ITI – AIPRO Track

Machine Learning Intern.

Job title Classification by industry

(Multi-text Text Classification Task)

By / Mohammed AL-Sayed Agoor

LinkedIn: MOHAMMED AGOOR | LinkedIn

GitHub: AGOOR97 (Mohammed Agoor) (github.com)

E-mail: mohammedagoor1997@gmail.com

Youtube: Mohammed Agoor - YouTube

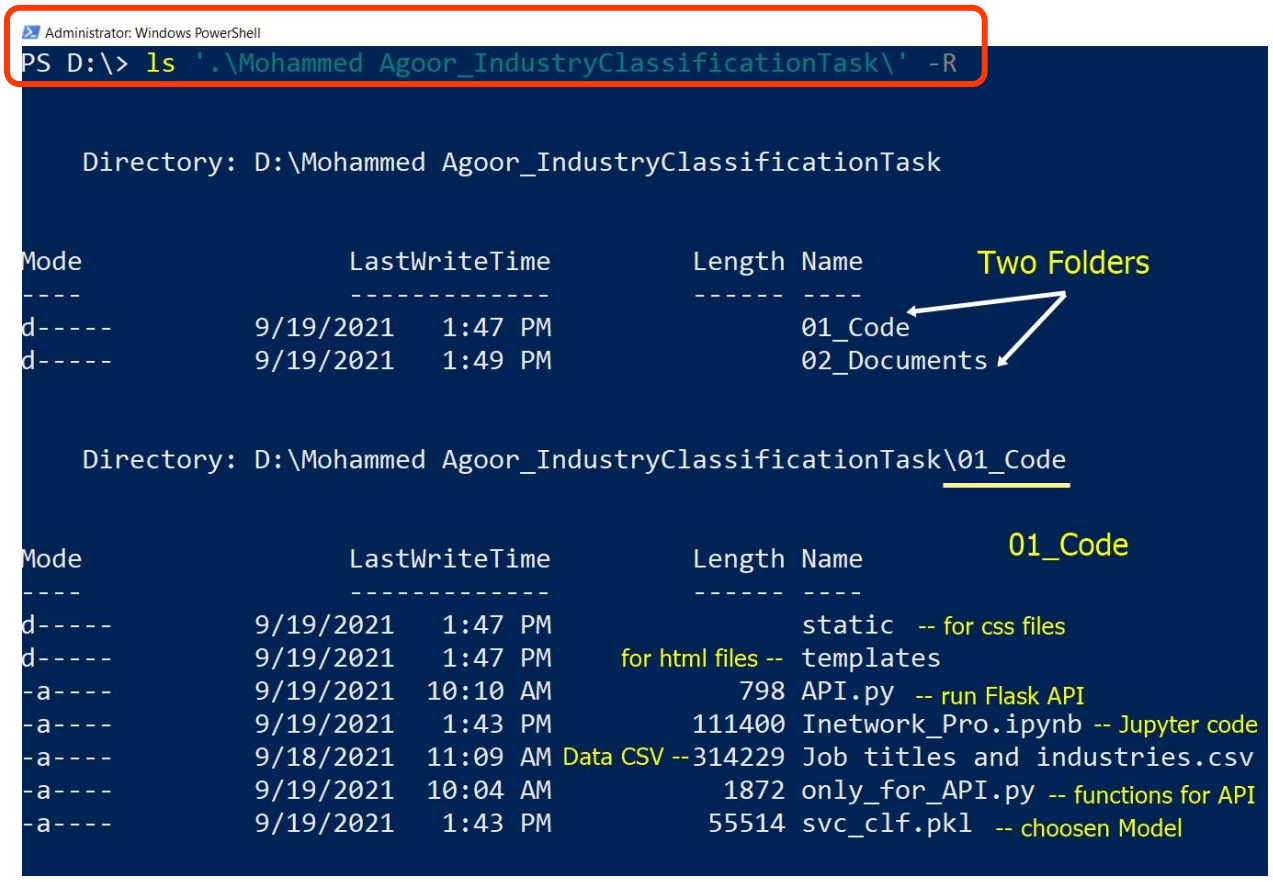
➢ Attached Files and Links:

- ✓ **My Video on General Steps to run API**

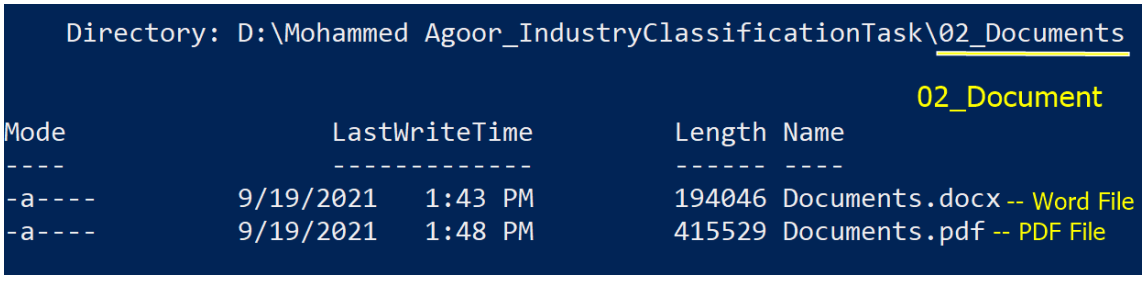
General Steps of my Project - YouTube

- ✓ **Files:**

01_Code:



02_Docuemnt:



1.1 Project Description:

You can think of the job industry as the category or general field in which you work. On a job application, "industry" refers to a broad category under which a number of job titles can fall. For example, sales is an industry; job titles under this category can include sales associate, sales manager, manufacturing sales rep, pharmaceutical sales and so on.

1.1.a Project Dataset:

Job titles and industries.csv - Google Drive

1.1.b Project Details:

You are given a dataset that has two variables (Job title & Industry) in a csv format of more than 8,500 samples.

This dataset is imbalanced (Imbalance means that the number of data points available for different classes is different) as follows:

IT 4746

Marketing 2031

Education 1435

Accountancy 374

1.2 Steps of My Code:

Check attached Jupyter NoteBook named ('Inetwork_Pro.ipynb')

- Import the required Libraries and Algorithms
- Loading the Data (attached CSV file)
- Looking at the Big Picture of this Data and try to make insights
- Check for Null Data (no nulls), decide to drop duplicates or not
- Before Preprocessing, split the Data to train & val and test sets
- EDA and some Preprocessing Steps including Vectorization
- Now we are ready to train Models
- But because of imbalanced Weights of each Class, I will give more weight to these class which are underrepresented (more details .. later)
- Starting by Naïve then Logistic then RandomForest then LinearSVC and ending with Xgboost Algorithms
- Try to Tuning the best One which is the LinearSVC
- Note; I choose LinearSVC not svm.SVC Classes (more details .. later)
- Ending of Tuning of LinearSVC and download this Model (.pckl)
- Evaluation on Test Set and Check out Accuracy (my choose Metric)
- Finally, I end with some Resources

1.3 Answer the Questions:

1. **Which techniques you have used while cleaning the data if you have cleaned it?**

- **I found no nulls in the Data, I think it is clear, but have much duplicates, so I decided to not drop these duplicates in this step and try to predict and get Accuracy if it is bad come back and remove duplicates, but in this step, I do not remove duplicates and the Accuracy is not bad.**
  But if I have much time I will come back, dropping duplicates and check what my Accuracy is.

---

2. **Why have you chosen this classifier?**

I choose (LinearSVC)

1. **The Highest Accuracy of these tried Algorithms**
2. **LinearSVC does not have much parameter to tune (I think only C need to be tuned), I mean LinearSVC not svm.SVC**
3. **LinearSVC is much faster than svm.SVC**

- ✓ **If I have much time I will do more Tuning using (skopt or Hpsklearn or RandomizedSearchCV)**
- ✓ **And try other Algorithms such as (SGDClassifier, VotingClassifier, other kernels of SVM).**

> With so many kernels to choose from, how can you decide which one to use? As a rule of thumb, you should always try the linear kernel first (remember that `LinearSVC` is much faster than `SVC(kernel="linear")`), especially if the training set is very large or if it has plenty of features. If the training set is not too large, you should try the Gaussian RBF kernel as well; it works well in most cases. Then if you have spare time and computing power, you can also experiment with a few other kernels using cross-validation and grid search, especially if there are kernels specialized for your training set's data structure.

Hands-on-ML

4. How do you deal with (Imbalance learning)?

- I give more Weights to the underrepresented Classes and give less wights to the overrepresented Classes
- I calculated these weights (check my Jupyter Notebook) by getting the Probability of each Class (we will reverse) these Probability so I subtract it from 1. And give these class_weights to the Algorithms as HyperParameter.

> If the training set was very skewed, with some classes being overrepresented and others underrepresented, it would be useful to set the `class_weight` argument when calling the `fit()` method, giving a larger weight to underrepresented classes, and a lower weight to overrepresented classes. These weights would be used by Keras when

Hands-on-ML

---

5. How can you extend the model to have better performance?

- I can extend the Model Performance as I said previously by try Tuning its HyperParameter using advanced techniques such as (Hpsklearn or Hyper-opt libraries using BayesianSearcCV)
- I can try more Algorithms such as (SGDClassifier, try other kernels in svm, VotingClassifier, using Ensemble Learning(ADaboost or Bagging or Pasting techniques))
- All these Steps need Time and Resources

---

6. How do you evaluate your model? (i.e. accuracy, F1 score, Recall)?

- I evaluate my Model using (Accuracy)
- I choose Accuracy, because it make sense here in multiclass
- Precision and Recall or even F1_score, I do not think none of them make sense to be the metric here although we have biased data, but they make sense much more in Binary Classification.

7. What are the limitations of your methodology or Where does your approach fail?

- ✓ After running this ,

```
(pd.DataFrame(np.c_[y_test, y_pred_test], columns=['actual', 'predicted']))
```

- In my Opinion:

  I found that IT may be misclassified with other Categories, I mean that IT is a side of error, the common is that every misclassified instance is containing IT whether it is actual or predicted, we can reduce error by giving a little bit more weight to class IT.

---

2. Resources:

Books

1. Hands-on-ML (by Aurilien Geron)
2. Deep Learning with Python (by Francois Chollet 'Author of Keras')

Links

- ✓ vectorize Troubleshoot

  scikit learn - Python Sklearn TfidfVectorizer Feature not matching; delete? - Data Science Stack Exchange

Flask API

- ✓ krishnaik06/Deployment-flask (github.com)
- ✓ Deploy Machine Learning Model Flask - YouTube
- ✓ Deploy Machine Learning Model using Flask - YouTube
