Classifying a sequence as coding and noncoding

INDEX

ACKNOWLEDGEMENT

First and foremost I would like to thank the Lord Almighty for his presence and immense blessings throughout the project work.

It's a matter of pride and privilege for me to express my deep gratitude to the management of SASTRA for providing me the necessary facilities and support.

I am highly elated in expressing my sincere and abundant respect to the PROFESSOR ASHOK PALANIAPPAN for giving me this opportunity to bring out and implement my ideas in this project.

PROJECT DONE BY:

1. PILLALAMARRI GOWTHAM KUMAR

2. AISHWARYA V

3. VARSHITHA REDDY V

Motivation and Introduction

Genome, the genetic material of an organism where the whole of its heredity information is encoded in its DNA. The genome contains genes, which are tightly packed in chromosomes. It is found in the cell's nucleus of organisms. It runs for a length of 10kbp to 100Gbp. A genome contains coding regions and non-coding regions. Genes are present in the coding region. The structure of a gene is very complex. It starts with a promoter sequence and ends with a terminator sequence. These are consensus sequences and highly variable from one organism to another.

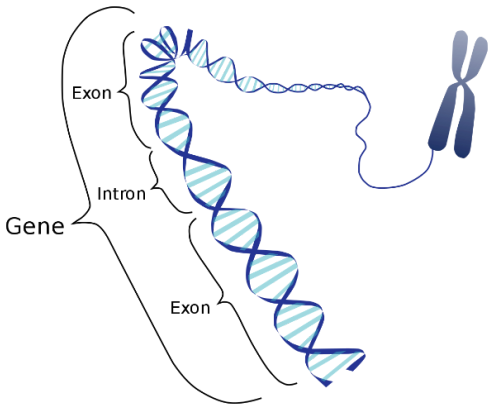


Fig 1.0 Structure of gene

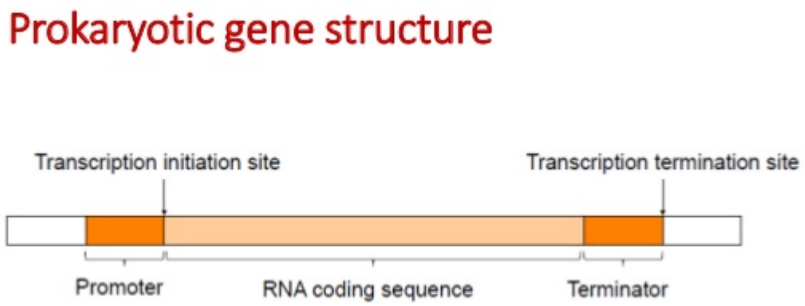


Fig 2.0 3' – 5' DNA template, showing promoter sequence, terminator sequence

The eukaryotic gene is much more complex than the prokaryotic gene. A eukaryotic gene consists of both introns and exons, whereas it is not in the case of prokaryotes. It consists of only exons in an RNA Coding sequence. The sequence of the promoter and terminator is not the same in every organism, it may vary with some exceptions.

Many statistical Algorithms and wet-lab experiments are there to predict whether a sequence can be in a coding region or a non-coding region. But these methods give very little accuracy for the given sequence. Many servers are there where prediction is based on the type of organism. When an organism has changed the method of prediction will change and it is impossible to write different prediction methods for different species.

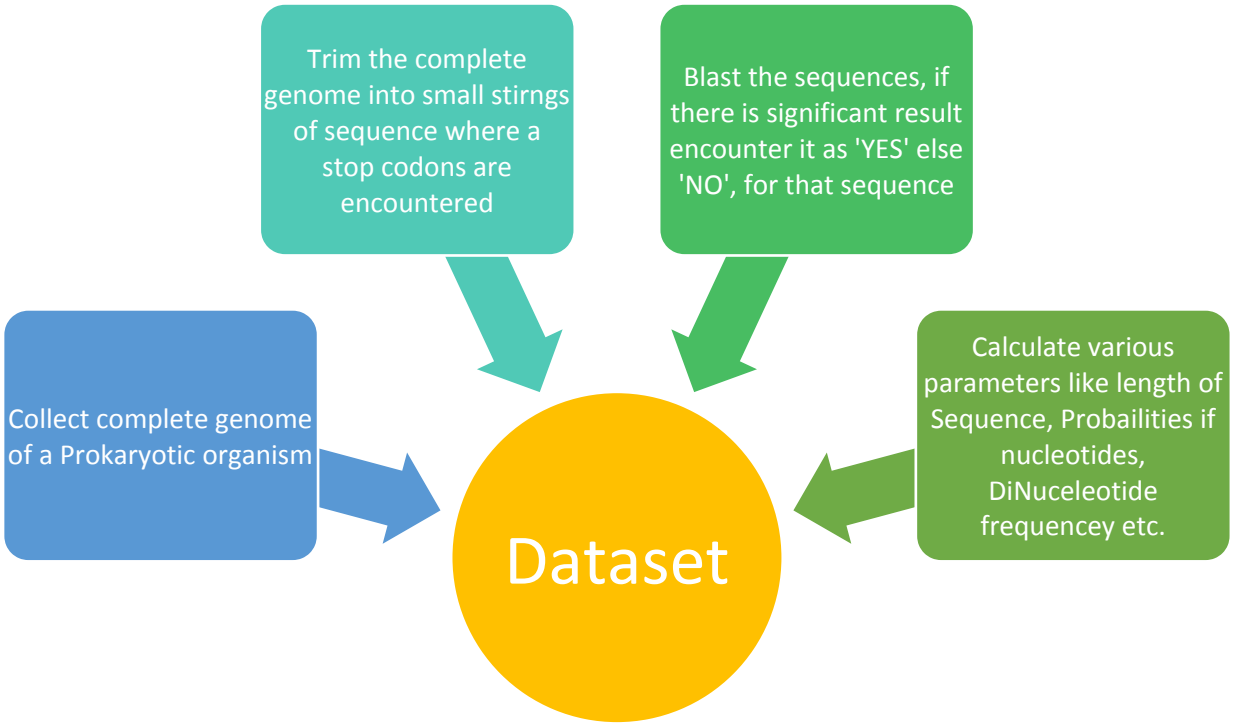
So, we tried to implement a Machine Learning Algorithm for better accuracy and we are training the model in such a way that it should give high accuracy for every prokaryote.

The data available with us consists of a long sequence runs for a variable length. In order to implement machine learning with this data is impossible. Because the algorithms that are available can under and predict analysis from numbers. So, we need to observe hidden pattern present in our data. These observations are fed to the algorithm as training data. It will understand the pattern present in this data and can able to map the outcome variable.

Objectives

- High accuracy.
- More Efficiency.
- It can be able to predict for every organism without restriction on the type of specie.

Steps Involved in Building Dataset



We considered 6 different species genomes to build the dataset. Following are the details of the species:

1. Mycoplasma Bovoculi M165/69(T) strain.

2. Escherichia coli K12.
3. Brassica napus phytoplasma.
4. Pneumonia.
5. Salmonella enterica.
6. Thermoplasmatales archaeon.

Initially, the genome is read into a string and this string is divided into several fragments where a stop codon is encountered. These small fragments are formatted into fasta and subjected as an input file to blastx (translated nucleotide -> Protein).

```
>code2
ATGCTGTATCCGGCGCCGACGACAACTATGAACAGAGTCGCTACGCTTA
>code3
AAGTGTCTTAAGAGAGTGCATTGAGTTGCCCATGACTTAAAACAGCT
>code4
ATCTTTGTTCACCAGAAAGTTTCATCCATTTTCTTCCACCCAAGTTGCG
>code5
AGGAATACCACAAAGCGTGGAAAAGTGGTGGTACATGTGTGGAATCGCT
>code6
TTATGTGGGATGGCTGGTTCAGACATCAGGATATGGCTGAGGGTTACCT
```

Fig 3.0 Short segments which are obtained from trimming a genome, is formatted into fasta file and subjected as an input for the blastx

Blast results are analyzed if there is a significant result found that sequence can be in the coding region else non-coding region. Also, the length of the sequence single-nucleotide frequencies, conditional probabilities of diNucleotides are considered.

Description of Attributes present in Dataset

1. Sequence.
2. Length of the Sequence.
3. GC Percentage.
4. Presence of ATG Codon.
5. Single Nucleotide Probabilities (A, C, G, T).
6. Conditional Probabilities (AA, AC, AG, AT, TA, TC, TG, TT, CA, CC, CG, CT, GA, GC, GG, GT).
7. Outcome Variable (Yes or No).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	Sequence	Length	GCPercer	ATG	A	C	G	T	AA	AC	AG	AT	TA	TC	TG	TT	CA	CC	CG	CT	GA	GC	GG	GT	Coding_Region
2	ATGAATA	68	1.471	1	0.485	0.059	0.103	0.353	0.575	0.031	0.06	0.303	0.334	0.082	0.167	0.416	0.254	0	0.254	0.492	0.573	0.146	0	0.282	1
3	TATGGTT	29	0	1	0.414	0.034	0.103	0.448	0.5	0	0.082	0.333	0.308	0.076	0.076	0.538	1	0	0	0	0.33	0	0.33	0.33	0
4	TGATTAT	33	0	1	0.364	0.121	0.152	0.364	0.5	0.082	0.082	0.332	0.168	0.168	0.25	0.332	0	0.248	0.248	0.504	0.796	0	0	0.197	0
5	ACACAAA	102	0.98	1	0.51	0.088	0.118	0.284	0.576	0.116	0.057	0.249	0.208	0.07	0.31	0.415	0.784	0	0	0.227	0.661	0.085	0	0.169	1
6	GTTTGGA	78	1.282	0	0.449	0.192	0.154	0.205	0.486	0.2	0.2	0.114	0.063	0.249	0.127	0.561	0.599	0.198	0	0.135	0.669	0.084	0.169	0.084	1
7	ATTTTAA	84	3.571	1	0.405	0.167	0.119	0.31	0.499	0.148	0.148	0.205	0.194	0.229	0.077	0.5	0.641	0	0.144	0.144	0.202	0.303	0.101	0.403	1
8	AAGAATT	90	2.222	1	0.289	0.178	0.156	0.378	0.422	0.114	0.076	0.384	0.148	0.177	0.148	0.529	0.315	0.315	0.062	0.247	0.282	0.141	0.429	0.141	1
9	GTAATTT	40	2.5	0	0.475	0.05	0.125	0.35	0.526	0.053	0.105	0.316	0.571	0	0.071	0.286	0.5	0	0	0.5	0	0.2	0.2	0.6	0
10	TGATTTT	161	0.621	1	0.366	0.081	0.174	0.379	0.423	0.085	0.101	0.374	0.23	0.082	0.296	0.393	0.383	0.148	0	0.457	0.534	0.034	0.144	0.287	1
11	TAAATAT	54	1.852	1	0.463	0.148	0.111	0.278	0.4	0.121	0.121	0.32	0.468	0.068	0.068	0.399	0.5	0.378	0.128	0	0.667	0.171	0.171	0	1
12	AATCCTT	22	0	0	0.273	0.136	0.182	0.409	0.333	0	0.165	0.333	0	0.222	0.222	0.555	0.331	0.331	0	0.331	0.5	0	0.247	0.247	0
13	ATCGTTT	7	0	0	0.143	0.143	0.143	0.571	0	0	0	1	0	0.25	0	0.501	0	0	1	0	0	0	0	1	0
14	ATGGGGC	83	0	1	0.398	0.072	0.229	0.301	0.455	0.151	0.181	0.211	0.199	0.04	0.199	0.522	0.5	0	0.167	0.333	0.472	0	0.367	0.157	1
15	ACAAATA	27	0	0	0.481	0.222	0.074	0.222	0.462	0.385	0.077	0.077	0.167	0.167	0.167	0.333	0.5	0	0	0.5	1	0	0	0	0

Fig 4.0 Dataset with 25 attributes.

Methods Implemented

- **Data Preprocessing**

Handling Null Values or NA's

Data contains the probabilities of Single nucleotides and dinucleotides. These frequencies replicate the sequence. In these records, we will observe 0's for some nucleotide frequencies. This indicates, that particular nucleotide does not appear in that sequence. To handle null values, these null values are replaced by the mean of that particular attribute. But here these frequencies indicate the state of the sequence. So, these null values shouldn't be replaced by mean values.

Label Encoder

Outcome variable is in the form of YES or NO. Algorithm cannot understand this type of outcome. Hence these can be labelled into if YES => 1, else 0. By using label encoder, we can transform outcome variable.

0	Yes
1	No
2	No
3	Yes
4	Yes
...	...
4380	Yes
4381	Yes
4382	No
4383	Yes
4384	No

Fig 5.1 Outcome variable for each instance transformation before using label encoder

0	1
1	0
2	0
3	1
4	1
..	..
4380	1
4381	1
4382	0
4383	1
4384	0

Fig 5.2 Outcome variable after after using label encoder.

Outliers

The length of the sequence is one of the attributes in this dataset. Nucleotide frequencies are range from 0-1. But the length and GC percentage vary sequence to sequence. The sequence length can vary from >5 to < length of genome length. Hence length attribute can act as an outlier. Standard scaler is used to scale attributes instead of Min-Max Normalization. Because in Min-Max normalization will normalize all attributes to the range (0,1). It will not able to predict good enough in the case where there are 1000's of records (4385 records are present in this dataset).

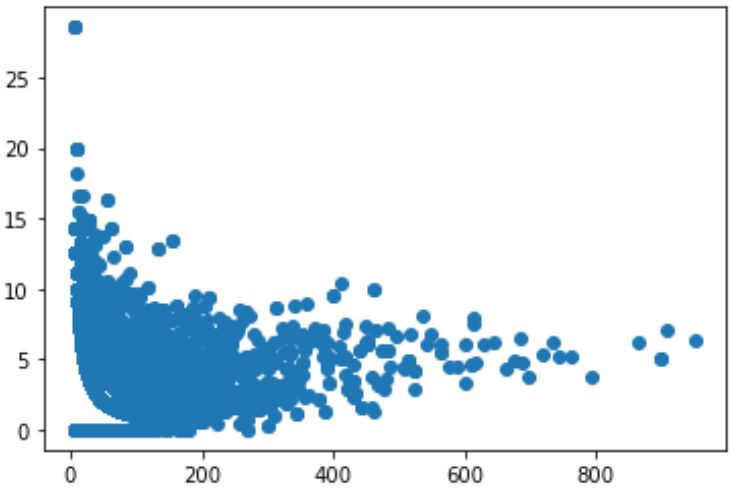
Fig 6.1 Scatter Plot of Raw Data

X-axis – Length of Sequence

Y-axis – GC Percentage

Range of Length (6, 949)

Range of GC Percentage (0, 7)



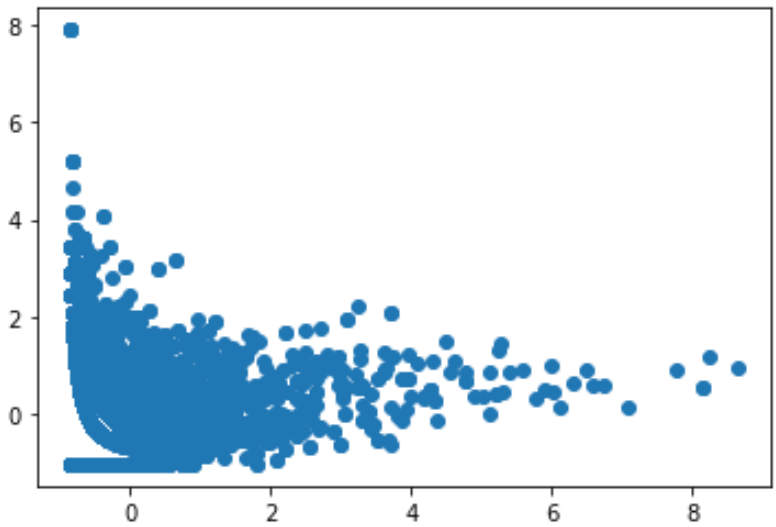


Fig 6.2 Scatter Plot after Standard Normalization

X-axis – Length of Sequence

Y-axis – GC Percentage

Range of Length (-0.8575, 8.6493)

Range of GC Percentage (-1.01, 7.89)

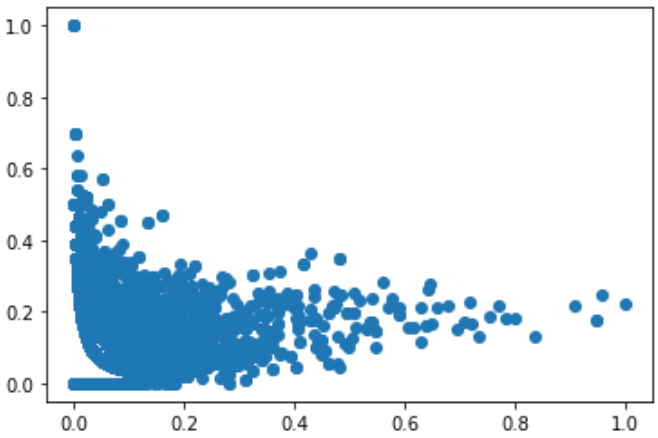
Fig 6.3 Scatter Plot after Min-Max Normalization

X-axis – Length of Sequence

Y-axis – GC Percentage

Range of Length (0, 1)

Range of GC Percentage (0, 1)



- **Representing our data**

Skewness in data

If skewness is not close to zero, then your data set is not normally distributed. If skewness is 0, the data are perfectly symmetrical, although it is quite unlikely for real-world data.

- If skewness is less than -1 or greater than 1, the distribution is highly skewed.
- If skewness is between -1 and -0.5 or between 0.5 and 1, the distribution is moderately skewed.
- If skewness is between -0.5 and 0.5, the distribution is approximately symmetric

Length	2.840213
GCPercentage	1.538186
ATG	-0.478011
A	0.613407
C	0.378739
G	0.560587
T	0.482186
AA	-0.455371
AC	1.653317
AG	2.404432
AT	1.089684
TA	1.530116
TC	1.631491
TG	1.720201
TT	-0.442105
CA	0.550027
CC	0.497133
CG	2.564767
CT	0.931863
GA	0.831523
GC	1.644941
GG	0.619589
GT	1.186909
Coding_Region	-0.198529

Fig 7.1 Skew measurement of attributes

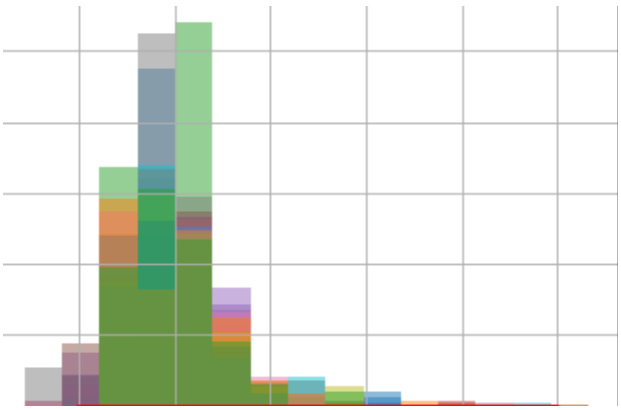
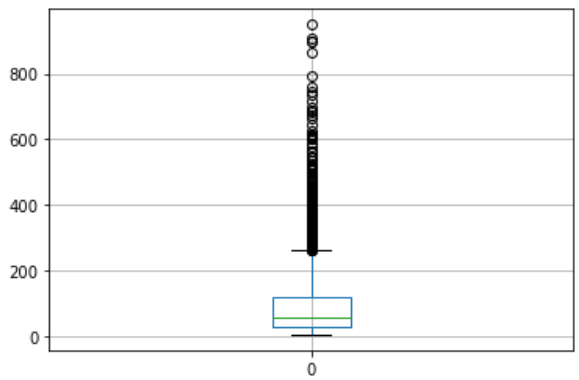


Fig 7.2 Histogram plot after Normalizing the data. Skewness in data is adjusted now.

Box Plot



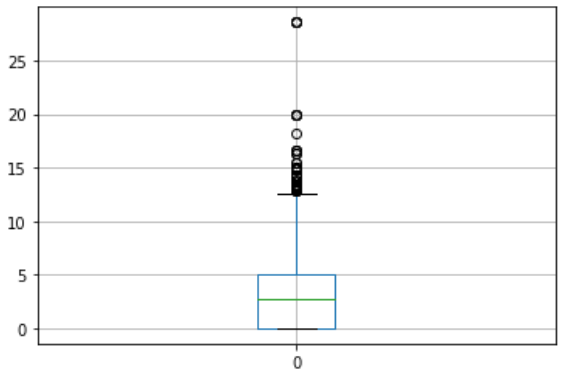


Fig 8.1 Box plot of the length attribute.

Fig 8.2 Box plot of GC Percentage attribute.

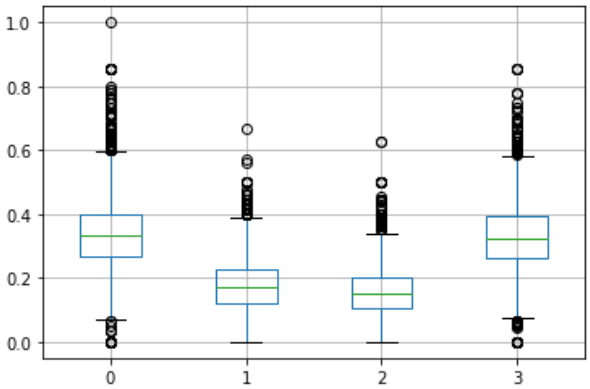


Fig 8.3 Box Plot of A, T, G, C Nucleotide frequencies.

Correlation

Correlation helps us to find dependencies of attributes. It ranges from -1 to 1. 1 indicates it is strongly related, -1 indicates badly related. A positive correlation indicates if one variable increases other increase vice versa. A negative correlation indicates if one variable decreases other increases vice versa.

	0	1
0	1.000000	0.104461
1	0.104461	1.000000

Fig 9.1 Correlation matrix between Length and GC Percentage

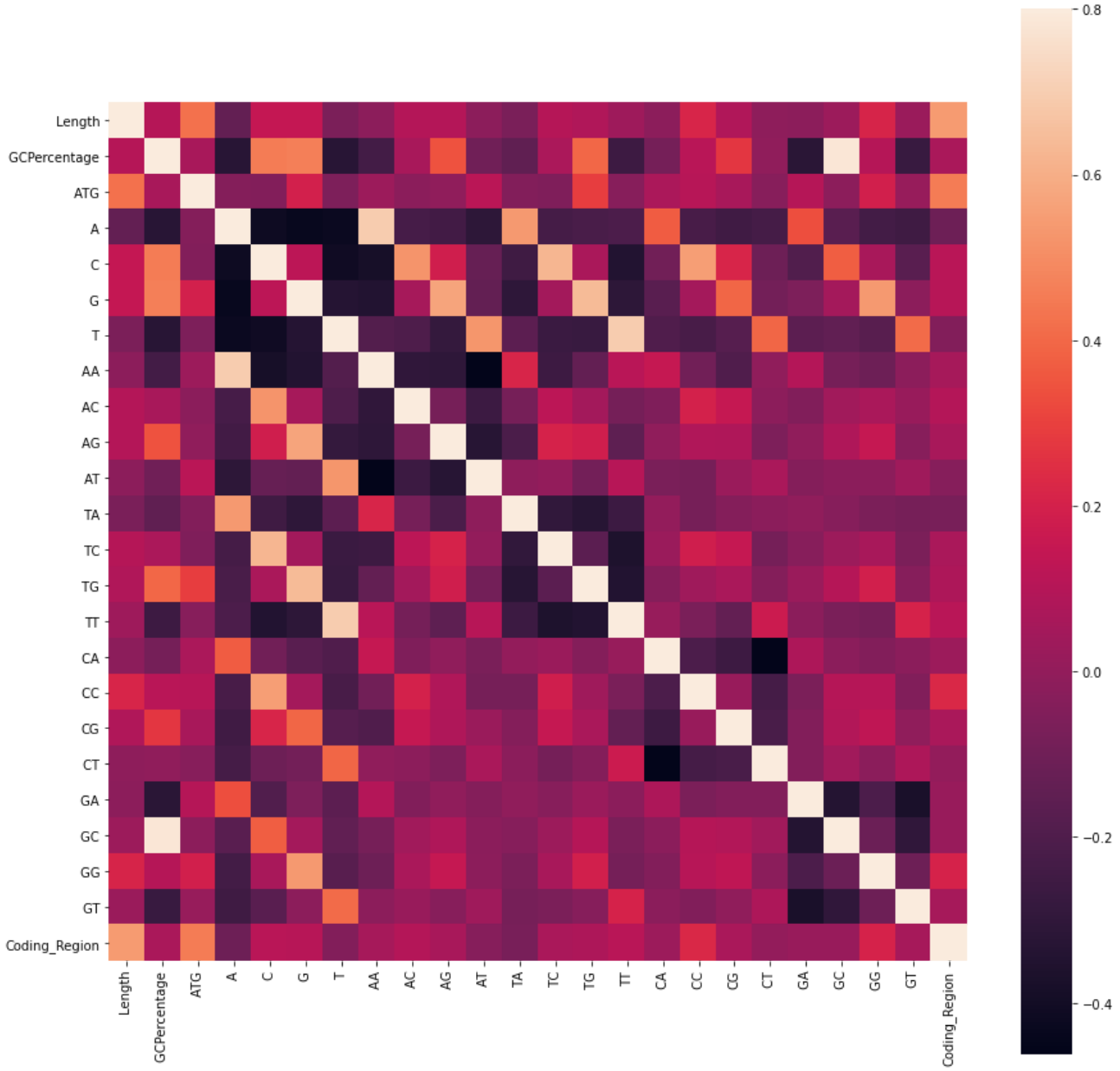


Fig 9.2 Heat Map representing the correlation between Attributes. GC Percentage and GC Probability attributes are highly correlated.

- # Models

Initially, we build our dataset without probabilities of nucleotides and Then we trained our data with different predictive algorithms, we got an accuracy of around 80-86 percent. So, the dataset was remodeled with nucleotide frequencies. Where it became easy to identify the pattern for the algorithm. Surprisingly we got a high accuracy of 97.035 with XGBoost, 96.5792 with Random Forest and then for Decision Tree with 94.5267.

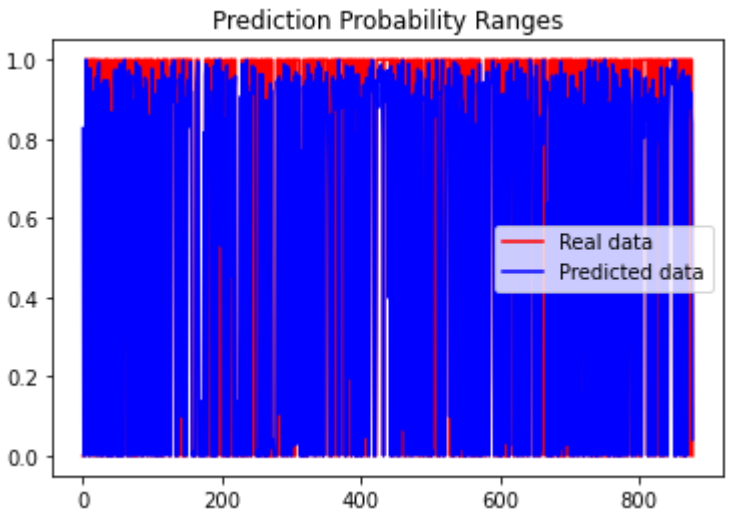
The outcome variable is separated from the data frame. 23 variables are considered for prediction. The outcome variable is categorical either 1 or 0. 1 indicates that the segment of a sequence can be in the coding region. 0 indicates the sequence is not found in the coding region. A standard scaler was used to scale the attributes.

Entire data is split into training and testing data. 20% of the entire data is considered as testing data. 3508 records for training and 877 records are used for testing. Various classification algorithms are used for prediction. The following are the details of the model and its accuracy.

Results

Sno	Model Name	Accuracy	Mean Accuracy from K-fold cross-validation	Standard Deviation of accuracies from cross-validation
1.	Logistic Regression	90.30	88.511	0.0113
2.	Decision Tree	94.52	94.041	0.0133
3.	Random Forest	96.57	95.495	0.0082
4.	Naïve-Bayes	85.74	85.661	0.0157
5.	SVC	90.42	89.196	0.0116
6.	XGBoost	97.035	95.865	0.0077
7.	Cat boost	93.50	92.303	0.0114
8.	Artificial Neural Network	93.38	-	-

Fig 10.0 Probability graph from the predictions of Artificial Neural Network. The artificial neural network will give the probability that a sequence can come under category 1 or 0.



Evaluation of Models

Random Forest

- **Precision and Recall Measures**

  Confusion matrix for Random Forest

For Random Forest	Classified Positive	Classified Negative
Actual Positive	**367**	**18**
Actual Negative	**12**	**480**

Accuracy: = 0.9657

Precision(P): = 0.9683

Recall(r): = 0.9532

- Value(– Score)

=

=0.9606

d(i, j) symmetry = = 0.0342

d(i, j) asymmetric = = 0.0755

Jaccard coefficient = = 0.92443

XGBoost

- **Precision and Recall Measures**

  Confusion matrix for Random Forest

For XGBoost	Classified Positive	Classified Negative
Actual Positive	**368**	**17**
Actual Negative	**9**	**483**

Accuracy: = 0.9703

Precision(P): = 0.9761

Recall(r): = 0.9558

- Value(– Score)

=

= 0.9658

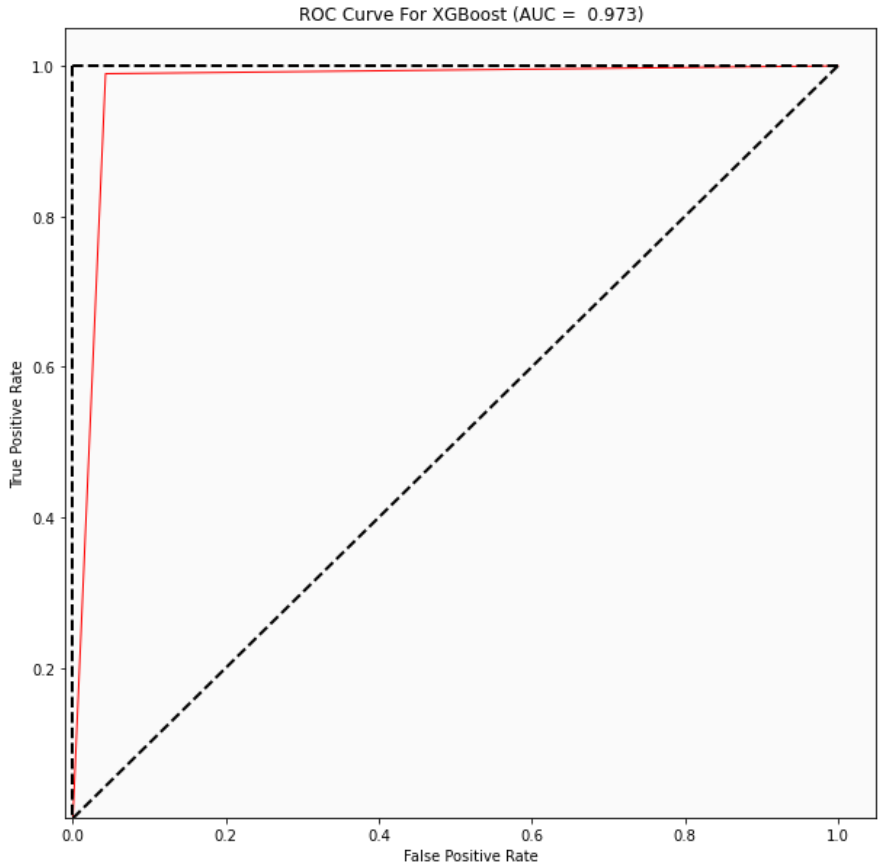
d(i, j) symmetry = = 0.0296

d(i, j) asymmetric = = 0.06598

Jaccard coefficient = = 0.9340

ROC CURVE and AUC

RECEIVER OPERATING CHARACTERISTIC CURVE. It is a plot between TPR (True Positive Rate) and FPR (False Positive Rate). It is a probability curve, where TPR is on X-axis and FPR on Y-axis. **AUC (**Area Under Curve**)** Through this we can understand how perfectly it is classifying classes. It is used for Binary Classification.



TPR = = 0.9761

FPR = = 0.024

Fig 10.1 ROC Curve and AUC. If AUC is 0.5 it is very bad model. If it is 1, the performance of classifier is pretty good.

Discussion and Conclusion

After observing each model XGBoost gave the best accuracy of 97.035 which is pretty good, the artificial neural network for 100 epochs gave an accuracy of 93.38. Cat Boost for 10 iterations it gave 93.50 whereas for 1000 iterations it gave 96.45 but, there will be overfitting of data. XGBoost and Cat Boost are Gradient Boosting frameworks where parallel boosting takes place

We clustered the entire dataset and our observation found 5 clusters. Cluster 1 consists of the entire non-coding sequences. But Cluster 2,3,4,5 contains different sequences coding for different proteins.

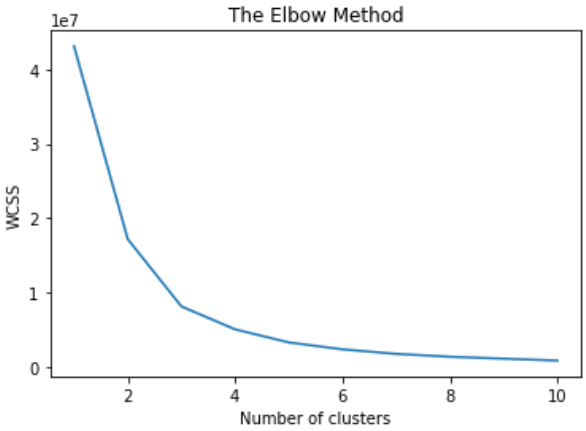


Fig 10.1 Elbow Method, to find no. of clusters can be formed from the dataset

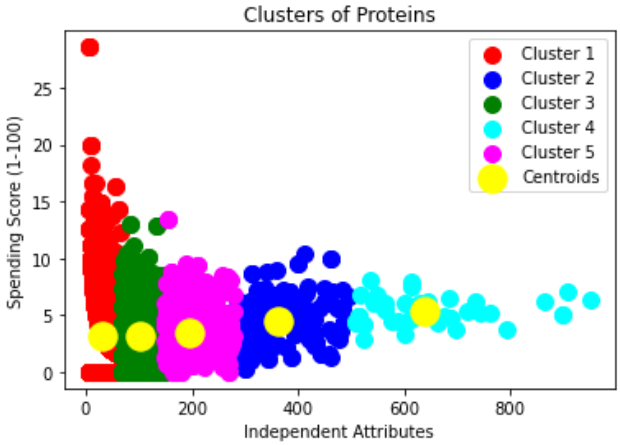


Fig 10.2 Five Clusters. Yellow dots are the centroids of the clusters

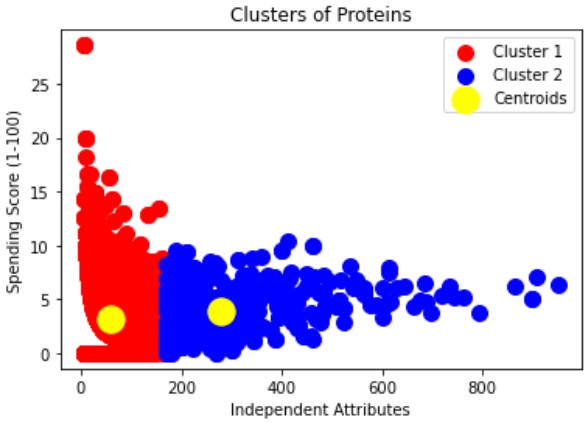


Fig 10.3
Two clusters are formed. Where cluster 1 consists most of non-coding sequences. Cluster 2 consists most of coding sequences.

Now, I took an **ermC gene** of **Staphylococcus aureus**. It is experimentally determined that it is a gene. It will present in the coding region. Following is the sequence of the gene.

Note: In the preparation of dataset **Staphylococcus aureus** genome is not used.

```
>NC_001395.1:c1722-988 Staphylococcus aureus strain T48 plasmid pT48, complete
sequence
ATGAACGAGAAAAATATAAAACACAGTCAAAACTTTATTACTTCAAAACATAATATAGATAAAATAATGA
CAAATATAAGATTAAATGAACATGATAATATCTTTGAAATCGGCTCAGGAAAAGGCCATTTTACCCTTGA
ATTAGTACAGAGGTGTAATTTCGTAACTGCCATTGAAATAGACCATAAATTATGCAAAACTACAGAAAAT
AAACTTGTTGATCACGATAATTTCCAAGTTTTAAACAAGGATATATTGCAGTTTAAATTTCCTAAAAACC
AATCCTATAAAATATTTGGTAATATACCTTATAACATAAGTACAGATATAATACGCAAAATTGTTTTTGA
TAGTATAGCTGATGAGATTTATTTAATCGTGGAATACGGGTTTGCTAAAAGATTATTAAATACAAAACGC
TCATTGGCATTACTTTTAATGGCAGAAGTTGATATTTCTATATTAAGTATGGTTCCAAGAGAATATTTTC
ATCCTAAACCTAAAGTGAATAGCTCACTTATCAGATTAAATAGAAAAAAATCAAGAATATCACACAAAGA
TAAACAGAAGTATAATTATTTCGTTATGAAATGGGTTAACAAAGAATACAAGAAAATATTTACAAAAAAT
CAATTTAACAATTCCTTAAAACATGCAGGAATTGACGATTTAAACAATATTAGCTTTGAACAATTCTTAT
CTCTTTTCAATAGCTATAAATTATTTAATAAGTAA
```

Prediction by our model (using XGBoost)

```
#Test case with a gene sequence

seq = readseq('seq.txt')
print("Length of the sequence:",len(seq))

result = prediction_XGBoost(seq)
if result[0] == 1:
    print("The given Sequence will present in coding region")
else:
    print("The given Sequence will present in non-coding region")
```

```
Length of the sequence: 735
The given Sequence will present in coding region
```

Fig 10.4 Results from a test sequence. It predicted as the sequence can be in the coding region.

References

NCBI	**https://www.ncbi.nlm.nih.gov/**
UniProt	https://www.uniprot.org/

END