IMPLEMENTATION OF ENSEMBLE METHOD ON DNA DATA USING VARIOUS CROSS VALIDATION TECHNIQUES

B. U. Bawankar
G.H. Raisoni University, Amaravati, India, (India).

Kotadi Chinnaiah
G.H. Raisoni University, Amaravati, India, (India).

Reception: 10/09/2022 **Acceptance**: 25/09/2022 **Publication**: 29/12/2022

Suggested citation:

Bawankar, B. U., y Chinnaiah, K. (2022). Implementation of ensemble method on DNA data using various cross validation techniques. *3C Tecnología. Glosas de innovación aplicadas a la pyme, 11*(2), 59-69. https://doi.org/10.17993/3ctecno.2022.v11n2e42.59-69

https://doi.org/10.17993/3ctecno.2022.v11n2e42.59-69

ABSTRACT

Due to the growing size of datasets, which contain hundreds or thousands of features, feature selection has drawn the interest of many scholars in recent years. Usually, not all columns show important values. As a result, the machine learning models may perform poorly since the noise or unnecessary columns may confound the algorithms. To address this issue, various feature selection methods have been developed to evaluate large dimensional datasets and identify their subsets of pertinent features. The data, however, frequently skews feature selection algorithms. As a result, ensemble approaches have emerged as a substitute that incorporates the benefits of single feature selection algorithms and makes up for their drawbacks. In order to handle feature selection on datasets with large dimensionality, this research aims to grasp the key ideas and links in the process of aggregating feature selection methods. The suggested idea is tested by creating a cross-validation implementation that combines a number of Python packages with functionality to enable the feature selection techniques. By identifying pertinent features in the human, chimpanzee, and dog DNA datasets, the performance of the implementation was demonstrated.

KEYWORDS

Cross-validation, Ensemble methods, Feature selection.

1. INTRODUCTION

In recent years, datasets with a lot of attributes have become more common in several fields. Microarray categorization serves as the best illustration. Numerous datasets containing this type of data have been produced as a result of improvements in DNA microarray. The majority of these datasets show that the ratio of instances to features, which range from 6 to 60 genes, is not greater than this. However, most of the genes in these datasets do not represent helpful information to support a machine learning process. In order to efficiently classify microarray data, a pre-processing stage is therefore required. This article will explain how to do so by choosing a representative subset of genes from the original set of genes(Mera-Gaona, LÅLopez, Vargas-Canas, and Neumann, 2021)[16]. The individual success of the ensemble's basis learners
and the independence of the base learners' results due to low error and great diversity are the two major factors that determine how well an ensemble performs. By utilising foundation learners of the same or different types, diverse base learners can be built. When using the same type of base learners, diversity is produced by giving each base learner in the ensemble a different training set. Different training data sets can be created using a variety of techniques, including bagging, boosting, random subspaces, random forests, and rotation forests. In order to créate a superior composite global model with more precise and trustworthy estimates or conclusions than can be produced by utilising a single model, an ensemble methodology combines a group of models, each of which addresses the same original problem. The fact that different classifier types have distinct inductive biases is one of the key reasons why ensemble methods are so successful (Gopika1 and Azhagusundari, 2014)[9]. Finding ways to enhance feature selection on datasets with high dimensionality and few examples is the major goal of this work. Additionally, cross validation is used in the display of ensemble methods to combine the benefits of several feature selection algorithms, avoid their biases, and make up for their shortcomings (Mera-Gaona et al., 2021)[16].

2. ENSEMBLE METHODS

The Ensemble categorization is founded on the idea that several experts can provide more accurate judgments than a single expert. A single composite model with higher accuracy is produced through ensemble modelling, which combines the collection of classifiers. According to research, predictions from a composite model provide better outcomes than predictions from a single model. Since the previous few decades, ensemble technique research has gained popularity. The outputs of many classifiers are combined, which minimises generalisation error, according to a number of experimental tests carried out by machine learning experts. The ensemble approaches are described in this section (Pandey and Taruna, 2014)[10-11].

(1) Bagging

The bagging technique is used to reduce variance, and the bagging ensemble method's goal is to divide the dataset into several subsets for training that are randomly chosen with replacement(Singh and Pal, 2020) [10]. The Bootstrap sampling approach provides the basis for bagging. A distinct set of bootstrap samples is produced for each iteration of the procedure in order to build a unique classifier. During the sample phase of the bootstrap sampling approach, data items are chosen at random with replacement, meaning that some instances may be repeated or some may be omitted from the original dataset. Combining all of the classifiers built in the previous phase is the next stage in the bagging process. To arrive at a final prediction, bagging combines the output of the classifiers with input from the voting process a12[11-12].

(2) Boosting

Another crucial ensemble method is the boosting classifier. It is used to develop a collection of classifiers. By fitting classifiers to data and then assessing mistakes, classifiers are serially trained in the boosting approach (Singh and Pal, 2020) [10]. The weak classifier's performance is improved by boosting to a strong level. With the help of reweighting the data instances, it creates sequential learning classifiers. All the instances are given initial weights that are equal and consistent. Each time a learning phase is completed, a new hypothesis is taught, and the examples are reweighted such that instances that were properly identified during that pase have a lower weight and the system may focus on instances that weren't. Instances that were incorrectly categorised are chosen so they can be correctly categorised in the following learning stage. This procedure keeps on till the final classifier is built. To arrive at the final forecast, the output of each classifier is finally merged using majority voting. The Boosting method has been generalised in AdaBoost(Breiman,)[12].

(3) Random Subspaces

The approach comes in two different types. Each base learner is taught using a distinct feature subspace of the initial training data set at the first form. Only decision trees may be utilised as the base learner at the second form (Gopika1 and Azhagusundari, 2014)[9].

(4) Random Forest

Breiman proposed Random Forest. Bagging plus the second kind of random subspaces can be used to formulate it (Breiman) [12]. The bagging and random subspace methods are combined to induce the tree. Although each model is a random tree rather than a single model, it differs from bagging in that each tree is created in accordance with the bootstrap sample of the training set to N. Each node is divided using yet another random step. Instead of examining all potential splits, a limited subset of features is randomly picked, and the optimum split is determined from this subset. Across all trees, the majority vote determines the final categorization [11].

(5) Rotation Forest

Rotation Forest is a brand-new ensemble approach built on the Principal Component Analysis (PCA) and decision trees. To create a training set for the base classifier using a K axis rotation of the feature subset, the attribute set F is randomly divided into K subgroups, and PCA is then performed separately to each subset. By keeping all of the PCA, Rotation Forest maintains all of the information. The basis classifier for Rotation Forest is the decisión tree(Pandey and Taruna, 2014) [11].

3. CROSS VALIDATION TECHNIQUES

A statistical technique called cross-validation determines how well a trained model will perform on unobserved data. By training the model on a subset of the input data and testing it on a different subset, the model's effectiveness is confirmed. Building a generalised model is assisted by cross-validation. Cross-validation is helpful for both performance estimate and model selection since modelling is an iterative process.

Cross-validation involves the following three steps:

i. Split the dataset into two sections: a training section and a testing section.
ii. Use the training dataset to train the model.
iii. Use the testing set to gauge the model's effectiveness. Check for problems if the model doesn't perform well with the testing set.

If a model can predict accurately for a variety of input data and does well on unknown data, it is stable and consistent. Evaluation of the stability of machine learning models is aided by crossvalidation.

The dataset has to be divided into three separate sections for training and testing the model:

- Training Data: Using the training data, the model is trained to discover the dataset's hidden characteristics and patterns. The model continually assesses the data to better understand its behaviour, and then it modifies itself to achieve its goal. Basically, it's employed to fit the models.
- Validation Data: This is used to confirm that the model's training results were accurate. It aids in adjusting the hyper-parameters and settings of the model appropriately. The prediction error for model selection is estimated using the validation data. Validation data helps prevent over-fitting models.
- Test Data: Following training, the test data confirms that the trained model is capable of making precise predictions. It is used to evaluate the generalisation error of the last model chosen (Hulu and Sihombing, 2020)[1](Jung and A K-Fold, 2015)[7][8](Wu,)[13-14](??,).

This paper discusses eight alternative cross-validation approaches, each with advantages and disadvantages that are stated below

(1) Leave p out cross-validation

An exhaustive cross-validation strategy called leave p-out cross-validation uses the p-observation as validation data while utilising the remaining data to train the model. This is repeated in all possible ways on a validation set of p observations and a training set to trim the original sample. In order to estimate the area under the ROC curve of a binary classifier in a virtually unbiased manner, leave-pair-out cross-validation, a variation of Leave p-out with p=2, has been suggested (Kumar, 2020)[14].

(2) Leave one out cross-validation

A thorough cross-validation method is leave-one-out cross-validation. It falls within the leave p-out cross validation category with the instance of p=1. The first row of a dataset of n rows is chosen for validation, and the remaining n-1 rows are utilised to train the model. The second row is chosen for validation and the remainder is used to train the model for the following iteration. Similar to that, the procedure is repeated up to n operations or phases. Cross-validation techniques i.e. leave p-out and leave One-out that learn and test in every conceivable way are known as exhaustive cross-validation techniques. They share the advantages such as straightforward, understandable, and simple to use and disadvantages such as the model might provide a little bias and a lot of computing time is needed[13-14].

(3) Holdout cross-validation

The dataset is randomly divided into training and validation data in holdout cross-validation. In general, training data are split more evenly than test data. The model is created using training data, and validation data is used to assess the model's effectiveness. The model becomes better as more data are used to train it. The holdout cross-validation approach isolates training data from a sizable amount of data. The advantages for this such as straightforward, understandable, and simple to use and disadvantages such as it's not suitable for an unbalanced dataset and a lot of data is not being used to train the model (Raschka, 2020)[5].

(4) Repeated random sub-sampling validation

The dataset is randomly divided into training and validation in repeated random subsampling validation, commonly known as Monte Carlo cross-validation. Unlikely k-fold cross-validation separates the dataset into random splits rather than groups or folds in this. Analysis determines the number of iterations; it is not a set quantity. The outcomes are then multiplied by the divides. Advantage for such validation is i.e. there is no relation exists between the number of iterations or divisions and the fraction of train and validation splits and the disadvantages such as possible that some samples won't be used for either training or validation and not appropriate for a dataset with imbalance [5][14].

(5) k-fold cross-validation

The original dataset is evenly divided into k subparts or folds for k-fold cross-validation. For each iteration, one of the k-folds or groups is chosen as the validation data, while the remaining (k-1) groups are chosen as the training data. Until each group is considered as validation and the rest as training data, the procedure is repeated k times. The mean accuracy of the kmodels validation data is used to calculate the model's final accuracy. The model exhibits little bias, low temporal complexity and both training and validation use the complete dataset is the advantages and the disadvantage is unsuitable for a dataset with imbalance[1-7](Hulu and Sihombing, 2020) (Darapureddy, Karatapu, and Tirumala, 2019)(PAYAM REFAEILZADEH, 2008)(Arumugam, Professor, Department of Statistics, Manonmaniam Sundaranar University, Tirunelveli (Tamil Nadu), India., Kadhirveni, Priya, Manimannan, Research Scholar, Department of Statistics, Manonmaniam Sundaranar University, Tirunelveli (Tamil Nadu), India., Assistant Professor, Department of Statistics, Dr. Ambedkar Government Arts College, Vyasarpadi, Chennai (Tamil Nadu), India., and Assistant Professor. Department of Statistics, TMG College of Arts and Science, Chennai (Tamil Nadu), India., 2021)(Raschka, 2020)(Raschka, 2020).

(6) Stratified k-fold cross-validation

All the cross-validation methods mentioned above might not be effective with an unbalanced dataset. Unbalanced dataset issue was resolved by stratified k-fold cross-validation. The dataset is divided into k groups or folds in stratified k-fold cross-validation such that the validation data has an equal number of instances of the target class label. This makes sure that, especially when the dataset is unbalanced, one specific class is not overrepresented in the validation or train data. The average of the scores for each fold is used to get the final score. As a benefit, it performs well for an unbalanced dataset (PAYAM REFAEILZADEH, 2008)[3][14].

(7) Time Series cross-validation

When dealing with problems involving time series, the data's order is crucial. Data divided randomly or in k-folds into train and validation for time-related datasets might not produce the best results. The forward chaining method, also known as rolling cross-validation, is used to divide the time-series dataset's data into train and validation groups. The subsequent instance of train data can be used as validation data for a certain iteration(a13,)[13].

(8) Nested cross-validation

We obtain a subpar estimate of the error in training and test data while using k-fold and stratified k-fold cross-validation. In the prior techniques, hyper-parameter adjustment is done individually. Nested cross-validation is necessary when cross-validation is used to tune the hyper-parameters and generalise the error estimate at the same time. Both the stratified k-fold and k-fold variations can use nested cross validation [14].

4. BASIC PROCESS OF MACHINE LEARNING

The field of machine learning blends traditional statistical methods with computer science techniques. In order to extract knowledge from massive volumes of data for application in science, computing, or industry. We thoroughly discuss the machine-learning process from six angles [2](Darapureddy et al., 2019).

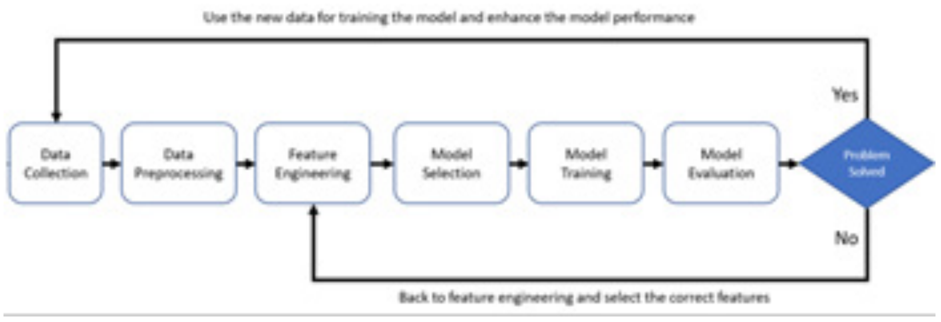

Use the new data for training the model and enhance the model performance
Yes
Data Collection
Data Preprocessing
Feature Engineering
Model Selection
Model Training
Model Evaluation
Problem Solved
No
Back to feature engineering and select the correct features

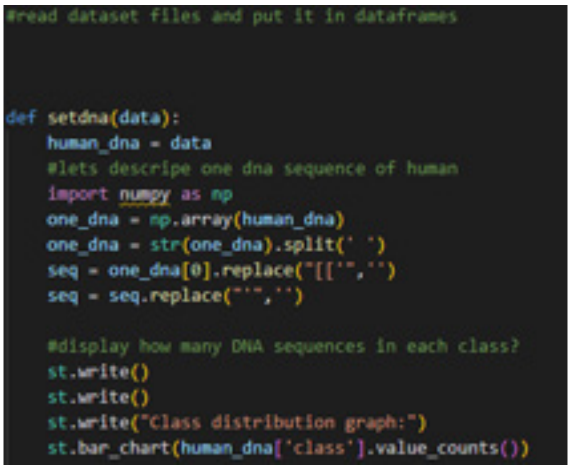
Fig. 1: Basic Process.
Source: datavalley.

1. Data Collection: Gather all the information you need from the many systems that might contribute to your situation.
2. Data Pre-processing: Prior to processing and analysis, raw data must be cleaned and transformed. Prior to processing, it is a crucial phase that frequently entails reformatting data, making adjustments to data, and fusing data sets to enhance data.- Data Cleaning: The initial phase in data mining consists of removing incomplete or inconsistent data since data sets frequently contain missing data and inconsistent data. Low data quality will have a significant negative influence on the information extraction process.- Data Integration: If the data to be examined come from many sources, they must be reliably aggregated.
3. Feature Engineering: This covers all modifications made to the data, from cleaning it up to ingesting it into the machine learning model. You choose and prepare the features that will be used in your machine learning model in this stage, making sure they are in the format required by the model.
4. Selection of Model: Choose the best model for the situation and then make any necessary adjustments.

5. Train the Model: A machine learning (ML) model is trained by feeding training data to the learning algorithm. The model artefact produced during training is referred recognised as a " ML model."
6. Evaluate the Model: This methodical technique will serve as a guide for evaluating the efficacy and efficiency of training.

5. RESULTS AND DISCUSSIONS

Data Selection: We provide three sorts of data sets i.e. human, chimpanzee and dog DNA for insights which represent DNA sequences that contain seven gene classes. Representation of gene family with seven gene classes and class label as G-protein coupled receptors(0), tyrosine kinase(1), tyrosine phosphate(2), synthetase(3), synthase(4), ion channel(5) and transcription factor(6). Data Processing: The dataset is processed and check how many DNA sequences in each class with class distribution graph by using tools VS code and streamlit through python.

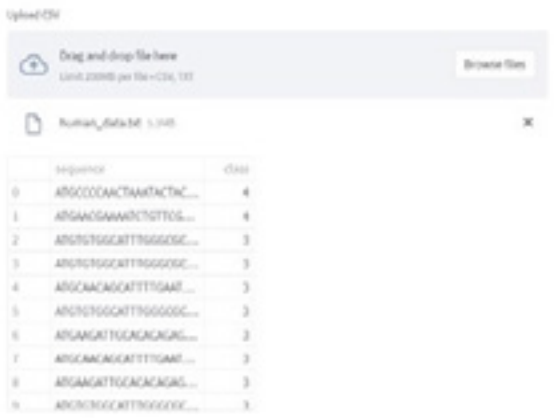


```python
#read dataset files and put it in dataframes


def setdna(data):
    human_dna = data
    #lets descripe one dna sequence of human
    import numpy as np
    one_dna = np.array(human_dna)
    one_dna = str(one_dna).split(' ')
    seq = one_dna[0].replace("[['",'')
    seq = seq.replace("'",'')

    #display how many DNA sequences in each class?
    st.write()
    st.write()
    st.write("Class distribution graph:")
    st.bar_chart(human_dna['class'].value_counts())
```

Fig 2: Python code for DNA sequence.



Upload CSV

Drag and drop file here
Limit 200MB per file • CSV, TXT
Browse files

human_data.txt 1.3MB

	sequence	class
0	ATGCCCCAACTAAATACTAC...	4
1	ATGAACGAAAATCTGTTCG...	4
2	ATGTGTGGCATTTGGGCGC...	3
3	ATGTGTGGCATTTGGGCGC...	3
4	ATGCAACAGCATTTTGAAT...	3
5	ATGTGTGGCATTTGGGCGC...	3
6	ATGAAGATTGCACACAGAG...	3
7	ATGCAACAGCATTTTGAAT...	3
8	ATGAAGATTGCACACAGAG...	3
9	ATGTGTGGCATTTGGGCGC...	3

Fig 3 DNA Sequence with class.

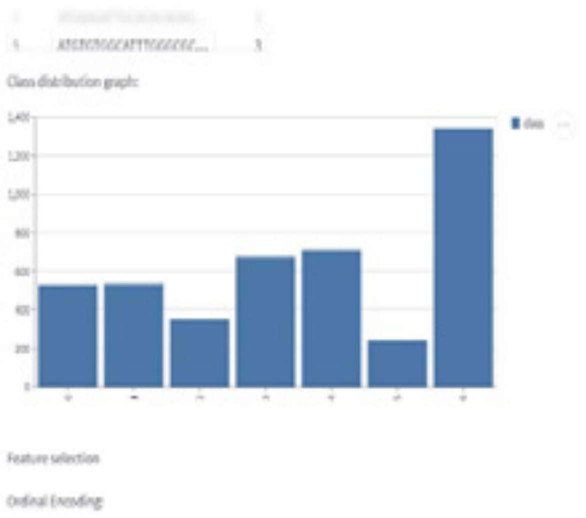
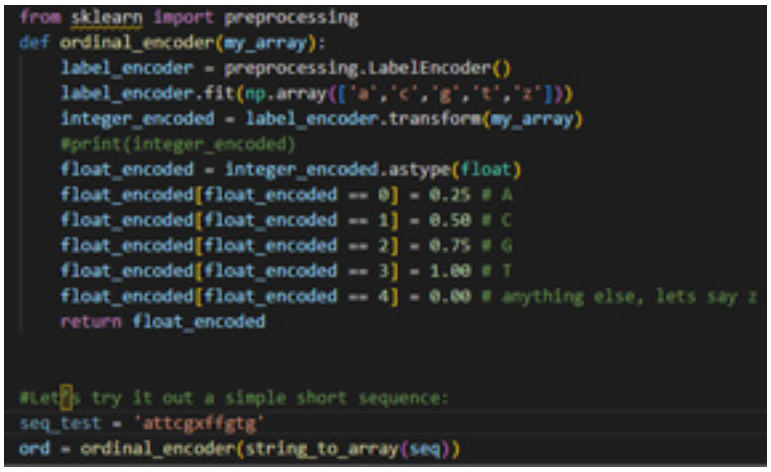


Fig 4 Class distribution graph.

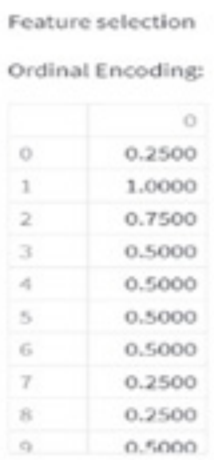
Feature Selection: Although the DNA sequence in a show is represented by characters, machine learning algorithms need numerical values or feature matrices. In order to convert these characters into values, we use three general approaches such as ordinal encoding, one hot encoding and kmers counting. Ordinal Encoding: With this method, each nitrogen base must be encoded as an ordinal value. ”A, T, G, and C,” for instance, becomes [0.25, 0.5, 0.75, and 1.0]. Any additional base, like ”Z,” may be a 0.



```
from sklearn import preprocessing
def ordinal_encoder(my_array):
    label_encoder = preprocessing.LabelEncoder()
    label_encoder.fit(np.array(['a','c','g','t','z']))
    integer_encoded = label_encoder.transform(my_array)
    #print(integer_encoded)
    float_encoded = integer_encoded.astype(float)
    float_encoded[float_encoded == 0] = 0.25 # A
    float_encoded[float_encoded == 1] = 0.50 # C
    float_encoded[float_encoded == 2] = 0.75 # G
    float_encoded[float_encoded == 3] = 1.00 # T
    float_encoded[float_encoded == 4] = 0.00 # anything else, lets say z
    return float_encoded


#Let's try it out a simple short sequence:
seq_test = 'attcgxffgtg'
ord = ordinal_encoder(string_to_array(seq))
```

Fig 5 Python Code for ordinal encoding.



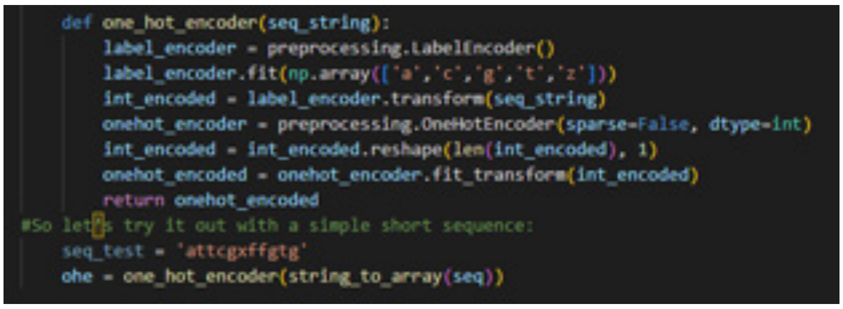
Feature selection

Ordinal Encoding:

	0
0	0.2500
1	1.0000
2	0.7500
3	0.5000
4	0.5000
5	0.5000
6	0.5000
7	0.2500
8	0.2500
9	0.5000

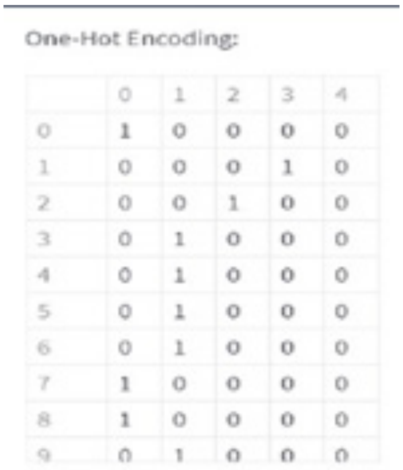
Fig 6 Feature Selection through ordinal encoding.

One-hot Encoding: “A,C,G,T,Z” would become [1,0,0,0,0], [0,1,0,0,0], [0,0,1,0,0], [0,0,0,1,0], [0,0,0,0,1], and these one-hot encoded vectors can either be concatenated or turned into 2- dimensional arrays.



```
    def one_hot_encoder(seq_string):
        label_encoder = preprocessing.LabelEncoder()
        label_encoder.fit(np.array(['a','c','g','t','z']))
        int_encoded = label_encoder.transform(seq_string)
        onehot_encoder = preprocessing.OneHotEncoder(sparse=False, dtype=int)
        int_encoded = int_encoded.reshape(len(int_encoded), 1)
        onehot_encoded = onehot_encoder.fit_transform(int_encoded)
        return onehot_encoded
#So let?s try it out with a simple short sequence:
    seq_test = 'attcgxffgtg'
    ohe = one_hot_encoder(string_to_array(seq))
```

Fig 7 Python code for One-hot encoder.

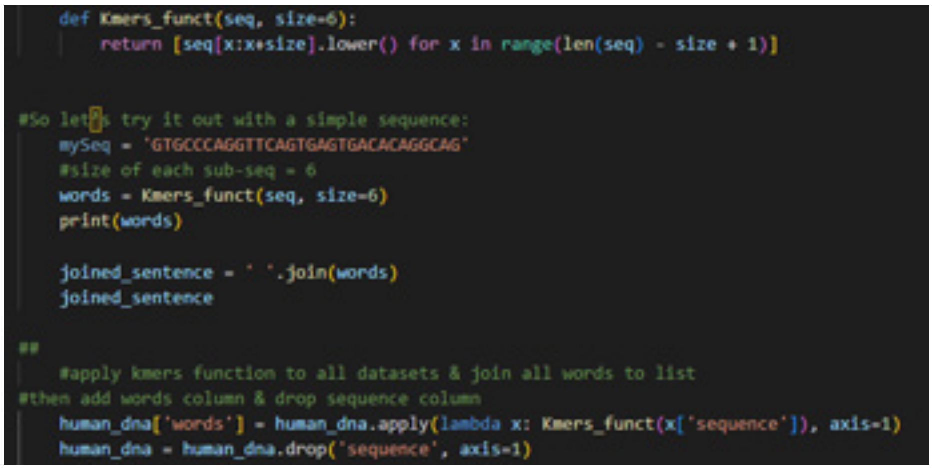


One-Hot Encoding:

	0	1	2	3	4
0	1	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	1	0	0	0
4	0	1	0	0	0
5	0	1	0	0	0
6	0	1	0	0	0
7	1	0	0	0	0
8	1	0	0	0	0
9	0	1	0	0	0

Fig 8 Feature Selection through One-hot encoding.

K-mers counting: The approach we take here is simple and manageable. We first divide the lengthy biological sequence into overlapping k-mer length ”words”. If we use ”words” of length 6 (hexamers), for instance, ”ATGCATGCA” becomes ”ATGCAT,” ”TGCATG,” ”GCATGC,” and ”CATGCA.” In light of this, our example sequence is divided into 6 hexamer words.



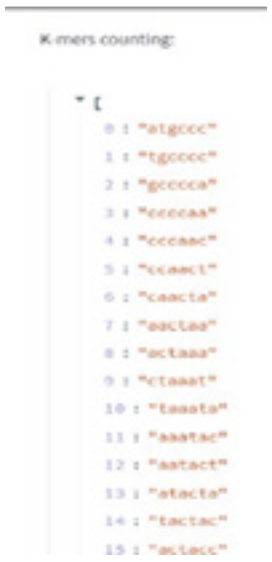
```
    def Kmers_funct(seq, size=6):
        return [seq[x:x+size].lower() for x in range(len(seq) - size + 1)]


#So let?s try it out with a simple sequence:
    mySeq = 'GTGCCCAGGTTCAGTGAGTGACACAGGCAG'
    #size of each sub-seq = 6
    words = Kmers_funct(seq, size=6)
    print(words)

    joined_sentence = ' '.join(words)
    joined_sentence

##
    #apply kmers function to all datasets & join all words to list
#then add words column & drop sequence column
    human_dna['words'] = human_dna.apply(lambda x: Kmers_funct(x['sequence']), axis=1)
    human_dna = human_dna.drop('sequence', axis=1)
```

Fig 9 Python code for K-mers counting.



K-mers counting:

```
▾ [
    0 : "atgccc"
    1 : "tgcccc"
    2 : "gcccca"
    3 : "ccccaa"
    4 : "cccaac"
    5 : "ccaact"
    6 : "caacta"
    7 : "aactaa"
    8 : "actaaa"
    9 : "ctaaat"
    10 : "taaata"
    11 : "aaatac"
    12 : "aatact"
    13 : "atacta"
    14 : "tactac"
    15 : "actacc"
```

Fig 10 Feature Selection through k-mers counting.

6. CONCLUSION AND FUTURE WORK

This method of feature selection and feature extraction from DNA data sequence was successfully completed. Here, we employed K-mer counting, one-hot encoding, and ordinal encoding as the language for choosing DNA sequence features in python libraries. We have demonstrated the result using these libraries in the forms of a matrix, vector, and graph. In future, we also retrieved K-mers to use in the classifier process.

REFERENCES

https://devopedia.org/cross-validation. Accessed: 2022-12-13.

[1] Arumugam, P., Professor, Department of Statistics, Manonmaniam Sundaranar University, Tirunelveli (Tamil Nadu), India., Kadhirveni, V., Priya, R. L., Manimannan, Research Scholar, Department of Statistics, Manonmaniam Sundaranar University, Tirunelveli (Tamil Nadu), India., Assistant Professor, Department of Statistics, Dr. Ambedkar Government Arts College, Vyasarpadi, Chennai (Tamil Nadu), India., and Assistant Professor. Department of Statistics, TMG College of Arts and Science, Chennai (Tamil Nadu), India. 2021. Prediction, cross validation and classification in the presence COVID-19 of indian states and union territories using machine learning algorithms. International Journal of Recent Technology and Engineering (IJRTE) 10, 1 (May), 16–20.

[2] Breiman, L. Bagging predictors”. Boston. Manufactured in The Netherlands.

[3] Darapureddy, N., Karatapu, N., and Tirumala, K. 2019. Research of machine learning algorithms using K-Fold cross validation”. International Journal of Engineering and Advanced Technology (IJEAT).

[4] Gopika, D. and Azhagusundari, B. 2014. An analysis on ensemble methods in classification tasks”. International Journal of Advanced Research in Computer and Communication Engineering 3, 7.

[5] Hulu, S. and Sihombing, P. 2020. Analysis of performance cross validation method and K-Nearest neighbor in classification data. International Journal of Research and Review 7.

[6] Jung, Y. and A K-Fold. 2015. Averaging cross-validation procedure”. Journal of Nonparametric Statistics.

[7] Kumar, S. 2020. Understanding 8 types of cross-validation. https://towardsdatascience.com/understanding-8-types-of-cross-validation-80c935a4976d. Accessed: 2022-12-13.

[8] Mera-Gaona, M., LÅLopez, D. M., Vargas-Canas, R., and Neumann, U. 2021. Framework for the ensemble of feature selection methods. Appl. Sci. (Basel) 11, 17 (Sept.), 8122.

[9] Pandey, M. and Taruna, S. 2014. A comparative study of ensemble methods for students' performance modeling”. International Journal of Computer Applications 103, 8, 975–8887.

[10] PAYAM REFAEILZADEH, LEI TANG, H. L. A. S. U. 2008. Cross-validation'. Cross validation Bootstrap.

[11] Raschka, S. 2020. Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning".

[12] Singh, R. and Pal, S. 2020. Machine learning algorithms and ensemble technique to improve prediction of students performance". International Journal of Advanced Trends in Computer Science and Engineering 9, 3, 3970–3976.

[13] Wu, S.-H. Cross Validation & Ensembling. Taiwan.