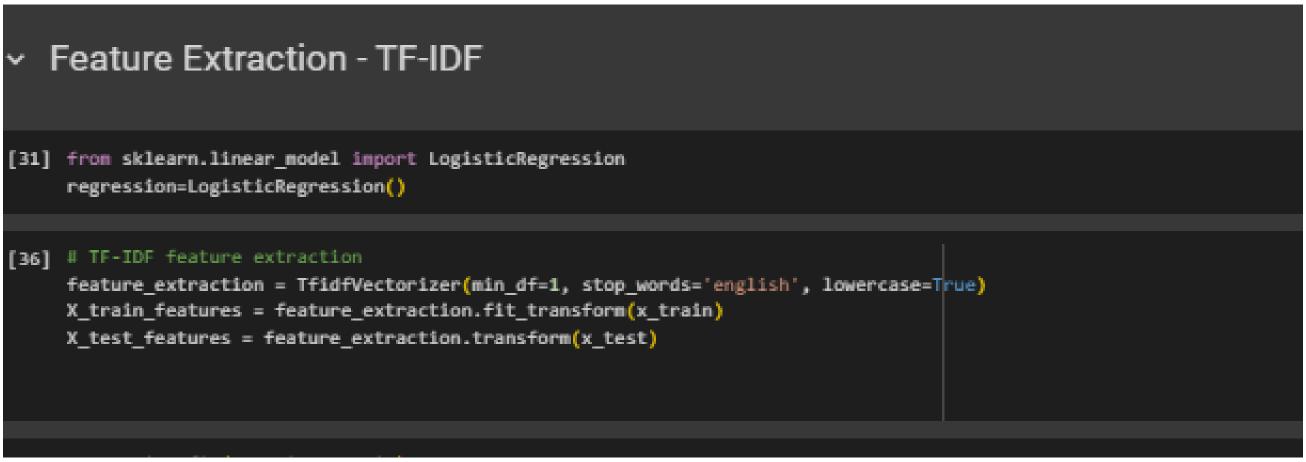
➢ Importing train_test_split from sklearn.model library for splitting the data into train and test sets. (we consider train dataset).



now applying algorithms

```
[29] #Divide the data into dependent and independent set
     #Divide the data into dependent and independent set
     x=new_data.drop(columns=['catagory'])
     y=new_data['catagory']
```

```
[30] from sklearn.model_selection import train_test_split
     x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.85,random_state=0)
```

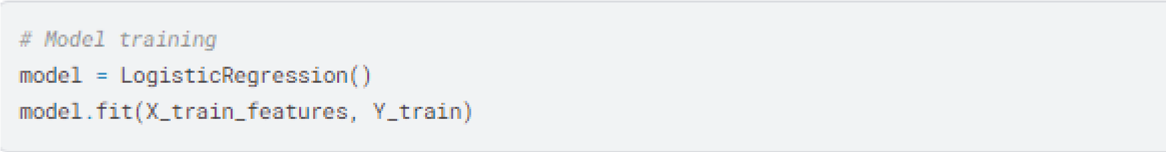


Feature Extraction - TF-IDF

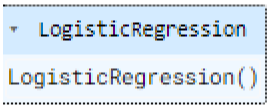
```
[31] from sklearn.linear_model import LogisticRegression
     regression=LogisticRegression()
```

```
[36] # TF-IDF feature extraction
     feature_extraction = TfidfVectorizer(min_df=1, stop_words='english', lowercase=True)
     X_train_features = feature_extraction.fit_transform(x_train)
     X_test_features = feature_extraction.transform(x_test)
```

➢ Importing logistic regression from sklearn Libaray & then activating the Machine learning Model .Then used regression.fit() to training the model by providing train & test sets as x & y. And then predicted the trained model with help of MLM & the checked score as regression.score(x,y)

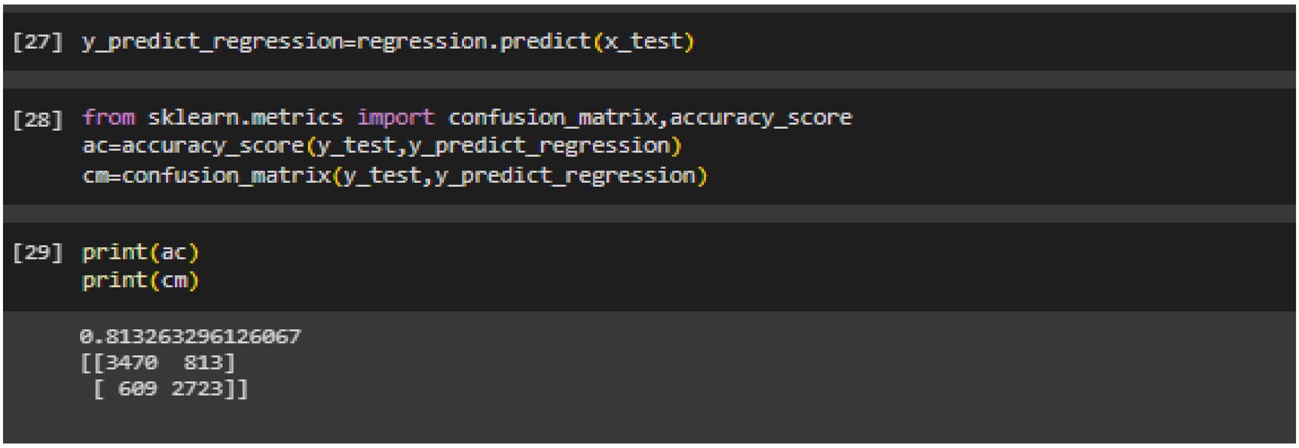


```
# Model training
model = LogisticRegression()
model.fit(X_train_features, Y_train)
```



```
▾ LogisticRegression
LogisticRegression()
```

❖ Checking the accuracy with help of confusion Matrix.



```
[27] y_predict_regression=regression.predict(x_test)

[28] from sklearn.metrics import confusion_matrix,accuracy_score
     ac=accuracy_score(y_test,y_predict_regression)
     cm=confusion_matrix(y_test,y_predict_regression)

[29] print(ac)
     print(cm)

     0.813263296126067
     [[3470  813]
      [ 609 2723]]
```

In the above model we can see that the accuracy obtained is 81%

Thank you