Introduction

- **Aim**
  - **The main aim of this tutorial is to predict the gender of an author from his/her written text using Scikit-Learn Machine Learning toolkit.**
- **Task**
  - **Learn Input-Output Function**
  - **Given a Facebook comment (input) predict the gender of the author (output) who wrote it.**

Introduction

- Goal
  - The problem of gender prediction is treated as a supervised learning problem.
  - We need
    - Labelled data
    - High quality data
    - Large amount of data

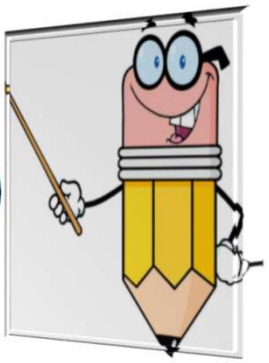
Input and Output

Input

- Facebook Comment - Plain text only
  - Represented as n-grams
  - Note that in this tutorial we will use word uni-grams (or 1-grams) as features to represent a Facebook comment (or text)

Output

- Gender of Author who wrote the comments
  - Represented as Gender attribute (Male/Female)



N-gram Models

- An n-gram is a contiguous sequence of n items from a given sample of text
- N-gram can be
  - Word based
  - Character based
- N represents the length of N-gram

Example – N-gram Generation from Input Text

- Input Text
  - R u coming?
- Word Uni-grams (N = 1)
  - Toeknized Text:
    - R
    - u
    - coming
    - ?
  - Set of Word Uni-grams = {R, u, coming, ?}

Example – N-gram Generation from Input Text

- Input Text
  - R u coming?
- Word Bi-grams (N = 2)
  - Toeknized Text:
    - R
    - u
    - coming
    - ?
  - Set of Word Bi-grams = {R u, u coming, coming ?}

Example – N-gram Generation from Input Text

- Input Text
  - R u coming?
- Word Tri-grams (N = 3)
  - Toeknized Text:
    - R
    - u
    - coming
    - ?
  - Set of Word Tri-grams = {R u coming, u coming ?}

Example – N-gram Generation from Input Text

- Input Text
  - R u coming?
- Character Tri-grams (N = 3)
  - Toeknized Text:
  - Note that space is also a character
    - R, ,u, ,c,o,m,l,n,g,?
  - Set of Character Tri-grams
    - {R u, u , co,com,omi,min,ing,gn?)

Three Phases of Machine Learning

Training	▪ Use subset of data (called Train data) to train model (learning)
Testing	▪ Use subset of data (called Test Data) to evaluate train model
Application	▪ Use your learned/trained models in real world applications

PHASE 1 & 2: TRAINING AND TESTING

Step 1: Import Libraries

Step 2: Read, Understand and Pre-process Train/Test Data

Step 2.1: Read Data

Step 2.2: Understand Data

Step 2.3: Pre-process Data

PHASE 1 & 2: TRAINING AND TESTING

Step 3: Label Encoding for Train/Test Data

Step 4: Feature Extraction – Changing Representation of Data
"from String to Vector"

Step 5: Train Machine Learning Algorithms using Training Data

Step 6: Evaluate Machine Learning Algorithms using Test Data

Step 7: Selection of Best Model

PHASE 3: Application Phase

Step 8: Application Phase

Step 8.1: Combine Data (Train + Test)

Step 8.2: Train Best Model (see Step 7) on all data(Train + Test)

Step 8.3: Save the Trained Model as Pickle File

PHASE 3: Application Phase

Step 9: Make prediction on unseen/new data

Step 9.1: Load the Trained Model (saved in Step 8.3)

Step 9.2: Take Input from User

PHASE 3: Application Phase

Step 9.3: Convert User Input into Feature Vector (Same as Feature Vector of Trained Model)

Step 9.4: Apply Trained Model on Feature Vector of Unseen Data and Output Prediction (Male/Female) to User

Step 1: Import Libraries

```
import re
import string
import scipy
import pickle
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import *
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import BernoulliNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import LinearSVC
from sklearn.metrics import accuracy_score
from prettytable import PrettyTable
from astropy.table import Table, Column
```

Step 2: Read, Understand and Pre-process Train/Test Data

Read, Understand and Pre-process Train/Test Data

Step 2.2: Understand Data

Train Dataset:

index	comment_text	gender
0	r u cmng or u not cmng	male
1	r you cmng	female
2	I am fine, r u fine	male
3	m fn and you	female
4	my frnd is gr8, wll dn.	male
5	my best friend is great	female

Step 2.2: Understand Data

```
Train Dataset Columns:

Index(['comment_text', 'gender'], dtype='object', name='index')


Number of instances in Train Dataset:

Train instances:  6
```

Step 2.2: Understand Data

Test Dataset:

index	comment_text	gender
0	plz go out, plz out with with frnd	male
1	r u going to walk, r u?	female
2	r you fine	male
3	are you fine	female

Step 2.2: Understand Data

```
Test Dataset Columns:

Index(['comment_text', 'gender'], dtype='object', name='index')


Number of instances in Test Dataset:

Test instances:  4
```

Step 2.2: Understand Data

Comments by 'Male' in Train Dataset:

index	comment_text	gender
0	r u cmng or u not cmng	male
2	I am fine, r u fine	male
4	my frnd is gr8, wll dn.	male

Step 2.2: Understand Data

Comments by 'Female' in Train Dataset:

index	comment_text	gender
1	r you cmng	female
3	m fn and you	female
5	my best friend is great	female

Step 2.2: Understand Data

Comments by 'Male' in Test Dataset:

index	comment_text	gender
0	plz go out, plz out with with frnd	male
2	r you fine	male

Step 2.2: Understand Data

Comments by 'Female' in Test Dataset:

index	comment_text	gender
1	r u going to walk, r u?	female
3	are you fine	female

Step 2.2: Understand Data

```
Words used by 'Male' in train data:

Words     Count

u           3
r           2
cmng        2
frnd        1
gr8,        1
wll         1
am          1
my          1
not         1
i           1
fine,       1
is          1
or          1
fine        1
dn.         1
dtype: int64
```

Step 2.2: Understand Data

```
Words used by 'Female' in train data:

Words    Count

you        2
fn         1
best       1
great      1
is         1
r          1
friend     1
cmng       1
m          1
and        1
my         1
dtype: int64
```

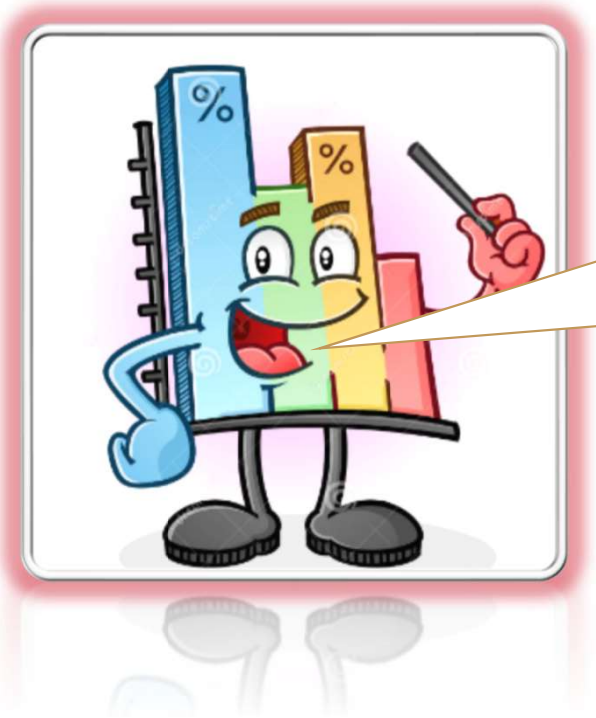
Step 2.2: Understand Data

```
Words used by 'Male' in test data:

Words    Count

plz      2
with     2
out,     1
go       1
r        1
frnd     1
out      1
fine     1
you      1
dtype: int64
```

Step 2.2: Understand Data

```
Words used by 'Female' in test data:

Words    Count

r        2
walk,    1
u?       1
u        1
you      1
are      1
fine     1
to       1
going    1
dtype: int64
```

Step 2.2: Understand Data



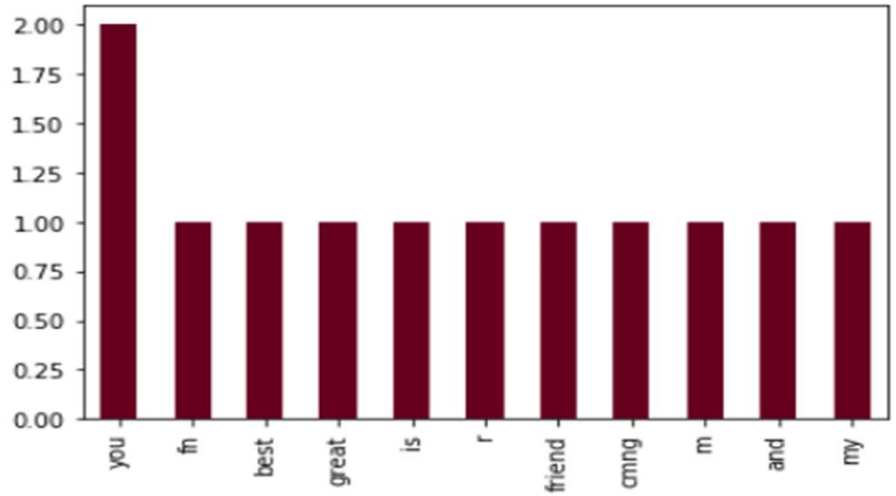
Understanding Data via

GRAPH is easy.

Let's Go!

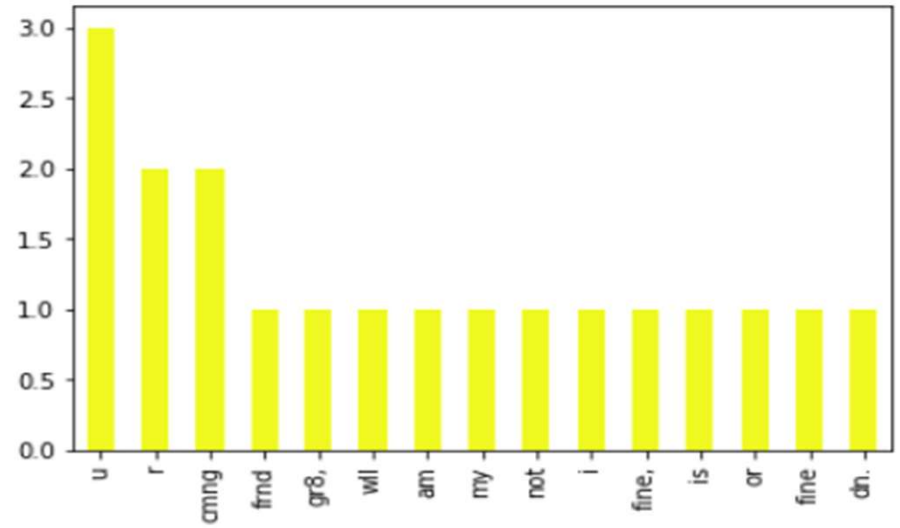
Step 2.2: Understand Data

```
Bar graph of words used by a female in train data:

<matplotlib.axes._subplots.AxesSubplot at 0xc348cf8>
```



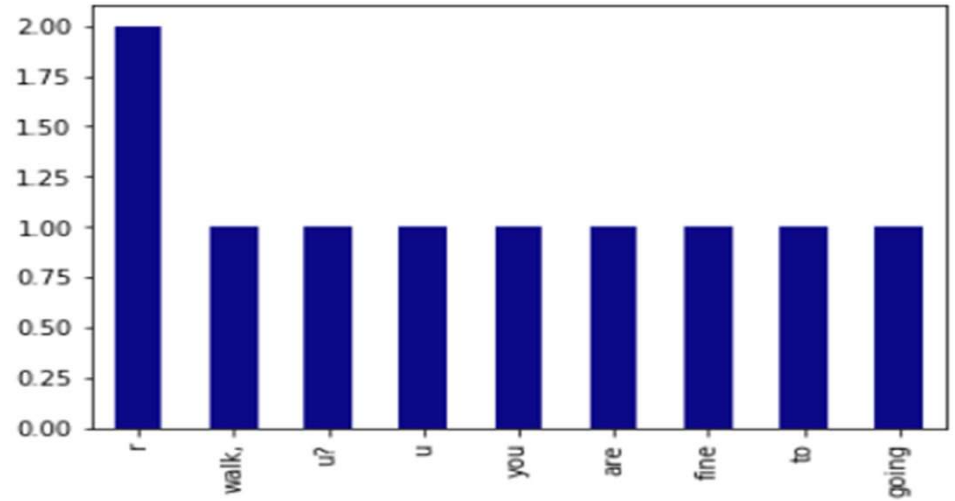
Step 2.2: Understand Data

```
Bar graph of words used by a male in train data:

<matplotlib.axes._subplots.AxesSubplot at 0xc3abef0>
```



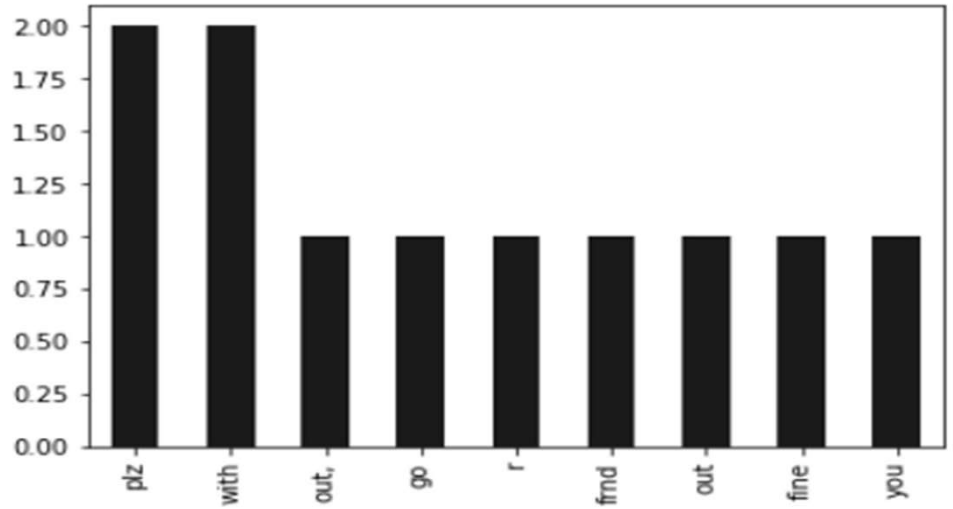
Step 2.2: Understand Data

```
Bar graph of words used by a female in test data:

<matplotlib.axes._subplots.AxesSubplot at 0xc3d8438>
```



Step 2.2: Understand Data

```
Bar graph of words used by a male in test data:

<matplotlib.axes._subplots.AxesSubplot at 0xc66c2b0>
```



Step 2.3: Pre-Process Data

Train dataset before pre-processing:

index	comment_text	gender
0	r u cmng or u not cmng	male
1	r you cmng	female
2	I am fine, r u fine	male
3	m fn and you	female
4	my frnd is gr8, wll dn.	male
5	my best friend is great	female

Step 2.3: Pre-Process Data

Test dataset before pre-processing:

index	comment_text	gender
0	plz go out, plz out with with frnd	male
1	r u going to walk, r u?	female
2	r you fine	male
3	are you fine	female

Step 2.3: Pre-Process Data

Train dataset after pre-processing:

index	comment_text	gender
0	r u cmng or u not cmng	male
1	r you cmng	female
2	i am fine r u fine	male
3	m fn and you	female
4	my frnd is gr wll dn	male
5	my best friend is great	female

Step 2.3: Pre-Process Data

Train dataset before pre-processing:

index	comment_text	gender
0	r u cmng or u not cmng	male
1	r you cmng	female
2	I am fine, r u fine	male
3	m fn and you	female
4	my frnd is gr8, wll dn.	male
5	my best friend is great	female

Train dataset after pre-processing:

index	comment_text	gender
0	r u cmng or u not cmng	male
1	r you cmng	female
2	i am fine r u fine	male
3	m fn and you	female
4	my frnd is gr wll dn	male
5	my best friend is great	female

Step 2.3: Pre-Process Data

Test dataset after pre-processing:

index		comment_text	gender
0	plz go out	plz out with with frnd	male
1		r u going to walk r u	female
2		r you fine	male
3		are you fine	female

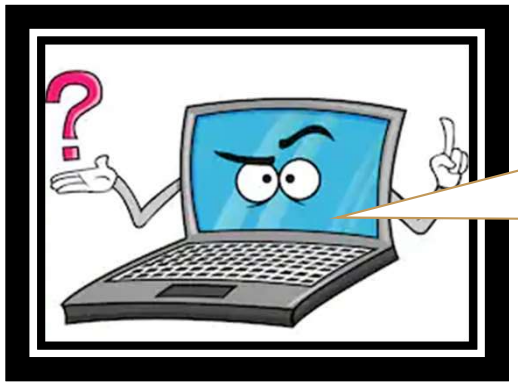
Step 2.3: Pre-Process Data

Test dataset before pre-processing:

index	comment_text	gender
0	plz go out, plz out with with frnd	male
1	r u going to walk, r u?	female
2	r you fine	male
3	are you fine	female

Test dataset after pre-processing:

index	comment_text	gender
0	plz go out plz out with with frnd	male
1	r u going to walk r u	female
2	r you fine	male
3	are you fine	female



Please convert data to a form that I can understand

Step 3: Label Encoding for Train/Test Data

Train Dataset Labels Encoding:

index	comment_text	gender	encoded_gender
0	r u cmng or u not cmng	male	1
1	r you cmng	female	0
2	i am fine r u fine	male	1
3	m fn and you	female	0
4	my frnd is gr wll dn	male	1
5	my best friend is great	female	0

Step 3: Label Encoding for Train/Test Data

Test Dataset Labels Encoding:

index		comment_text	gender	encoded_gender
0	plz go out	plz out with with frnd	male	1
1		r u going to walk r u	female	0
2		r you fine	male	1
3		are you fine	female	0

Step 4: Feature Extraction

```
vect = CountVectorizer(
                    strip_accents='unicode',
                    analyzer='word',
                    token_pattern=r'\w{1,}',
                    stop_words='english',
                    ngram_range=(1, 1),
                    max_features=10)

print("Parameters of TfidfVectorizer and its values:\n\n")
print(vect)
```

Step 4: Feature Extraction

```
Parameters of TfidfVectorizer and its values:


CountVectorizer(analyzer='word', binary=False, decode_error='strict',
        dtype=<class 'numpy.int64'>, encoding='utf-8', input='content',
        lowercase=True, max_df=1.0, max_features=10, min_df=1,
        ngram_range=(1, 1), preprocessor=None, stop_words='english',
        strip_accents='unicode', token_pattern='\\w{1,}', tokenizer=None,
        vocabulary=None)
```

Step 4: Feature Extraction

```
Check shape of the Features:


Train Features' Shape:  (6, 10)
Test Features' Shape:  (4, 10)
```

Step 4: Feature Extraction

Train Features Before Assigning a 'gender' Column:

index	best	cmng	dn	fine	fn	friend	frnd	plz	r	u
0	0	2	0	0	0	0	0	0	1	2
1	0	1	0	0	0	0	0	0	1	0
2	0	0	0	2	0	0	0	0	1	1
3	0	0	0	0	1	0	0	0	0	0
4	0	0	1	0	0	0	1	0	0	0
5	1	0	0	0	0	1	0	0	0	0

Step 4: Feature Extraction

Test Features Before Assigning a 'gender' Column:

index	best	cmng	dn	fine	fn	friend	frnd	plz	r	u
0	0	0	0	0	0	0	1	2	0	0
1	0	0	0	0	0	0	0	0	2	2
2	0	0	0	1	0	0	0	0	1	0
3	0	0	0	1	0	0	0	0	0	0

Step 4: Feature Extraction

Train Features after Assigning a 'gender' Column:

index	best	cmng	dn	fine	fn	friend	frnd	plz	r	u	gender
0	0	2	0	0	0	0	0	0	1	2	1
1	0	1	0	0	0	0	0	0	1	0	0
2	0	0	0	2	0	0	0	0	1	1	1
3	0	0	0	0	1	0	0	0	0	0	0
4	0	0	1	0	0	0	1	0	0	0	1
5	1	0	0	0	0	1	0	0	0	0	0

Step 4: Feature Extraction

Test Features after Assigning a 'gender' Column:

index	best	cmng	dn	fine	fn	friend	frnd	plz	r	u	gender
0	0	0	0	0	0	0	1	2	0	0	1
1	0	0	0	0	0	0	0	0	2	2	0
2	0	0	0	1	0	0	0	0	1	0	1
3	0	0	0	1	0	0	0	0	0	0	0

Step 4: Feature Extraction

```
Train Features' shape:  (6, 11)

Test Features' shape:  (4, 11)
```

Train Machine Learning Algorithms as I Am doing.

Step 5: Train ML Algorithms using Train Data

```
Parameters and their values:

LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
```

Step 5: Train ML Algorithms using Train Data

```
Parameters and their values:

RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False)
```

Step 5: Train ML Algorithms using Train Data

```
Parameters and their values:

LinearSVC(C=1.0, class_weight=None, dual=True, fit_intercept=True,
     intercept_scaling=1, loss='squared_hinge', max_iter=1000,
     multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
     verbose=0)
```

Step 5: Train ML Algorithms using Train Data

Parameters and their values:

```
BernoulliNB(alpha=1.0, binarize=0.0, class_prior=None, fit_prior=True)
```

Step 6: Evaluate ML Algorithms using Test Data

Before Prediciton using Logistic Regression:

index	comment_text	gender
0	plz go out plz out with with frnd	male
1	r u going to walk r u	female
2	r you fine	male
3	are you fine	female

Step 6: Evaluate ML Algorithms using Test Data

```
After Prediction using Logistic Regression:

index                          comment_text  gender predicted_gender
0       plz go out  plz out with with frnd    male             male
1                   r u going to walk  r u  female             male
2                             r you fine    male             male
3                          are you fine  female           female


Accuracy score =  0.75
```

Step 6: Evaluate ML Algorithms using Test Data

Before Prediciton using Random Forest Classifier:

index		comment_text	gender
0	plz go out	plz out with with frnd	male
1		r u going to walk r u	female
2		r you fine	male
3		are you fine	female

Step 6: Evaluate ML Algorithms using Test Data

After Prediction using Random Forest Classifier:

index	comment_text	gender	predicted_gender
0	plz go out plz out with with frnd	male	male
1	r u going to walk r u	female	female
2	r you fine	male	male
3	are you fine	female	female

Accuracy score = 1.0

Step 6: Evaluate ML Algorithms using Test Data

```
Before Prediciton using LinearSVC:

index                          comment_text  gender
0      plz go out  plz out with with frnd     male
1                  r u going to walk  r u   female
2                                r you fine    male
3                               are you fine  female
```

Step 6: Evaluate ML Algorithms using Test Data

After Prediction using LinearSVC:

index	comment_text	gender	predicted_gender
0 plz go out	plz out with with frnd	male	male
1	r u going to walk r u	female	male
2	r you fine	male	male
3	are you fine	female	female

Accuracy score = 0.75

Step 6: Evaluate ML Algorithms using Test Data

Before Prediciton using BernoulliNB:

index		comment_text	gender
0	plz go out	plz out with with frnd	male
1		r u going to walk r u	female
2		r you fine	male
3		are you fine	female

Step 6: Evaluate ML Algorithms using Test Data

After Prediction using BernoulliNB:

index	comment_text	gender	predicted_gender
0	plz go out plz out with with frnd	male	male
1	r u going to walk r u	female	male
2	r you fine	male	male
3	are you fine	female	female

Accuracy score = 0.75

Step 7: Selection of Best Model

```
Detailed Performance of all the models
======================================
+------------------------+----------+
|          Model         | Accuracy |
+------------------------+----------+
|   LogisticRegression   |   0.75   |
| RandomForestClassifier |   1.0    |
|       LinearSVC        |   0.75   |
|      BernoulliNB       |   0.75   |
+------------------------+----------+


Best Model.
======================================
+------------------------+----------+
|          Model         | Accuracy |
+------------------------+----------+
| RandomForestClassifier |   1.0    |
+------------------------+----------+
```

Step 8: Application Phase

PHASE 3: APPLICATION PHASE

Step 8.1: Combine Data (Train+Test)

All Features Dataframe with Labels/output:

index	best	cmng	dn	fine	fn	friend	frnd	plz	r	u	gender
0	0	2	0	0	0	0	0	0	1	2	1
1	0	1	0	0	0	0	0	0	1	0	0
2	0	0	0	2	0	0	0	0	1	1	1
3	0	0	0	0	1	0	0	0	0	0	0
4	0	0	1	0	0	0	1	0	0	0	1
5	1	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	1	2	0	0	1
1	0	0	0	0	0	0	0	0	2	2	0
2	0	0	0	1	0	0	0	0	1	0	1
3	0	0	0	1	0	0	0	0	0	0	0

Step 8.2: Train Best Model on All Data

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False)
```

Step 9.2: Take Input from User

```
Please enter your comment here: plz go out, plz out with with frnd
```

Step 9.3: Convert User Input into Feature Vector (Same ss Feature Vector of Trained Model)

```
Vector features:  ['best', 'cmng', 'dn', 'fine', 'fn', 'friend', 'frnd', 'plz', 'r', 'u']
```

```
User input features with weights

    best  cmng  dn  fine  fn  friend  frnd  plz  r  u
0     0     0   0     0   0       0     1    2  0  0
```

Step 9.4: Apply Trained Model on Feature Vector of Unseen Data and Output Prediction to User

```
Prediction: Male
```