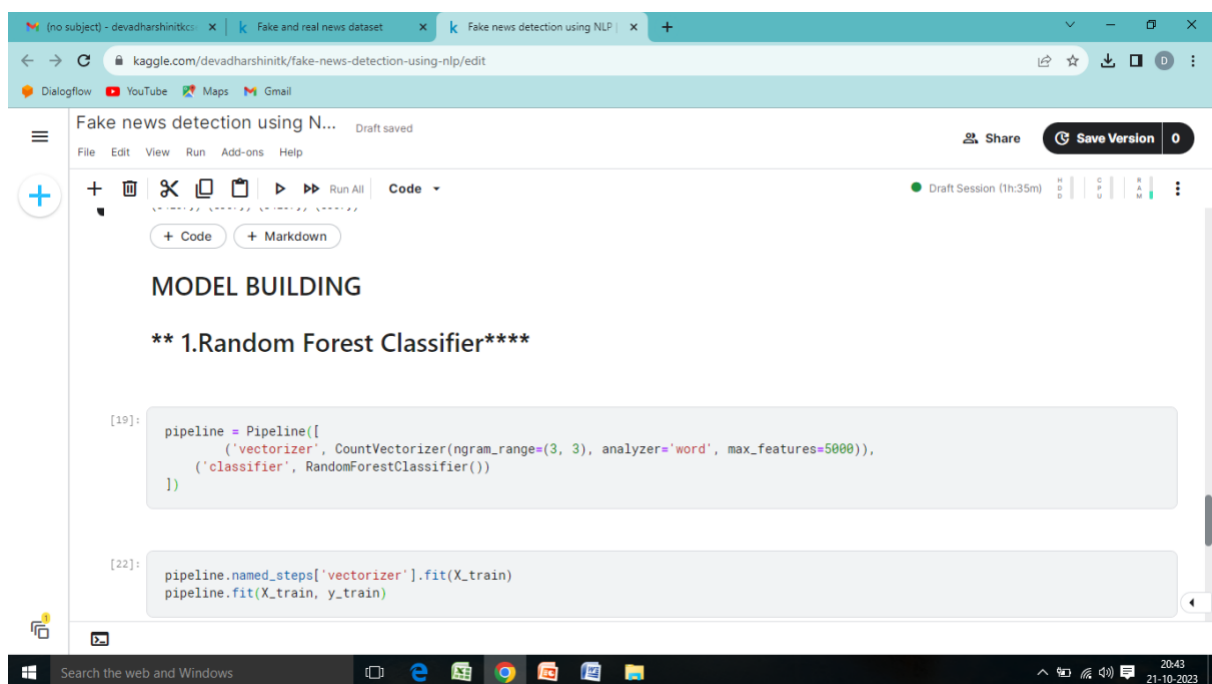


Fake News Detection Using NLP

Phase 4

- Selecting Machine Learning algorithm
- Splitting dataset into training and testing data
- Evaluating the training data using testing data
- Evaluating the model Performance



The screenshot shows a Google Colab notebook interface. The browser tabs at the top include 'no subject - devadharshinitk...', 'k Fake and real news dataset', and 'k Fake news detection using NLP'. The notebook title is 'Fake news detection using NLP'. The code editor displays the following Python code:

```
[19]: pipeline = Pipeline([
      ('vectorizer', CountVectorizer(ngram_range=(3, 3), analyzer='word', max_features=5000)),
      ('classifier', RandomForestClassifier())
    ])

[22]: pipeline.named_steps['vectorizer'].fit(X_train)
      pipeline.fit(X_train, y_train)
```

Browser tabs: (no subject) - devadharshinitk..., Fake and real news dataset, Fake news detection using NLP

URL: kaggle.com/devadharshinitk/fake-news-detection-using-nlp/edit

Fake news detection using N...

File Edit View Run Add-ons Help

Share Save Version 0

Run All Code

CountVecorizer
RandomForestClassifier

+ Code + Markdown

```
[23]: y_pred = pipeline.predict(X_test)
```

```
[24]: print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.98	0.97	0.97	4341
1	0.97	0.98	0.97	4226
accuracy			0.97	8567
macro avg	0.97	0.97	0.97	8567
weighted avg	0.97	0.97	0.97	8567

20:44 21-10-2023

Browser tabs: (4) WhatsApp, (no subject) - devadharshinitk..., Fake and real news dataset, Fake_news_classification_Simple

URL: kaggle.com/code/devadharshinitk/fake-news-classification-simple-model/edit

Fake_news_classification_Simple_Mod...

File Edit View Run Add-ons Help

Share Save Version 0

Cancel Run Code

Draft Session (6m)

Model Building

2.Logistic Regression

+ Code + Markdown

```
[13]: from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.pipeline import make_pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
```

+ Code + Markdown

```
[14]: X_train, X_test, y_train, y_test = train_test_split(concat_data.vector,
                                                    concat_data.label,
                                                    test_size = 0.2)
```

21:21 21-10-2023

4) WhatsApp (no subject) - devadharshinik... Fake and real news dataset Fake_news_classification_Simple... +

kaggle.com/code/devadharshinik/fake-news-classification-simple-model/edit

Dialogflow YouTube Maps Gmail

Fake_news_classification_Simple_Mod... Draft saved

File Edit View Run Add-ons Help

Share Save Version 0

+ Draft Session (7m)

[14]:

```
X_train, X_test, y_train, y_test = train_test_split(concat_data.vector,
                                                    concat_data.label,
                                                    test_size = 0.2,
                                                    random_state = 1,
                                                    stratify = concat_data.label)

print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```

(35918,) (35918,)
(8988,) (8988,)

[15]:

```
X_train = np.stack(X_train)
X_test = np.stack(X_test)

X_train
```

[15]: array([[-0.23301356, 4.5374684, -2.3352568, ..., 1.769452,
 -3.347071, 3.5270379],
 [-0.31101125, 0.70727515, -3.050925, ..., -1.6768737,
 -1.6511288, 1.9876038],
 [0.39467925, 1.669555, 0.59880286, ..., 1.7896794,
 -2.3822052, 1.9330199],
 ...,
 [-3.259486, 2.728493, -3.6598694, ..., -0.78519315,
 -2.2183661, 0.13245803],
 [-0.92145747, -0.53599155, 1.6962891, ..., -0.33306703,
 -0.24070585, -0.43264183],
 [-1.3668569, 1.0633429, -0.39573893, ..., -0.5022214,
 -1.0951021, 0.8527349]], dtype=float32)

Search the web and Windows

21:21 21-10-2023

4) WhatsApp (no subject) - devadharshinik... Fake and real news dataset Fake_news_classification_Simple... +

kaggle.com/code/devadharshinik/fake-news-classification-simple-model/edit

Dialogflow YouTube Maps Gmail

Fake_news_classification_Simple_Mod... Draft saved

File Edit View Run Add-ons Help

Share Save Version 0

+ Draft Session (7m)

[15]:

```
X_train = np.stack(X_train)
X_test = np.stack(X_test)

X_train
```

[15]: array([[-0.23301356, 4.5374684, -2.3352568, ..., 1.769452,
 -3.347071, 3.5270379],
 [-0.31101125, 0.70727515, -3.050925, ..., -1.6768737,
 -1.6511288, 1.9876038],
 [0.39467925, 1.669555, 0.59880286, ..., 1.7896794,
 -2.3822052, 1.9330199],
 ...,
 [-3.259486, 2.728493, -3.6598694, ..., -0.78519315,
 -2.2183661, 0.13245803],
 [-0.92145747, -0.53599155, 1.6962891, ..., -0.33306703,
 -0.24070585, -0.43264183],
 [-1.3668569, 1.0633429, -0.39573893, ..., -0.5022214,
 -1.0951021, 0.8527349]], dtype=float32)

+ Code + Markdown

[16]:

```
LogReg = make_pipeline(
    MinMaxScaler(),
    LogisticRegression(max_iter=1000) ## Logistic_Regression_Classifier
```

Search the web and Windows

21:22 21-10-2023

The screenshot displays a Jupyter Notebook titled "Fake_news_classification_Simple_Mod...". The interface includes a top menu bar with "File", "Edit", "View", "Run", "Add-ons", and "Help". On the right, there are buttons for "Share", "Save Version", and a version count of "0". A left sidebar contains icons for file management and navigation. The main area shows two code cells. Cell [16] defines a pipeline:

```
LogReg = make_pipeline(
    MinMaxScaler(),
    LogisticRegression(max_iter=1000)  ## Logistic_Regression_Classifier
)
```

 Cell [17] defines a prediction function:

```
def predict(model):
    model.fit(X_train,y_train)
    print(classification_report(y_test, model.predict(X_test)))
```

 Below the code, the command `predict(LogReg)` has been executed, resulting in a classification report table. The table shows perfect performance (1.00) for all metrics (precision, recall, f1-score) across both classes (0 and 1), with a support of 4696 for class 0 and 4284 for class 1. The overall accuracy, macro average, and weighted average are all 1.00 with a support of 8980. At the bottom of the output, there are buttons for "+ Code" and "+ Markdown".

```
[16]:
LogReg = make_pipeline(
    MinMaxScaler(),
    LogisticRegression(max_iter=1000)  ## Logistic_Regression_Classifier
)

[17]:
def predict(model):
    model.fit(X_train,y_train)
    print(classification_report(y_test, model.predict(X_test)))

predict(LogReg)
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4696
1	1.00	1.00	1.00	4284
accuracy			1.00	8980
macro avg	1.00	1.00	1.00	8980
weighted avg	1.00	1.00	1.00	8980

+ Code + Markdown

❖ The dataset has been successfully modelled and executed.