Statistical Office
in Poznań

Machine Learning
Coding & Classification

ECOICOP classification

Code and data

Krystyna Piątkowska

Marta Kruczek-Szepel, Claude Julien (UNECE)

16.11.2020 Poznań

Agenda

1. Learning and training – statswiki UNECE:
    1.1. ECOICOP dataset
    1.2. Machine Learning tutorial

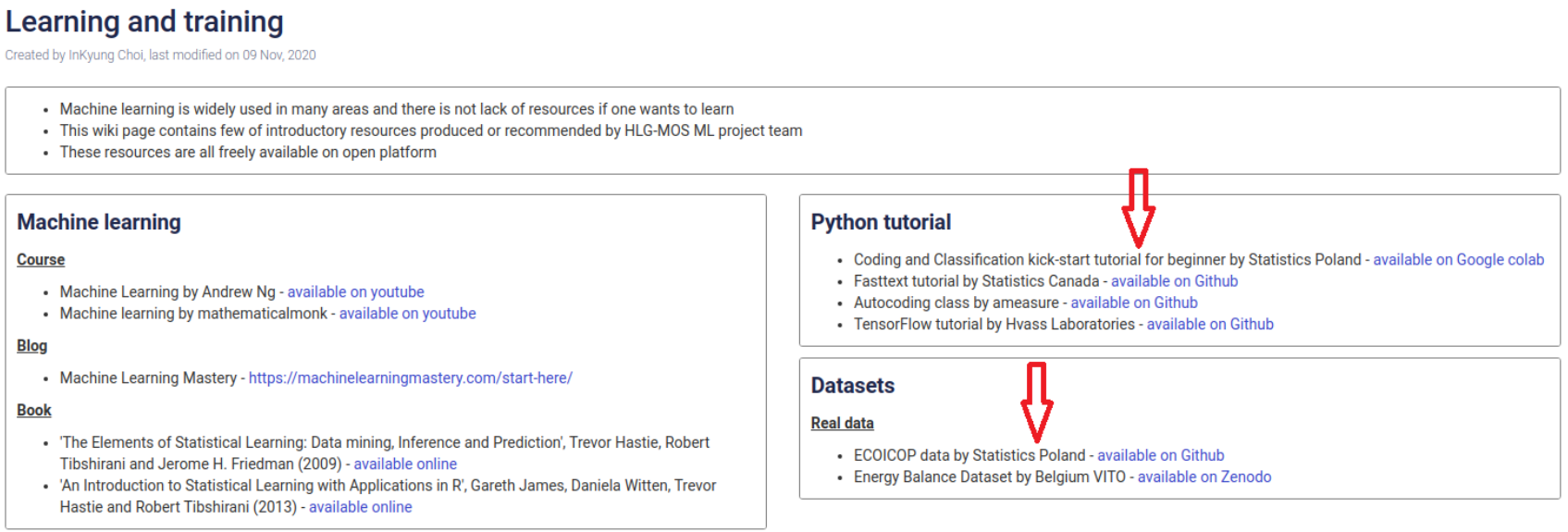
2. Studies and Code – statswiki UNECE:
   Github files
    2.1. Hyperparameter tunning „for” loop – input/output
    2.2. Hyperparameter tunning GridsearchCV - input/output
    2.3. Best parameters & results

3. User's experiences with the ML code and data shared

1. Learning and training – statswiki UNECE

1.1. ECOICOP dataset

1.2. ML tutorial



Learning and training

Created by InKyung Choi, last modified on 09 Nov, 2020

- Machine learning is widely used in many areas and there is not lack of resources if one wants to learn
- This wiki page contains few of introductory resources produced or recommended by HLG-MOS ML project team
- These resources are all freely available on open platform

Machine learning

Course

- Machine Learning by Andrew Ng - available on youtube
- Machine learning by mathematicalmonk - available on youtube

Blog

- Machine Learning Mastery - https://machinelearningmastery.com/start-here/

Book

- 'The Elements of Statistical Learning: Data mining, Inference and Prediction', Trevor Hastie, Robert Tibshirani and Jerome H. Friedman (2009) - available online
- 'An Introduction to Statistical Learning with Applications in R', Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani (2013) - available online

Python tutorial

- Coding and Classification kick-start tutorial for beginner by Statistics Poland - available on Google colab
- Fasttext tutorial by Statistics Canada - available on Github
- Autocoding class by ameasure - available on Github
- TensorFlow tutorial by Hvass Laboratories - available on Github

Datasets

Real data

- ECOICOP data by Statistics Poland - available on Github
- Energy Balance Dataset by Belgium VITO - available on Zenodo

Link statswiki:

https://statswiki.unece.org/display/ML/Learning+and+training

2. Studies and Code – statswiki UNECE:

Statswiki link:

https://statswiki.unece.org/pages/viewpage.action?spaceKey=ML&title=Studies+and+Codes

6	Coding & Classification	Production description to ECOICOP	Poland	Web scraping data	Naïve bayes, Logistic regression, Random forest, Support vector machine, Neural network	**Yes** (Click Github link)	Python

Github source code:

https://github.com/statisticspoland/ecoicop_classification

2.1. Github Files
Hyperparameter tunning „for” loop – input/output

- Linear_SVC
- Logistic_Regression
- Naive_Bayes
- Random_Forest

logistic_regression.py

random_forest.py

INPUT

	A	B
1	product	category
2	"słynne roślinne"	Margaryna i inne tłuszcze roślinne
3	#Hejki - Emotki lizaki ręcznie robione o smakach owocowych	Wyroby cukiernicze
4	100% Pur jus d'orange - sok pomarańczowy z miąższe...	Soki owocowe i warzywne
5	100% sukraloza bez cukru (substancje słodzące)	Sztuczne substytuty cukru
6	100% z brzoskwiń produkt owocowy słodzony zag.sokiem winogronowym	Dżemy, marmolady i miód
7	100% z czarnych porzeczek produkt owocowy słodzony zag.sokiem winogro	Dżemy, marmolady i miód

Split to train, validation, test dataset

vectorizer = CountVectorizer()
vectorizer.fit() vectorizer.transform()

```
for c in [0.1, 1, 2, 3]:
    for fit_intercept in [True, False]:
        for class_weight in [None, 'balanced']:
            for solver in ["newton-cg", "lbfgs", "liblinear", "sag", "saga"]:
                for multi_class in ["ovr", "multinomial"]:
```

OUTPUT

	C	fit_intercept	class_weight	solver	multi_class	max_iter	val_accuracy	train_accuracy	f1_score_micro
0	0.1	TRUE		newton-c	ovr	200	0.8200	0.8566	0.8200
1	0.1	TRUE		newton-c	multinomial	200	0.8253	0.8667	0.8253
2	0.1	TRUE		lbfgs	ovr	200	0.8200	0.8566	0.8200
3	0.1	TRUE		lbfgs	multinomial	200	0.8253	0.8667	0.8253
4	0.1	TRUE		liblinear	ovr	200	0.8327	0.8679	0.8327
5	0.1	TRUE		sag	ovr	200	0.8200	0.8566	0.8200

2.2. Github Files
Hyperparameter tunning GridSearchCV – input/output

- Linear_SVC
- Logistic_Regression
- Naive_Bayes
- Random_Forest

naive_bayes_params_tunn.py

linear_SVC_params_tunn.py

INPUT

	A	B
1	product	category
2	"słynne roślinne"	Margaryna i inne tłuszcze roślinne
3	#Hejki - Emotki lizaki ręcznie robione o smakach owocowych	Wyroby cukiernicze
4	100% Pur jus d'orange - sok pomarańczowy z miąższe...	Soki owocowe i warzywne
5	100% sukraloza bez cukru (substancje słodzące)	Sztuczne substytuty cukru
6	100% z brzoskwiń produkt owocowy słodzony zag.sokiem winogronowym	Dżemy, marmolady i miód
7	100% z czarnych porzeczek produkt owocowy słodzony zag.sokiem winogro	Dżemy, marmolady i miód

Split to train, validation, test dataset

```
linSVC_pipeline = Pipeline([
        ('vect', TfidfVectorizer(max_df=0.1, ngram_range=(1, 2), stop_words=stop_words_list, sublinear_tf=True, \
                token_pattern='\w\w+|[1-9]\.[1-9]\%|[1-9]\,[1-9]\%|[1-9]\.[1-9]|[1-9]\,[1-9]|[1-9]\%')),
        ('tfidf', TfidfTransformer(norm='l2', smooth_idf=True, sublinear_tf=False, use_idf=True)),
        ('clfLin', svm.LinearSVC(dual=False, max_iter=1200)),
    ])
linSVC_pipeline.fit(X, y)
```

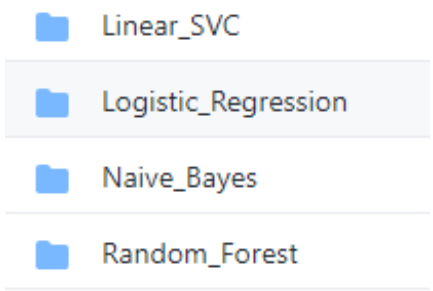
```
grid_params = {
        'clfLin__penalty': ('l1', 'l2'),
        'clfLin__multi_class': ('ovr', 'crammer_singer'),
        'clfLin__C': (0.01, 0.1, 1, 10, 100, 1000),
        'clfLin__loss': ('hinge', 'squared_hinge'),
    }
```

```
gs = GridSearchCV(linSVC_pipeline, grid_params, cv=5, n_jobs=-1,
verbose=1, error_score=0, scoring= ‚accuracy')
gs = gs.fit(X, y)
```

OUTPUT

	A	B	C	D	E
1		clfLin__C	clfLin__loss	clfLin__multi_class	clfLin__penalty
2	0	1	hinge	crammer_singer	l1

2.3. Github Files – best parameters & results



linear_SVC_results.py

logistic_regression_results.py

naive_bayes_results.py

random_forest_results.py

https://colab.research.google.com/drive/1_XqJxRYZ588gaq5geqjzWYPhVr67JS5e?usp=sharing#scrollTo=MfTY-ogAEnDF

3. User's experiences with the ML code and data shared

- I have a background in statistics/methodology and had very little knowledge of ML
- I was introduced to ML by the work of the project members
- With the ML code/data shared and assistance of ML team members, I was quickly able to experiment
- I share my experiences using ML to:
  - Identify misclassified products on the shared data
  - Use this imperfect dataset to simulate the integration of ML into a manual coding operation
  - Make many mistakes and learn many lessons along the way
- My experiences will continue and more may be added