COVID-19 SYMPTOM IDENTIFICATION

A Supervised Learning approach

IART - MIEIC@FEUP 2021/2022
Group 25
Ana Teresa Cruz - up201806460@up.pt
André Nascimento - up201806461@up.pt
António Bezerra - up201806854@up.pt

Specification

In this project, we will analyse a dataset containing information about patient's symptoms and classify them as **COVID-19**, **flu**, **cold** and **allergy** cases.

The dataset is available at:
https://www.kaggle.com/walterconway/covid-flu-cold-symptoms

# COUGH	# MUSCLE_...	# TIREDNESS	# SORE_THR...	# RUNNY_N...	# STUFFY_N...	# FEVER	# NAUSEA	# VOMITING	# DIARRHEA	A TYPE
0	0	1	0	1	0	0	0	0	0	ALLERGY
0	0	1	0	0	0	0	0	0	0	ALLERGY
0	1	1	1	0	0	0	0	0	0	ALLERGY
0	0	0	1	1	0	0	0	0	0	ALLERGY
0	0	1	0	1	0	0	0	0	0	ALLERGY
0	0	0	0	0	0	0	0	0	0	ALLERGY
1	0	0	0	1	1	0	0	0	0	ALLERGY
0	1	1	1	0	0	0	0	0	0	ALLERGY
1	1	0	0	1	0	0	0	0	0	ALLERGY
1	0	1	1	1	0	0	0	0	0	ALLERGY

Related Work

The dataset in analysis was based on medical data provided by the Mayo Clinic:
https://www.mayoclinic.org/diseases-conditions/coronavirus/in-depth/covid-19-cold-flu-and-allergies-differences/art-20503981

The dataset was automatically generated using this algorithm:
https://github.com/WalterConway/SymptomGenerator

We used resampling techniques found in this guide:
https://beckernick.github.io/oversampling-modeling/

To learn more about the algorithm's implementation we consulted SciKit Learn's documentation:
https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier
https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

Approach

Our approach was essentially comprised by three steps:

1. **Data analysis:** we explored the raw dataset to identify missing or wrong information and also to decide how to better use it and what problems it might have.
2. **Algorithm implementation:** we used SciKit Learn's algorithm implementations to obtain experimental results of the classification.
3. **Evaluation and refinement:** after initial results are obtained, we combined that information with our knowledge of the dataset to both tune the algorithms and devise new strategies for the classification, such as using resampling.

Tools and Algorithms

- **Tools**
  - Python 3.8
  - Jupyter Notebook
  - Pandas
  - Numpy
  - MatPlotLib
  - Seaborn
  - SciKit Learn
  - Imbalanced Learn
- **Algorithms implemented**
  - Decision Tree
  - Nearest neighbor
  - Support Vector Machines
  - Neural Networks

All of these tools are installed with Anaconda, with the exception of the Imbalanced Learn library. This should be installed with:

```
conda install -c conda-forge imbalanced-learn
```

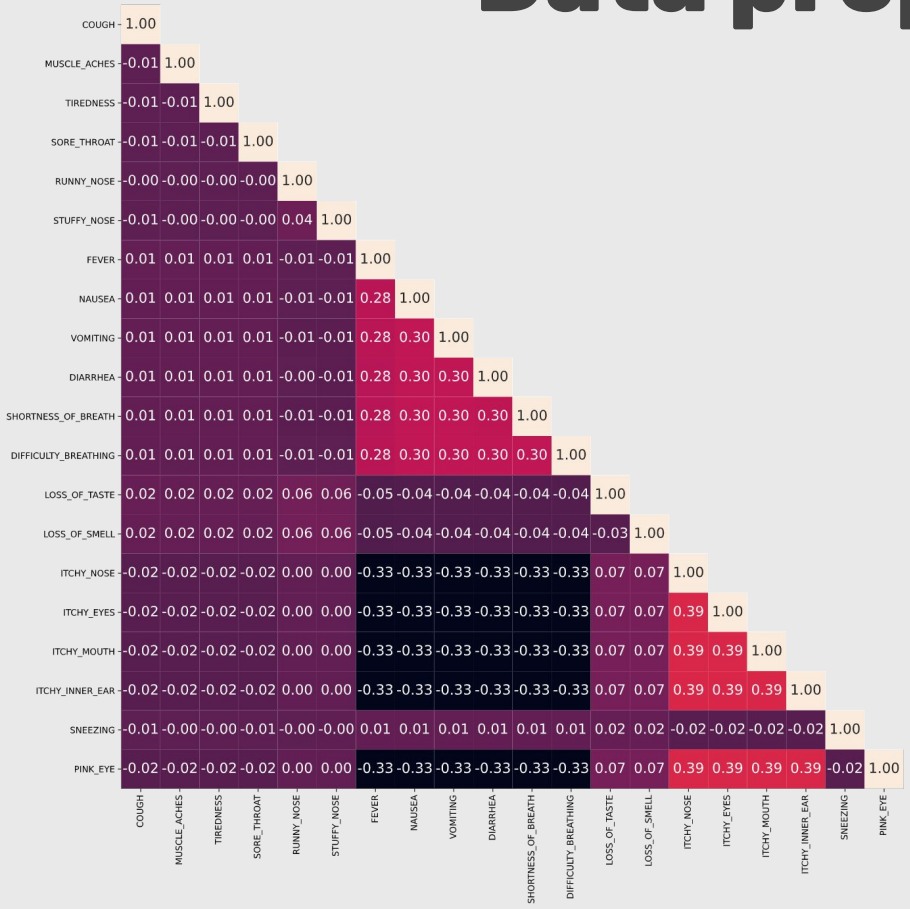
or

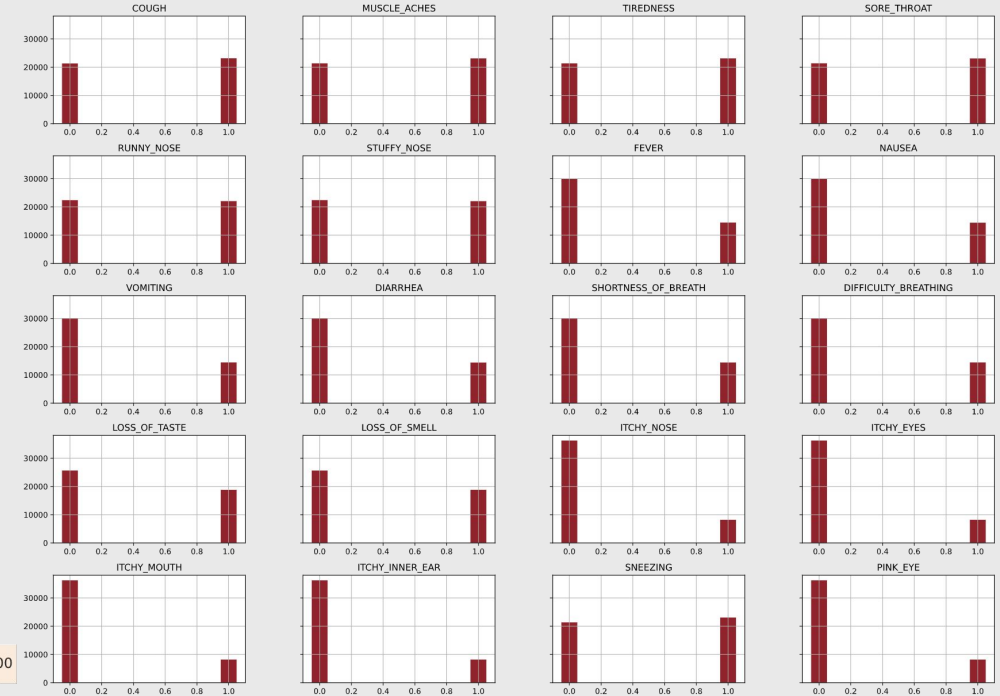
```
pip install -U imbalanced-learn
```

Implementation

- Data processing
- Data analysis using visualization
- Algorithm implementation
- Parameter tuning
- Result analysis

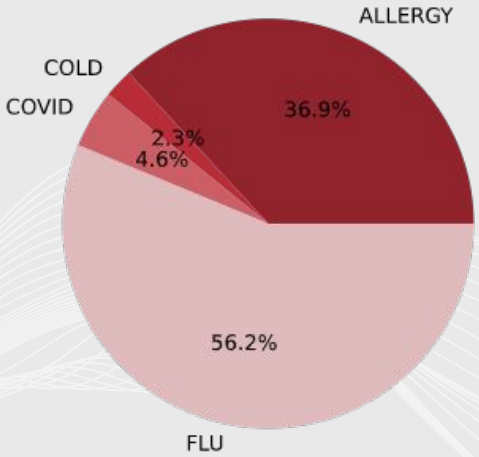
Data preprocessing





Resampling

What we quickly observed was that the class distribution was extremely unbalanced. Over half of the cases were Flue cases, and only 2.3% were Cold cases.



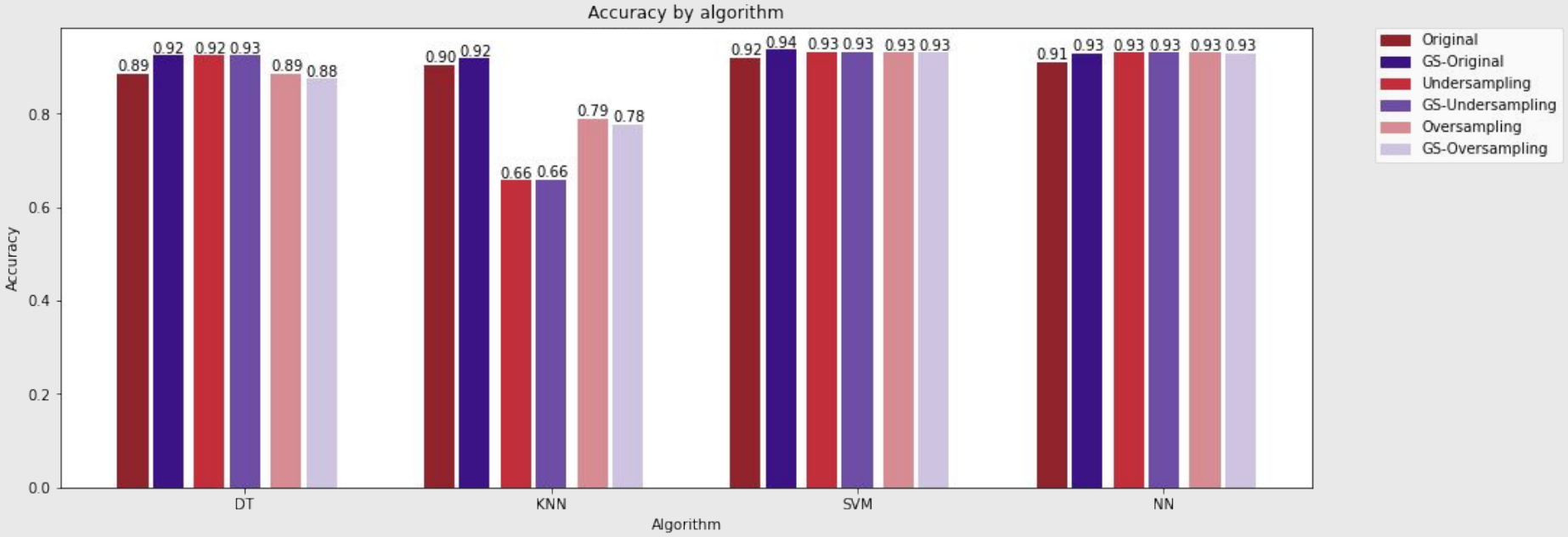

```
from imblearn.under_sampling import RandomUnderSampler

rus = RandomUnderSampler()

us_inputs, us_labels = rus.fit_resample(train_in, train_classes)

print(Counter(us_labels))
```

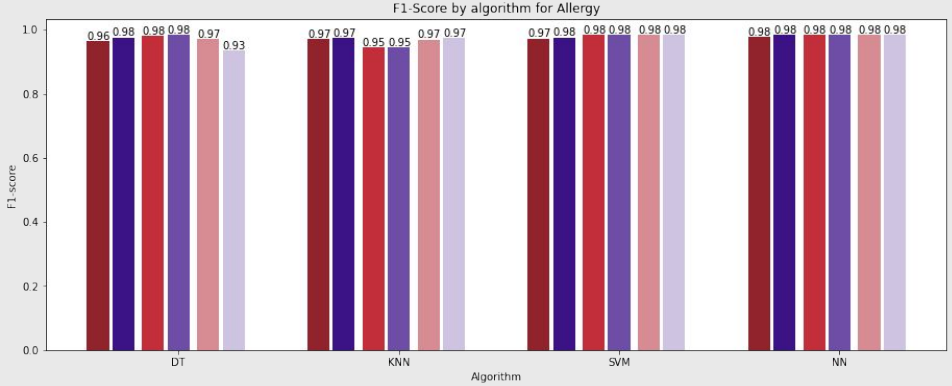
```
Counter({'ALLERGY': 768, 'COLD': 768, 'COVID': 768, 'FLU': 768})
```

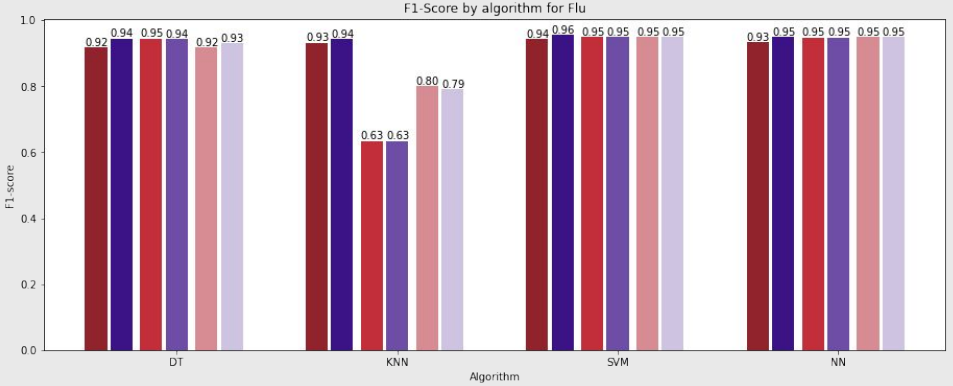
```
from imblearn.over_sampling import SMOTE

ros = SMOTE()

os_inputs, os_labels = ros.fit_resample(train_in, train_classes)

print(Counter(os_labels))
```

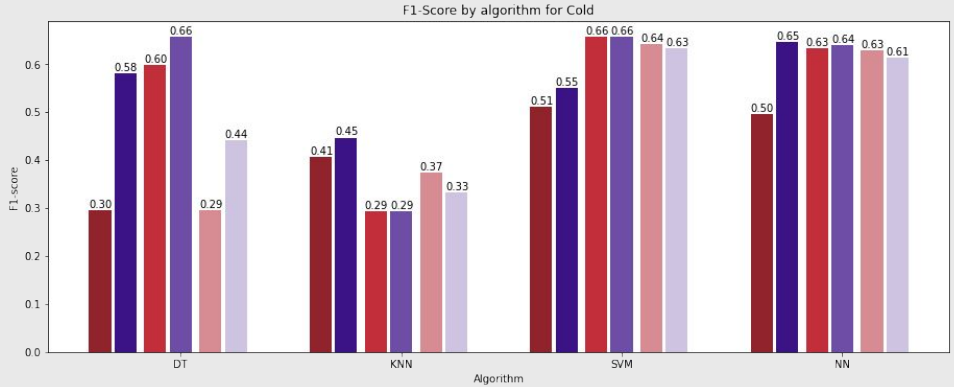
```
Counter({'ALLERGY': 18750, 'FLU': 18750, 'COLD': 18750, 'COVID': 18750})
```

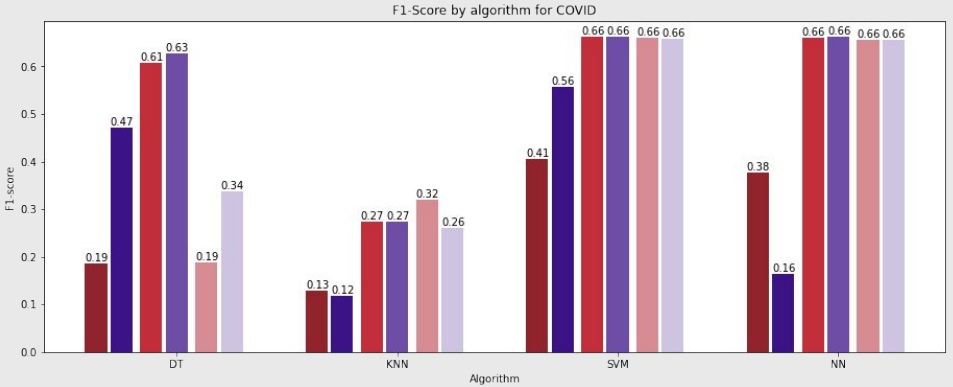
Result Comparison

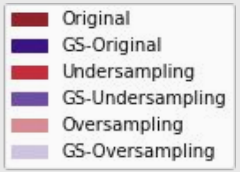



F1-Score by algorithm for Allergy
F1-score
Algorithm
DT
KNN
SVM
NN
0.96 0.98 0.98 0.98 0.97 0.93
0.97 0.97 0.95 0.95 0.97 0.97
0.97 0.98 0.98 0.98 0.98 0.98
0.98 0.98 0.98 0.98 0.98 0.98


F1-Score by algorithm for Flu
F1-score
Algorithm
DT
KNN
SVM
NN
0.92 0.94 0.95 0.94 0.92 0.93
0.93 0.94 0.63 0.63 0.80 0.79
0.94 0.96 0.95 0.95 0.95 0.95
0.93 0.95 0.95 0.95 0.95 0.95


F1-Score by algorithm for Cold
F1-score
Algorithm
DT
KNN
SVM
NN
0.30 0.58 0.60 0.66 0.29 0.44
0.41 0.45 0.29 0.29 0.37 0.33
0.51 0.55 0.66 0.66 0.64 0.63
0.50 0.65 0.63 0.64 0.63 0.61


F1-Score by algorithm for COVID
F1-score
Algorithm
DT
KNN
SVM
NN
0.19 0.47 0.61 0.63 0.19 0.34
0.13 0.12 0.27 0.27 0.32 0.26
0.41 0.56 0.66 0.66 0.66 0.66
0.38 0.16 0.66 0.66 0.66 0.66


Original
GS-Original
Undersampling
GS-Undersampling
Oversampling
GS-Oversampling

Conclusions

+90% accuracy

Good overall accuracy without much tuning.

Unbalanced data

The class distribution affected the result quality, attenuated by resampling

SVC with Undersampling

Best algorithm and training set, 93% accuracy and best F1-score

Overall...

We are happy with the results and were able to explore various concepts in AI.