bagidata

Data Scientist Test

Submitted by Nikko P. Gunara

February 25th, 2022

Overview

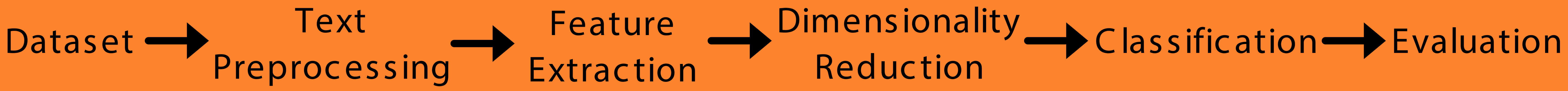
General Framework

News Title Classification

Spam Comment Classification

General Framework*

*details in Jupyter Notebook.



Dataset → Text Preprocessing → Feature Extraction → Dimensionality Reduction → Classification → Evaluation

Preprocessing

Why?

Most text and document data sets contain many unnecessary words such as stopwords, misspelling, slang, etc.

How?

Text preprocessing conducted in this projects includes lowering, cleaning (weird chars, links, numbers, etc.), tokenization, stopwords removal (using nltk English stopwords), spell correction, and finally, lemmatization.

Feature Extraction

Why?

In general, texts and documents are unstructured data sets. So, these data must be converted into structured feature space (numbers).

How?

For News Title Classification, we used Gensim's pre-trained Word2Vec Google News model that has been trained on about 100 billion words. While for Spam Comment Classification, we used Gensim's pre-trained GloVe Twitter model that has been trained on about 2 billion tweets.

Dimensionality Reduction

Why?

features extracted from could yield up to 300 dimensions (even thousands) for each title. To save computation time and visualize, it's common to reduce it to fewer dimensions.

How?

by conducting Principal Component Analysis (PCA).
The "elbow method" is commonly used to choose the appropriate number of components for PCA. But in this case, n_components is tuned in the hyperparameter optimization process.

Classification & Hyperparameter Tuning

Why?

To find an optimal model for a specific task, we need to tune its hyperparameter (HP) while training it.

How?

Hyperparameter optimization is conducted by using a searching algorithm. Here, we used the Bayesian Optimization from skopt. This algorithm is one of the simplest methods besides Grid search (GS) and Randomized search (RS). It optimizes HP with consideration of previous results, while GS and RS doesn't.

Evaluation

Why?

Since we train and test using a few types of classification algorithms, we need a metric to compare which one is the best.

How?

For these 2 cases, we can use Accuracy and F1score. F1score is calculated by considering both Precisions and Recall so we can use it for an imbalanced dataset. While if we use Accuracy, our result may be biased by the category with most data.

News Title Classification

Feature Extraction

Gensim's pre-trained Word2Vec model that has been trained on Google News with about 100 billion words.

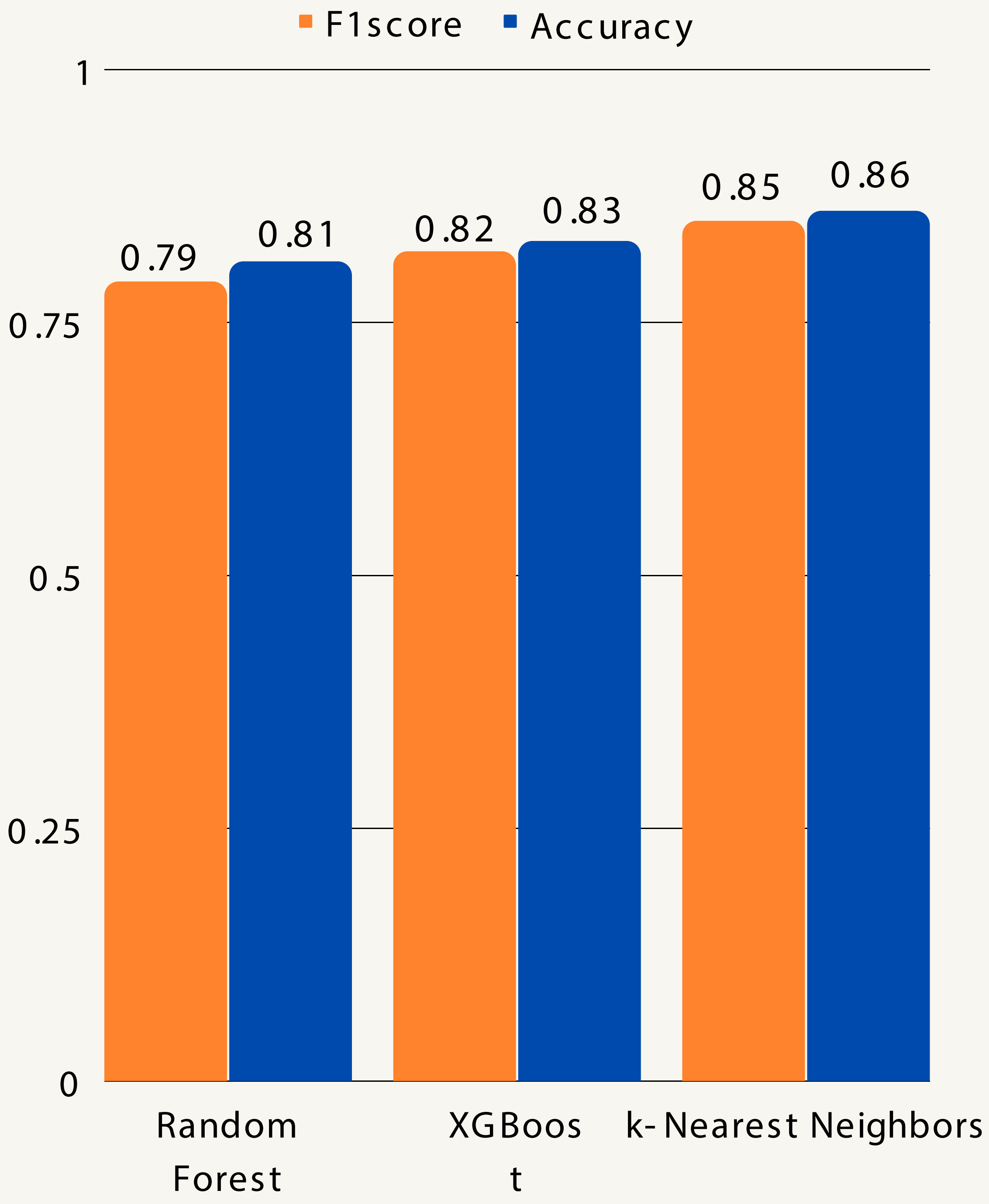
Classification & Hyperparameter Optimization

Using stratified k-fold cross validation because the dataset is imbalanced for each category.

Evaluation

Using F1score as the main performance metric to compare and evaluate models.

Results


■ F1score
■ Accuracy
1
0.75
0.5
0.25
0
0.79
0.81
0.82
0.83
0.85
0.86
Random Forest
XGBoost
k-Nearest Neighbors

Spam Comment Classification

Feature Extraction

Gensim's pre-trained GloVe model that has been trained with about 2 billion tweets.

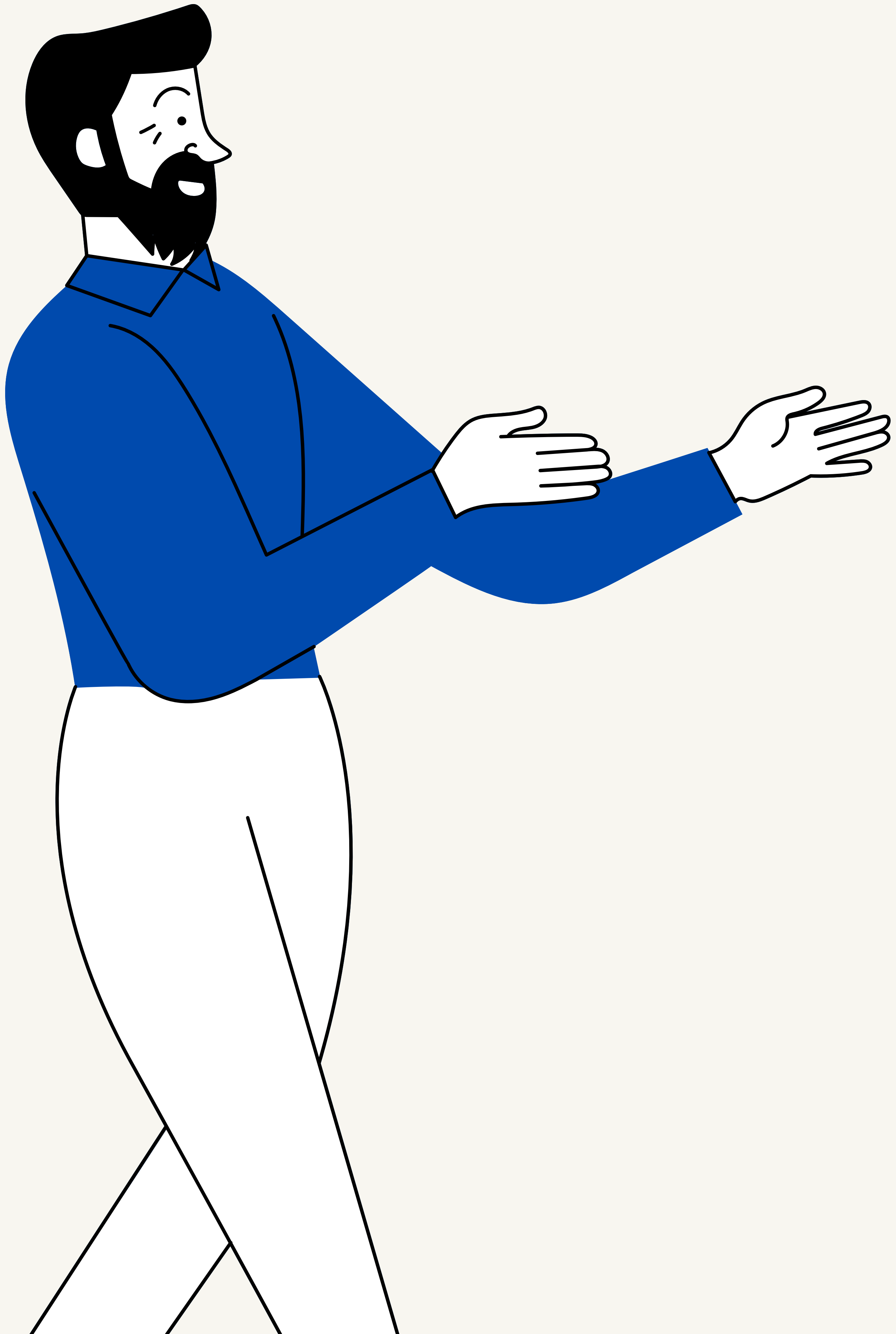
Classification & Hyperparameter Optimization

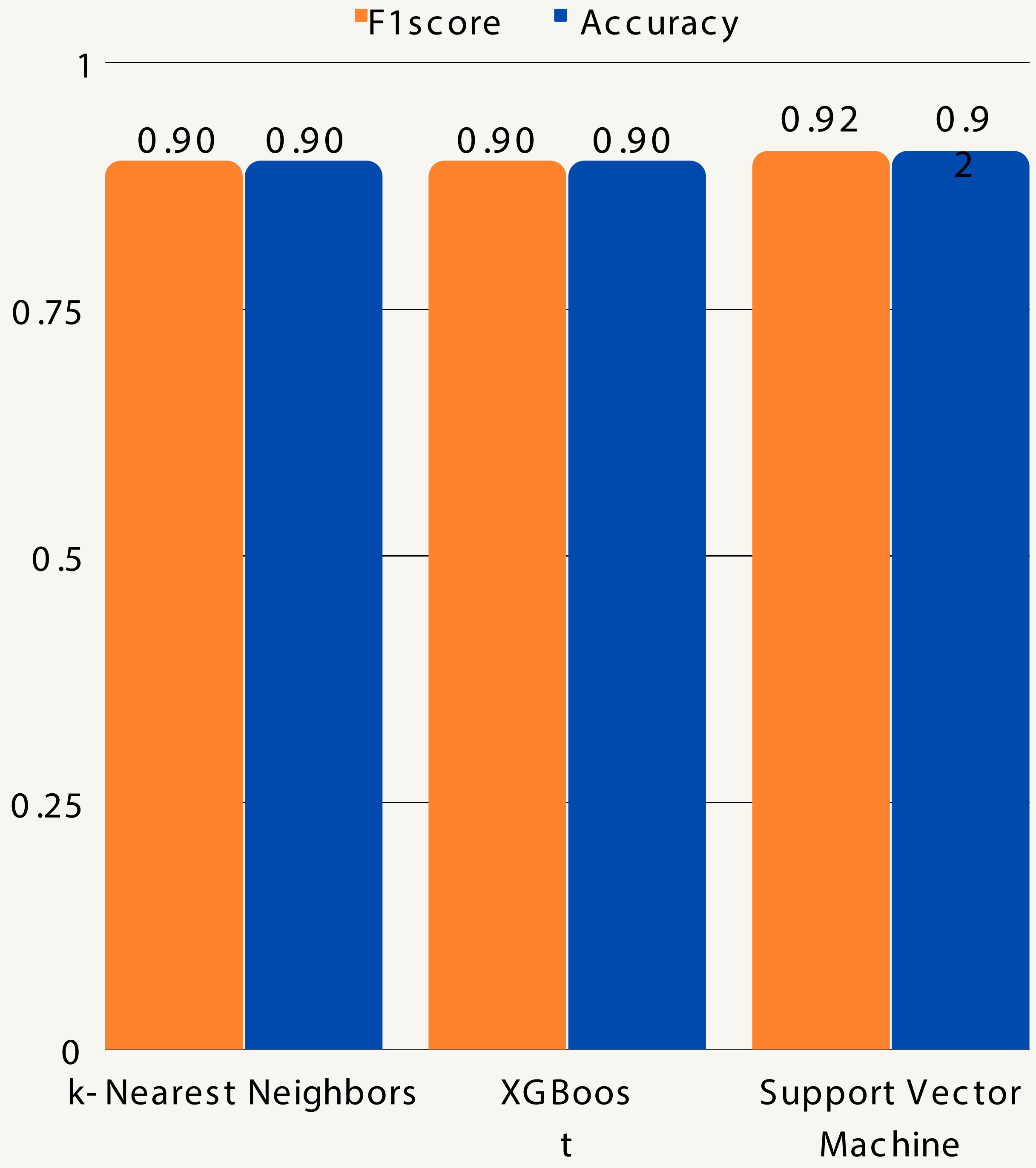
Using standard k-fold cross validation because the dataset is balanced for each category.

Evaluation

Using Accuracy as the main performance metric to compare and evaluate models.

Results





🟧F1score
🟦 Accuracy
1
0.75
0.5
0.25
0
0.90
0.90
0.90
0.90
0.92
0.9
2
k- Nearest Neighbors
XGBoos
t
Support Vector
Machine

Conclusion

News Title Classification kNN

F1score 85%

Accuracy 86%

Spam Comment Classification SVM

F1score 92%

Accuracy 92%

References

01 Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. Information, 10(4), 150.

02 Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing, 415, 295-316.

03 https://github.com/RaRe-Technologies/gensim-data

04 https://github.com/kk7nc/Text_Classification# id9

Thank you, bagidata !