Data&Data Challenge

Libraries used

I used common libraries used by data science community for small classification tasks.

1. **Scipy (**for numpy and pandas)
2. **Beautiful soup** (Just to remove html tags, this step can be skipped if you don't have it installed. Just comment the line in **text_processing** function)
3. **Scikit-learn** (for machine learning algorithms)

Approach

1. **Data Cleaning and Data wrangling :-**
    a. I started by converting text features into numeric ones e.g. "owner_type" and "INDEX new"
    b. For converting description or facebook posts I used Tf-Idf technique to make a feature vector and concatenated other features like likes, shares and owner_type to that vector for building models with multiple features.
    c. In data cleaning, I removed rows with nan description, duplicate rows where columns description, owner_type, likes and shares are same. After that I changed description column into bag of words removing html tags, pronunciation, stopwords, spaces, tabs and newline characters.
    d. After data cleaning step I removed rows where description string was empty.
    e. In this whole process I lost around 12,000 rows which was a big amount but I thought the above steps are important for making good features.
    f. After this I split the data into training and test set with 80:20 ratio.

2. **Classifiers Approach :-**
    a. Before building any model and doing any text processing on the text, I build a Random forest model on the data to do the preliminary analysis.
    b. After Data cleaning and Data wrangling step I trained many classifiers like Multinomial Naive Bayes, Support Vector Machines and Random forest and found out that Random forest performs the best with our data.
    c. Next step was to learn the parameters for Random forest. So I did a Grid search for many combinations of parameters through a pipeline using scikit-learn.
    d. Then I compared different metrics of the model like Precision, Recall and F1-score and created a confusion matrix of the model with best parameters learned before.
    e. In the next step to improve accuracy I went for the ensemble of different models and therefore I trained VotingClassifier model with the combination of three different classifiers (Multinomial Naive Bayes, Random forest and svm) with different weight for different classifiers.

f. Then I used more features apart from just description with the same models mainly owner_type, likes and shares to see if the accuracy can be improved. But the increase in accuracy was minimal.

3. **Results :-**
   a. I sticked with three main algorithms :- Naive Bayes, Svm and Random forest.
   b. Preliminary analysis without any text cleaning gave accuracy of **86%**.
   c. Since Random forest was the best performing classifier. Below is the confusion matrix of the classification

0 - 'OK', 1 - 'Reseller', 2 - 'Non'

Predicted	**0**	**1**	**2**	**All**
True				
0	1069	76	117	1262
1	89	846	248	1183
2	141	80	1807	2028
All	1299	1002	2172	4473

Table below shows different accuracy scores of different classifiers.

#	**Classifier**	**Features**	**Accuracy Score %**
1	Multinomial Naive Bayes	Post Description	79.83
2	Multinomial Naive Bayes	Description, likes, shares and owner_type	79.83
3	Svm	Post Description	82.03
4	Svm	Description, likes, shares and owner_type	82.36
5	Random Forest	Post Description	83.21
6	Random Forest	Description, likes, shares and owner_type	83.52
7	OneVsOneClassifier with Random forest	Post Description	83.39
8	OneVsOneClassifier with Random forest	Description, likes, shares and owner_type	83.70
9	Ensemble Voting Classifier	Post Description	83.28

Ideas/ Things that can be done to improve classification

1. Results shows that my solution does not improve the accuracy while adding more features. This can be because of one important reason.
    a. I am concatenating all features together in one vector.
    b. I think giving different weights to different features can improve accuracy.
2. Stemming is something I think can improve the robustness of classification while Data Cleaning.
3. I lost a lot of data points while doing data cleaning and with more data the classification can surely be improved because the data points were not equal in count for each label as there was variance. That is why from confusion matrix we can see that classification of label "**Non**" is better than the other two labels.
4. Deep Learning methods can most probably improve the classification as compared to supervised learning methods but they also need more data.