PROMOTING LOW IMPACT LIFESTYLE

A classification challenge

Problem statement

- An active group on Reddit working towards a sustainable future by sharing of ideas and information to promote and enjoy a ‘Low Impact Lifestyle’.
- Started on Mar 30, 2013. Currently 8800 members and growing
- Concern of Moderators who feel that some posts are more in line with another subreddit Vegetarian (which has some common members)
- Such posts are more about Vegetarian way of life and not relevant to Low Impact Lifestyle
- Solution: Identify and redirect such posts and inform the members about changes

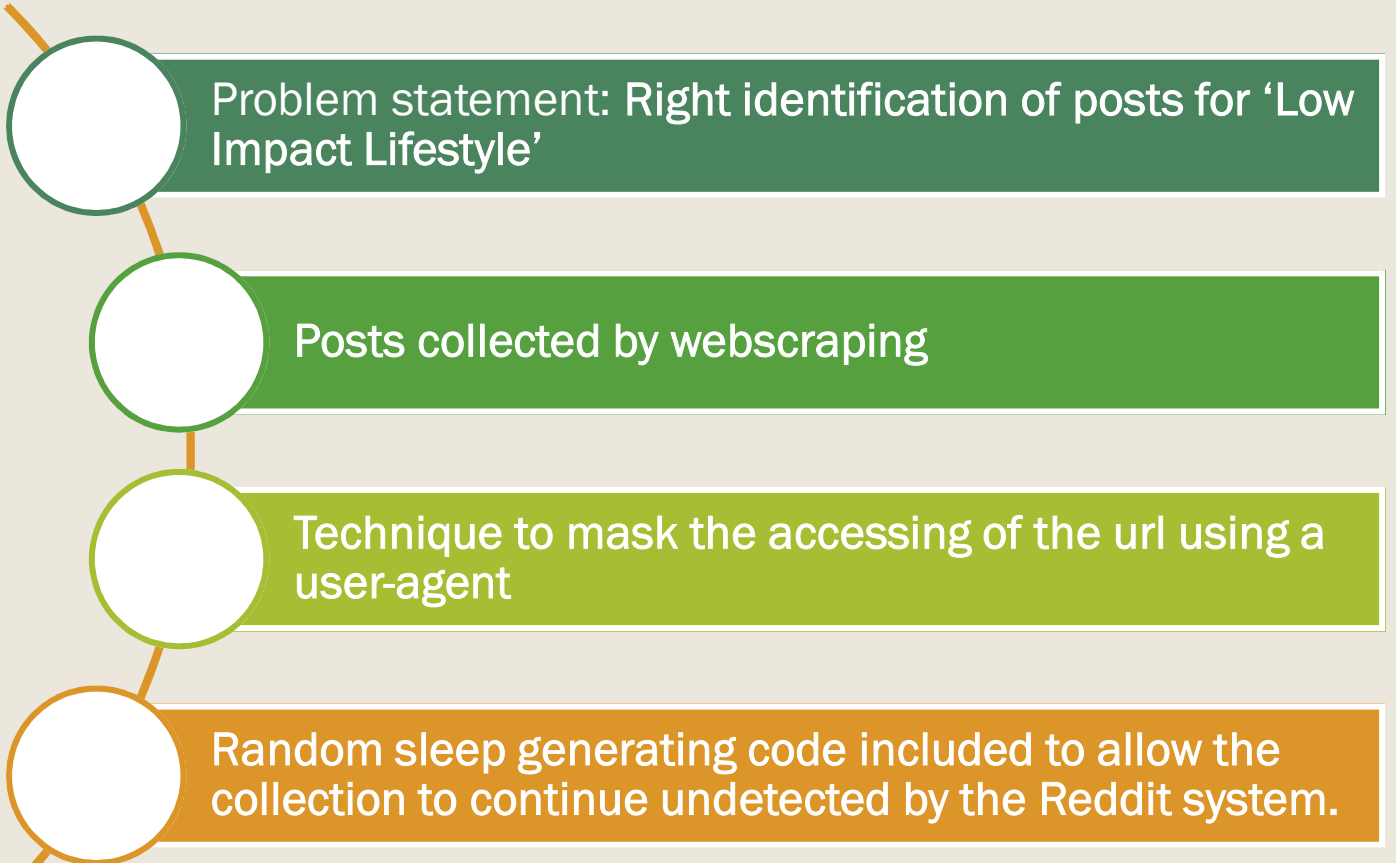
Process of developing model to classify posts

- Machine learning technique to be used for classification
- Since various classification techniques are available, 2 popular techniques to be used for the model building
- Text content needs to be vectorised before classification Model was selected based on fitting scores for training and validation data as well as metrics of the classification.
- Selected model was exposed to testing data and scores/metrics studied.
- It was also used to assess recent sample posts to demonstrate effectiveness

Details of the process

- Defining problem statement
- Collecting data from Subreddits – ‘Vegetarian’ and ‘Low Impact Lifestyle’
- Data Cleaning and Exploratory Data Analysis
- Preprocessing and Modelling
- Train-test-split of data for model design
- Applying Gridsearch to identify optimum parameters for Vectorizers and application of optimum parameters to Vectorizers
- Application of Classifiers and collection of metrics
- Evaluation of models and selection of model
- Conclusions and application of selected model

- Defining problem statement
- Collecting data from Subreddits
  - ‘Vegetarian’ and ‘Low Impact Lifestyle’



- Data Cleaning and Exploratory Data Analysis
- Preprocessing and Modelling

- Checking for duplicates, resolving null value issue and identifying columns to be used for modelling
- 20% of data kept aside for testing purposes (489 posts)
- Tokenisation, Lemmatisation and Removing of special characters and html links
- Tokenised data used for model building

	content	content_token	content_lemma
0	Roasted butternut squash Buddha Bowl	[Roasted, butternut, squash, Buddha, Bowl]	[Roasted, butternut, squash, Buddha, Bowl]
1	Entirely plant-based 'Chorizo' Paella!	[Entirely, plant, based, Chorizo, Paella]	[Entirely, plant, base, Chorizo, Paella]
2	100-year-old ballet teacher credits longevity to vegetarianism	[year, old, ballet, teacher, credits, longevity, to, vegetarianism]	[year, old, ballet, teacher, credit, longevity, to, vegetarianism]

- Train-test-split of data for model design
- Applying Gridsearch

- 2 vectorizers used – – **Count Vectorizer (CV) and Term Frequency-Inverse Document Frequency (TFIDF) Vectorizers were used.**
- Pipeline with Gridsearch and Logistic Regression used to identify optimum values of features of vectorizers
- Optimum values used in the identified Vectorizers to vectorize the tokens

```
pipe = Pipeline([
    ('cvec', CountVectorizer()),
    ('lr', LogisticRegression())
    ])

pipe_params = {
    'cvec__max_features': [2500, 3000, 3500],
    'cvec__min_df': [2, 3],
    'cvec__max_df': [.9, .95],
    'cvec__ngram_range': [(1,1), (1,2)]
}
gs = GridSearchCV(pipe, param_grid=pipe_params, cv=5)
gs.fit(X_train, y_train)
print(gs.best_score_)
gs.best_params_
```

```
{'cvec__max_df': 0.9,
 'cvec__max_features': 3500,
 'cvec__min_df': 2,
 'cvec__ngram_range': (1, 2)}
```

Count Vectorizer

Tfidf Vectorizer

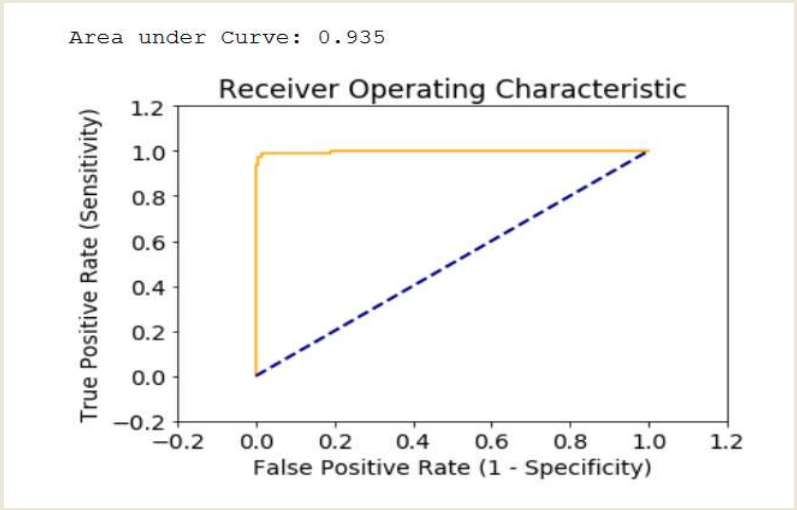
```
{'tvec__max_df': 0.9,
 'tvec__max_features': 3000,
 'tvec__min_df': 2,
 'tvec__ngram_range': (1, 1)}
```

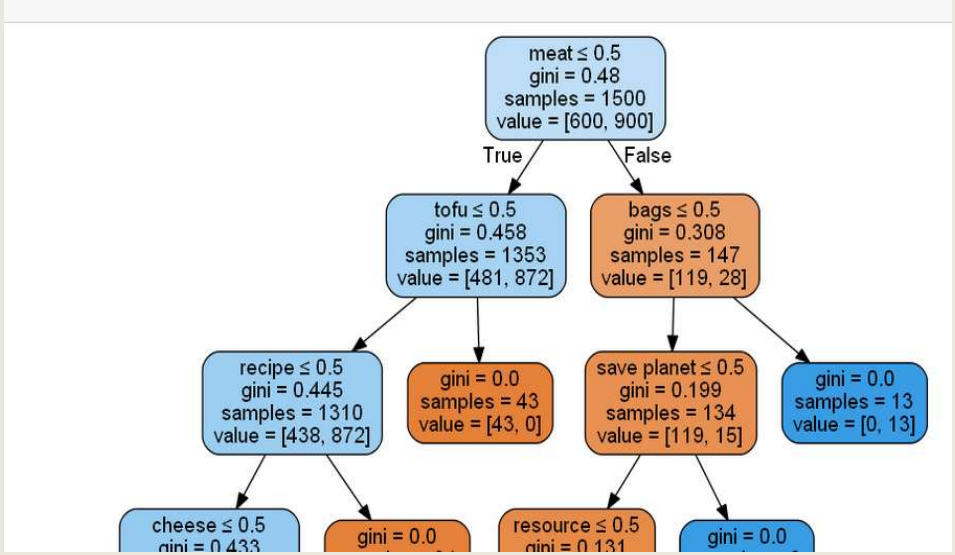
- Application of Classifiers
- Evaluation of models

- Naïve Bayes and Decision Tree Classifiers
- Naïve Bayes
  - *Classification algorithm suitable for binary and multiclass classification.*
  - *In supervised classification, training data are already labeled with a class.*
- Decision Tree
  - *Visually and explicitly represent decisions and decision making.*
  - *Uses a tree-like model of decisions.*
- Evaluation by comparing training and validation sets and comparing metrics of models.

Metrics and model selection

Model	CV_NB	TFIDF_NB	CV_DT	TFIDF_DT
Training score	0.9627	0.9873	0.8827	0.9600
Validation score	0.9300	0.9460	0.9660	0.8920
TN	188.0000	176.0000	183.0000	146.0000
FP	12.0000	24.0000	17.0000	54.0000
FN	23.0000	3.0000	0.0000	0.0000
TP	277.0000	297.0000	300.0000	300.0000
Accuracy	0.9300	0.9460	0.9660	0.8920
Sensitivity	0.9233	0.9900	1.0000	1.0000
Precision	0.9585	0.9252	0.9464	0.8475
f1 score	0.9406	0.9565	0.9724	0.9174





TFIDF Vectorizer and Naïve Bayes Classifier - model, scores and metrics

Model	TFIDF_NB	TFIDF_NB Test
Training score	0.9873	0.987300
Validation score	0.9460	0.946581
TN	176.0000	217.000000
FP	24.0000	23.000000
FN	3.0000	2.000000
TP	297.0000	226.000000
Accuracy	0.9460	0.946600
Sensitivity	0.9900	0.991200
Precision	0.9252	0.907600
f1 score	0.9565	0.947600

Success with testing with new posts:

```
def identify_subreddit(text, subreddit):

    val = tvec.transform(pd.Series(text))
    label=model_tfidf_nb.predict(val)
    print("Original subreddit:", subreddit)
    if label == 1:

        print("Text given is from subreddit Low Inpact Lifestyle")
    else:
        print("Text given is from subreddit Vegetarian")
```

```
Original subreddit: Vegetarian
Text given is from subreddit Vegetarian
```

```
Original subreddit: Low Impact Lifestyle
Text given is from subreddit Low Inpact Lifestyle
```

Questions?