Akanksha Dewangan(MT19049)

**********MLBA HACKATHON**********

Readme:

A. Command :

One CMD.ipynb is uploaded where command line argument codes are written to run the command.

!python3 /content/akankshadewangan_mt19049_hackathon.py "Training_dataset.csv" "New_Validation_Dataset.csv" "training_dipeptide_result.csv" "testing_dipeptide_result.csv" "logistic.txt" "svm.txt" "randomforest.txt" "extraTree.txt" "mlp.txt"

B.Run code : Hackathon.py of best MCC and accuracy value:

(i) Initially we go to the folder where python is installed and put our .py file and other input files .csv into that:

(ii) Path where python is present:

C:\Users\akanksha\AppData\Local\Programs\Python\Python37\Scripts

(iii) **>ipython3 !python3 /content/akankshadewangan_mt19049_hackathon.py "Training_dataset.csv" "New_Validation_Dataset.csv" "training_dipeptide_result.csv" "testing_dipeptide_result.csv" "logistic.txt" "svm.txt" "randomforest.txt" "extraTree.txt" "mlp.txt"**

(iv)prediction of training set are present in csv: ouput.**csv**

(v) metrics value present in:**"extraTree.txt"**

Akanksha Dewangan(MT19049)

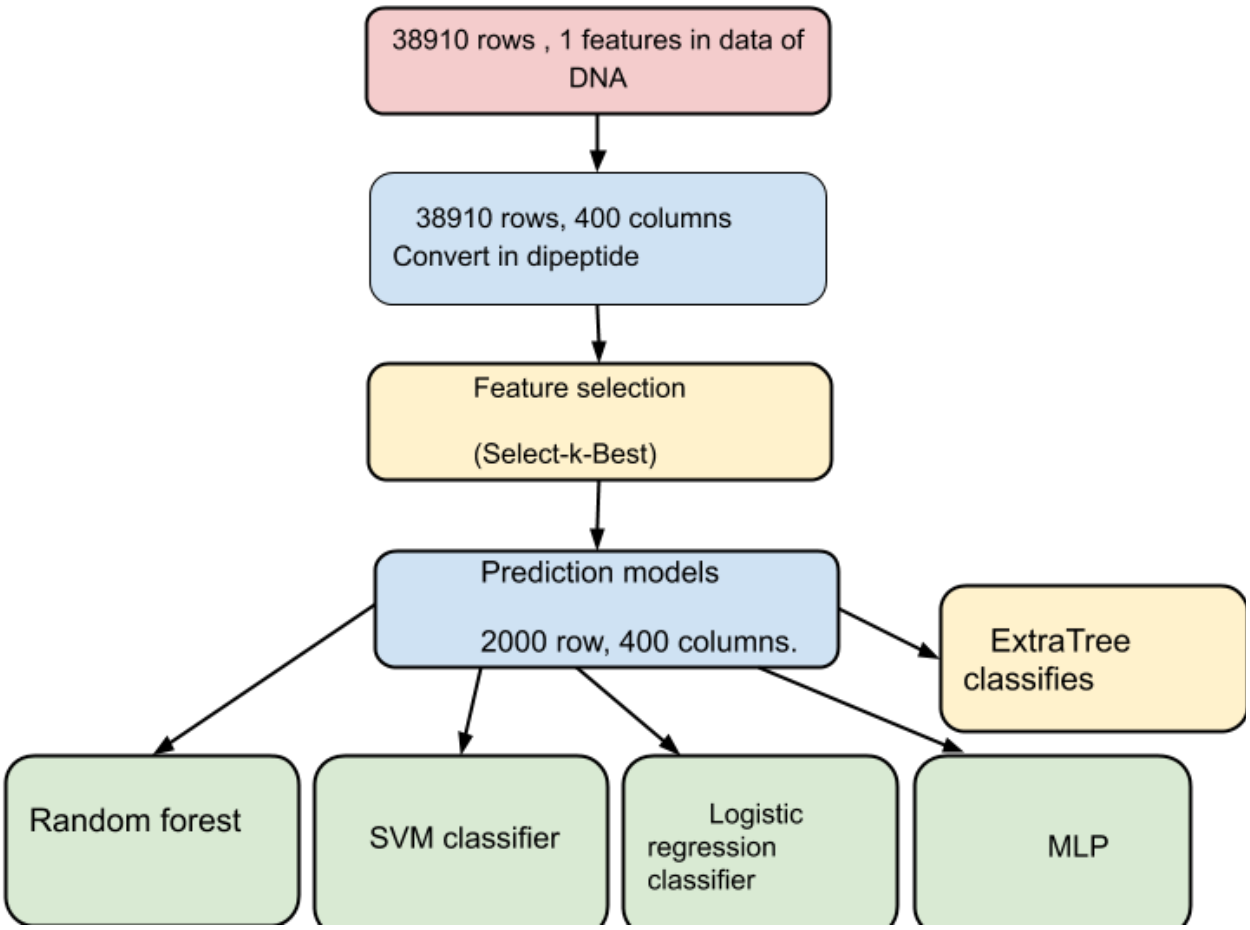
REPORT

AIM:

Prediction of DNA interacting residue. Classify DNA interacting and noninteracting residue in DNA sequence.

Preprocessing and Methodology:

Complete description through flow chart:



Akanksha Dewangan(MT19049)

1.Transformation of overall training data into two types of feature extraction/generated from a p-feature web server site.

A.Amino acid composition

B.Dipeptide composition.

A. Amino acid Composition:

$$AACi = Ri/L$$

Where, AACi= amino acid composition of residue type I

Ri and L =number of residues of type I and length of sequence.

B. Dipeptide Composition :

$$DPCi\ j = Di\ j/\ L - j$$

Where, DPCj i = the fraction or composition of dipeptide of type i for jth order.

D j i and L = the number of dipeptides of type i and length of a protein.

-**Traditional dipeptide**:-(if j=1 then that dipeptide is traditional)

higher order dipeptide D j i is made of residue Ri and Ri+j where value of j is 2 or more.

2. Feature Selection:

(i) SelectKBest RFE(Recursive feature elimination) for:

- 100 feature extraction
- 70 feature extraction
- 40 feature extraction

(ii) SelectKBest, f_classif sel_f = SelectKBest(score_func=f_classif, k=100)

- k= 70
- k= 40

2.Try to convert it into standardScalar() but it is not useful as the converted dipeptide data range between 0 to 1.

3.Then various models were applied and their observations were checked.

4.Deep learning LSTM is also tried; it didn't give any gud score.

MODEL-

-Logistic regression

-SVM classifier

-MLP classifier

-Extratree classifier

-Random forest classifier.

-Deep learning(sequential deep neural network)

The validation evaluation metric used is MCC (Mathews correlation coefficient) and accuracy value.

Classifiers	**Validation Accuracy**	**Validation MCC**	**Leaderboard**

Akanksha Dewangan(MT19049)

1. `LogisticRegression(solver='liblinear',class_weight='balanced').fit(X_train, y_train)`	67.32	34.66	22-32
2.`svm.SVC(kernel='linear')`	87.6	67.66	48
3. `RandomForestClassifier(max_depth=150,n_estimators=500,n_jobs=-1,random_state=42)`	84.6	69.2	47.101
4.`ExtraTreesClassifier(n_estimators=500,max_depth=110,random_state=0,warm_start=True)`	86.66	72.88	47.33-48.082
5.`MLPClassifier(solver='sgd', alpha=1e-50,hidden_layer_sizes=(70, 2), random_state=0)`	67.8	38.5	37.55

`6.model.fit(X_train.values,np.asarray(y_train), epochs=100,validation_split=0.2,callbacks=[mcp_save],batch_size=10,verbose=0)`	78.87	57.7	35.55

All the above models are used and were trained with varying parameters. They are tuned with different values and observe the accuracies and MCC scores an evaluation metric.

CONCLUSION-

-ExtraTreeClassifier is best among all having following parameters:

```
ExtraTreesClassifier(n_estimators=500,max_depth=110,random_state=0,warm_start=True)
```

It is giving 49.088 MCC value in the leaderboard and 73 accuracy in validation data.

Akanksha Dewangan(MT19049)