19CSE304 FOUNDATIONS OF DATA SCIENCE

NOVEMBER 2021

CODON USAGE

PREPARED AND PRESENTED BY

AKKINA SAIRAMSRIKAR - BL.EN.U4CSE19004
K SUKHA - BL.EN.U4CSE19059
TRISHA CHANDER - BL.EN.U4CSE19133

ABOUT CODON USAGE DATA SET

A LITTLE BIT ABOUT OUR DATASET

The DNA consists of four amino acids: **C**,**T**,**U** and **G**. Codon Usage frequencies in the genomic coding DNA of a large sample of diverse organisms were examined. 514 different genus names are present.

The characteristic of the dataset is ***multivariate.*** The task associated with the dataset includes ***classification*** and ***clustering***. There are **13028 instances** and **69 attributes**.

ATTRIBUTE OVERVIEW

ATTRIBUTE NAME	
COLUMN 1 KINGDOM	**COLUMN 4** N CODONS
COLUMN 2 DNA TYPE	**COLUMN 5** SPECIES NAME
COLUMN 3 SPECIES ID	**COLUMN 6 - 69** CODON - NUCLEOTIDE BASES

ATTRIBUTE INFORMATION

The **Kingdom** is a 3 letter code corresponding to the different kingdoms in the form 'xxx' which stands for: 'arc'(archaea), 'bct'(bacteria), 'phg'(bacteriophage), 'plm' (plasmid), 'pln' (plant), 'inv' (invertebrate), 'vrt' (vertebrate), 'mam' (mammal), 'rod' (rodent), 'pri' (primate), and 'vrl'(virus) sequence entries.

The **DNA type** is an integer from 0 to 12 denoting the genomic composition in species: 0-genomic, 1-mitochondrial, 2-chloroplast, 3-cyanelle, 4-plastid, 5-nucleomorph, 6-secondary_endosymbiont, 7-chromoplast, 8-leucoplast, 9-NA, 10-proplastid, 11-apicoplast, and 12-kinetoplast.

Species ID is an integer to uniquely indicate the entries of an organism.

Ncodons (number of codons) is the algebraic sum of the numbers listed for different codons in an entry of CUTG. This is basically the codon frequency.

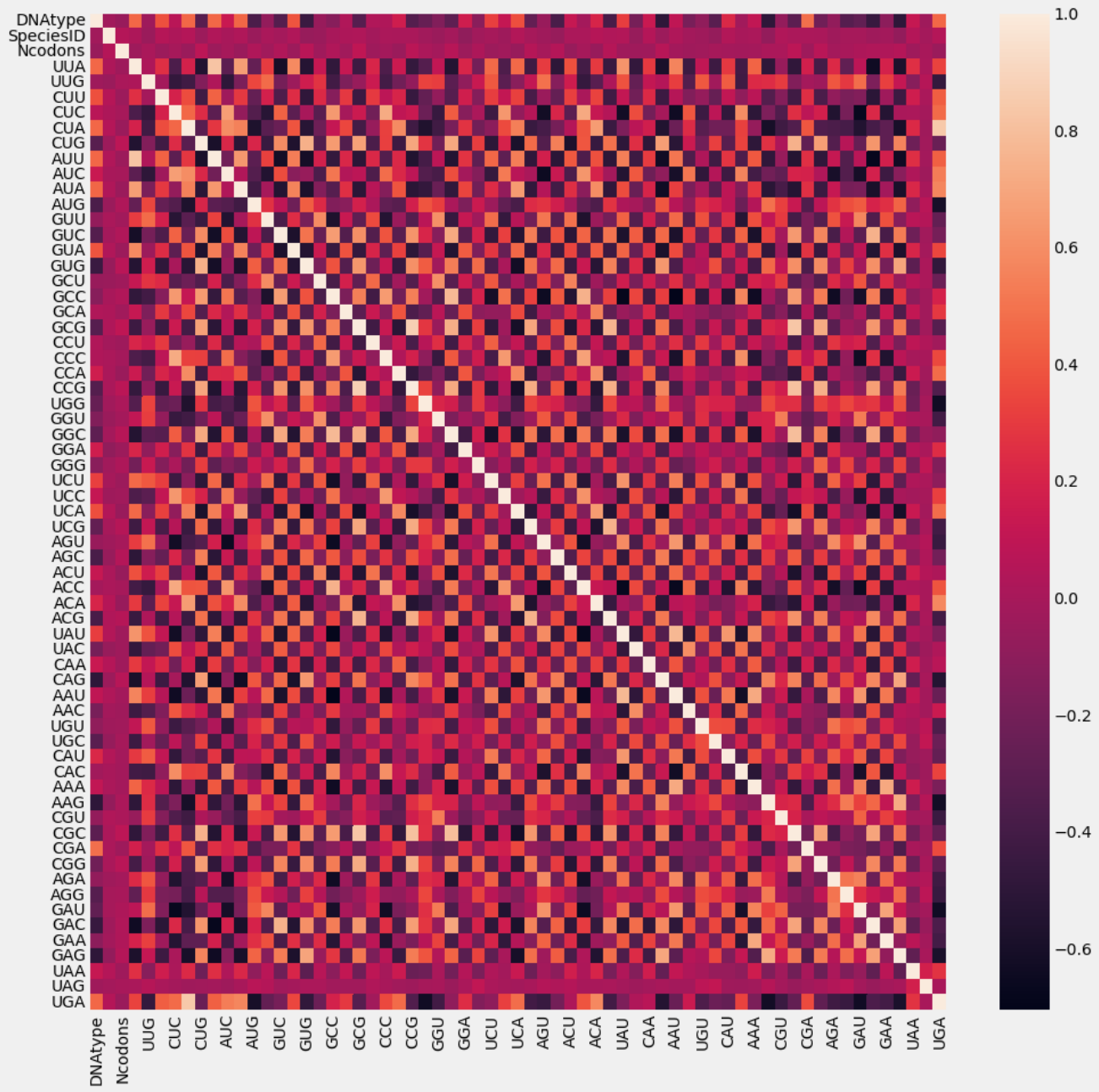
Species Name is a descriptive feature which is the name of the species.

Codon Frequencies are values like 'UUU', etc which are float values.

SCOPE OF OUR WORK

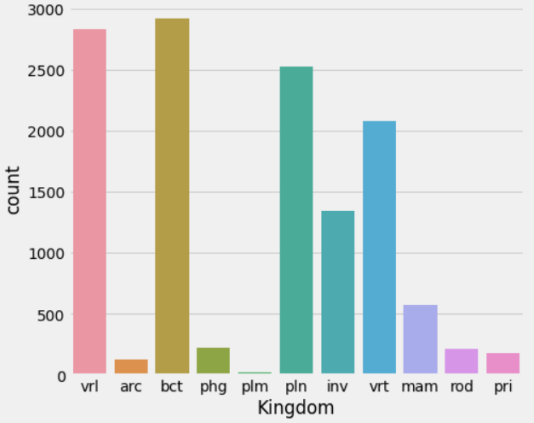
We've cleaned the data and preprocessed it before running machine learning algorithms to predict Species ID and Kingdom. Now you may be wondering how this is of use.
Codon usage is an important usage of gene expression. Gene expression, simply put, is extremely useful for identifying how the molecules of diseases like cancer, etc are structured. It tells us how a specific cell is functioning at a specific type.
Another use is a model that is created to predict a dna or kingdom can be used to determine which kingdom a newly discovered species belongs to just from its DNA.

VISUALISATIONS



This is a correlation matrix between the attributes represented as a heatmap.

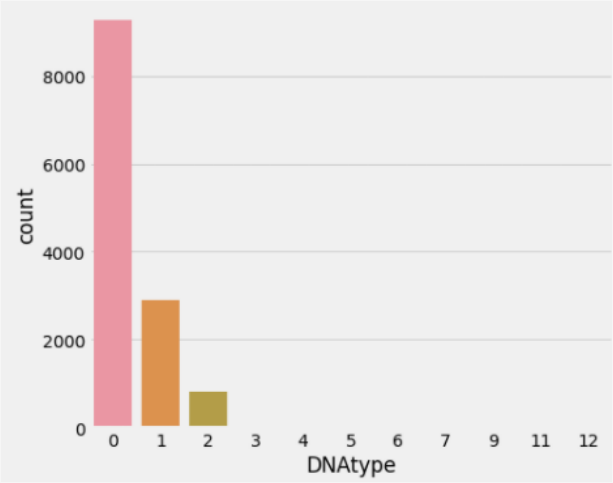
VISUALISATIONS



KINGDOM VS COUNT

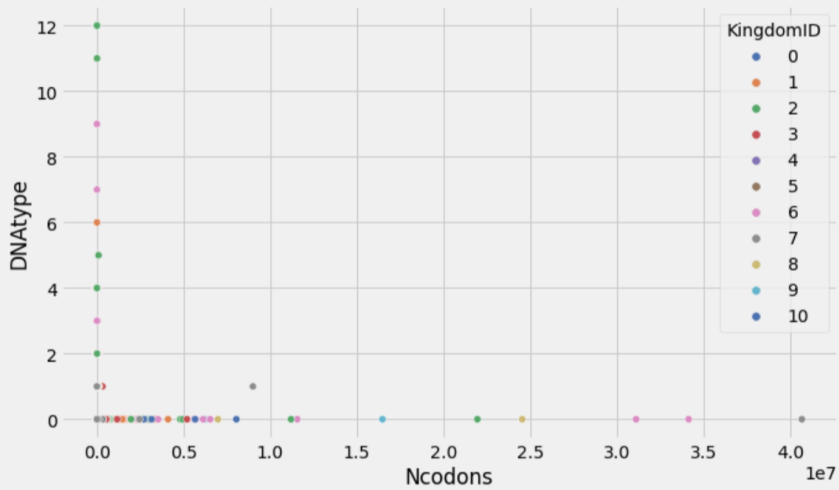
This is a countplot from seaborn. It shows the count of the types of kingdoms. **bct** has the highest count followed by **vrl**. **plm** has the least count followed by **arc**.

This is a countplot of DNA type. There are 13 DNA types from 0 to 12. Given the size of the dataset, the most frequent DNA types are 0,1 or 2. The other DNA types, although present, are relatively lower hence its count is not reflected in the plot.



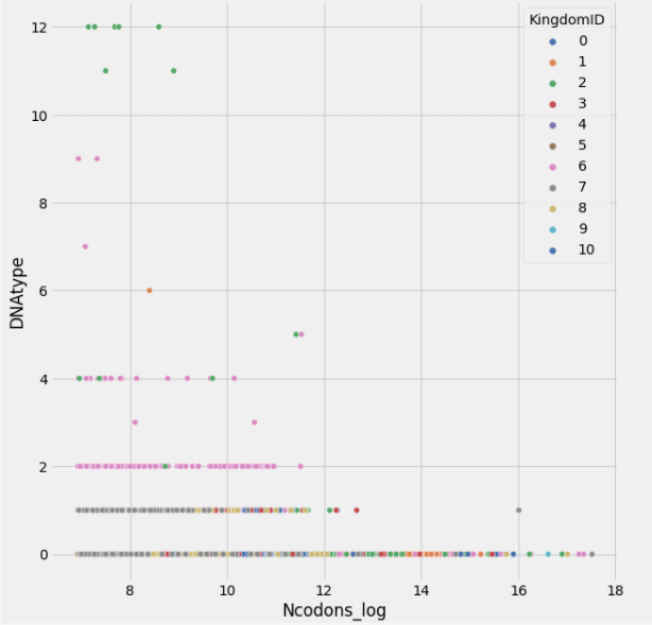
DNATYPE VS COUNT

VISUALISATIONS



N CODONS VS DNA TYPE

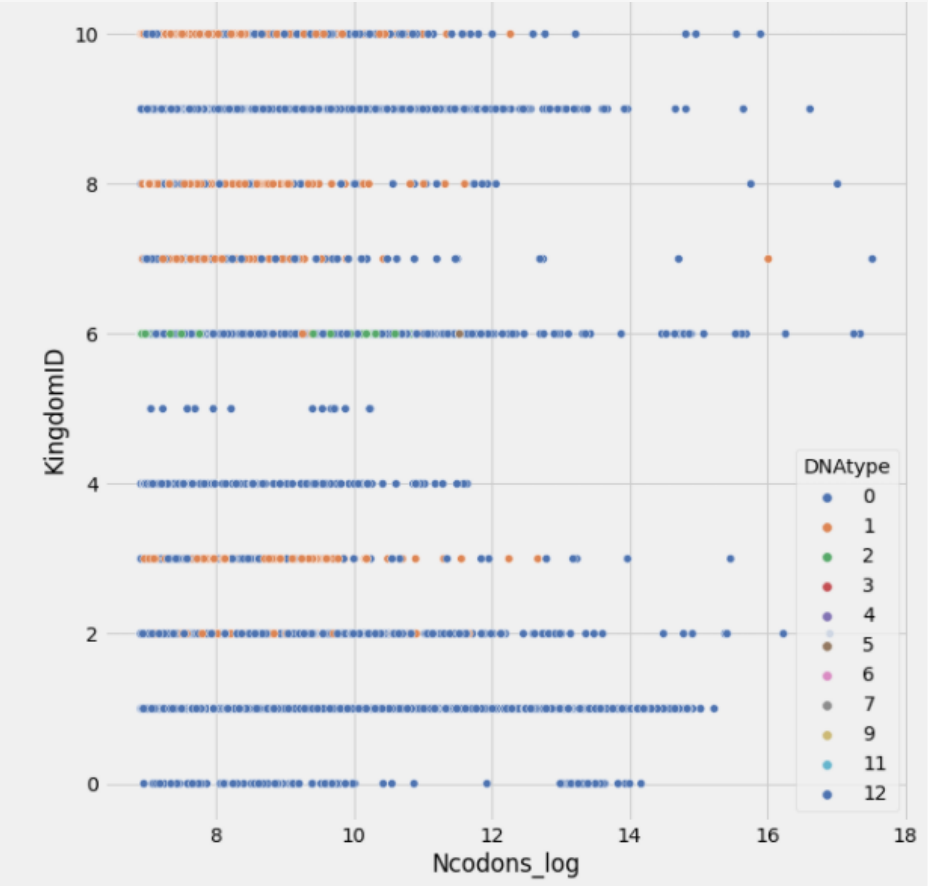
A scatter plot was created as Ncodons vs DNA type. This was not decipherable. Hence, the log of the Ncodons were taken and once again a scatter plot against the DNA type was visualised. After label encoding **Kingdom** attribute, we formed **KingdomID**,



N CODONSLOG VS DNA TYPE

VISUALISATIONS

By label encoding the **Kingdom** attribute, we engineered a new feature, **KingdomID**, Once again we plot the N CODONS LOG against this to obtain a far better scatterplot.

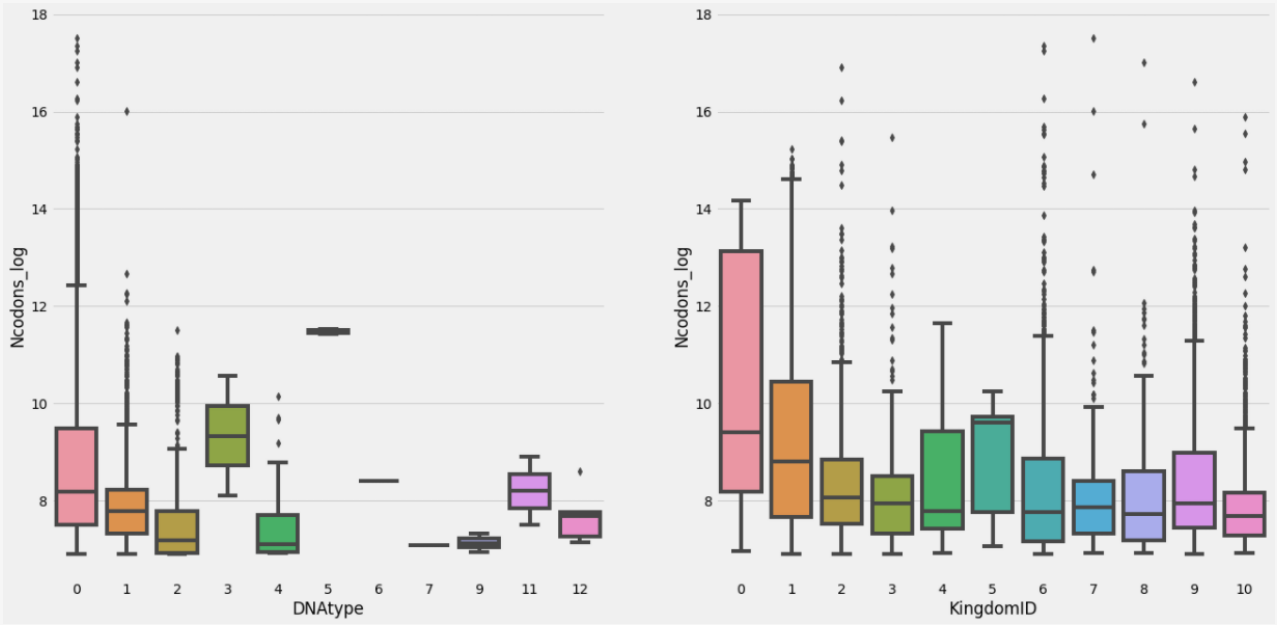


N CODONSLOG VS KINGDOM ID

At a glance, DNA type 0,1 and 2 are the ones that are visible and predominantly cover the plot. Kingdom ID 0,1,2,4,5,6 and 11 are mostly, entirely in some cases, of DNA type 0. Majority of Kingdom ID 3,7,8 and 10 is of DNA type 1. Traces of DNA type 2 can be found in Kingdom ID 6.

VISUALISATIONS

Now coming to boxplots, we have plotted two: DNA type VS N codons log and Kingdom ID VS N codons log.



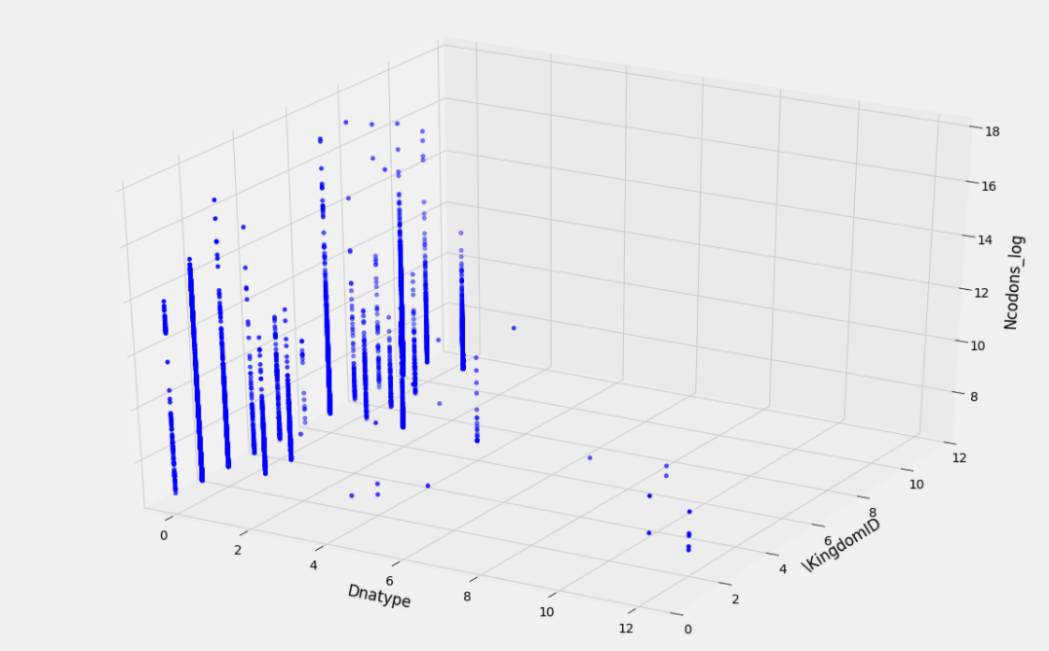
1A 1B

BOXPLOTS:
1A - DNA TYPES VS NCODONS_LOG
1B - KINGDOM ID VS NCODONS_LOG

The readings of the box plots are more understandable in 1B as the data is not enough in 1A. Some of the attributes have been hidden. DNAtype 0 has the highest range and also the most number of outliers.

In the second boxplot, there are more outliers for most Kingdom IDs. The range of kingdom id 0 is the most followed by kingdom id 1. Kingdom ID 5 has the highest mean.

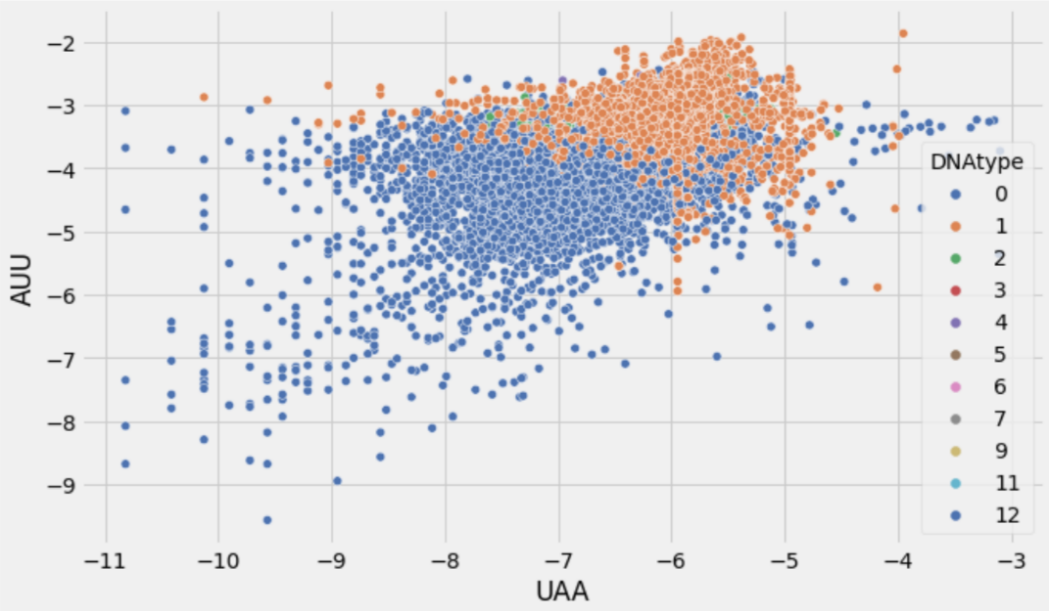
VISUALISATIONS



3D SCATTER PLOTS

This visualization was done to experiment with the data. The data for DNA type decreases drastically as you go along that axis, in the sense, DNA type 0 has the most information. The data for Kingdom ID is pretty much consistently present as you go along its axis. The further up you go along the Ncodons_log axis, the more outliers are found with respect to Kingdom ID.

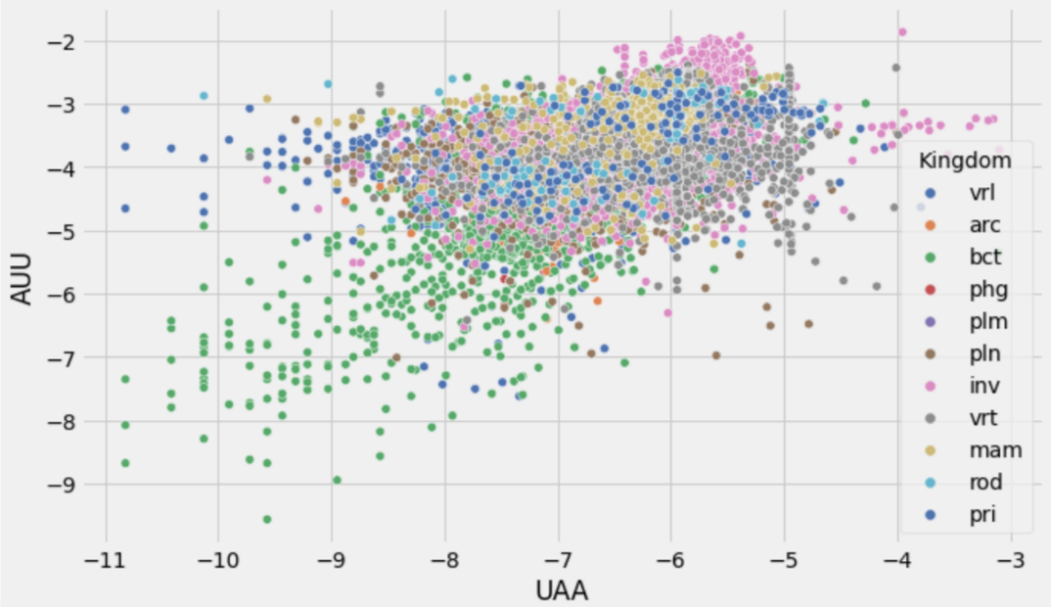
VISUALISATIONS



2D SCATTER PLOT
'UAA' AND 'AUU'

'UAA' and 'AUU' are two codon frequencies plotted against each other here. Most of the types present here are either of DNA type of 0 or 1. Upon taking a closer look, DNA type 2, although sparse, is visible as well. The ranges across the DNA type can be noted. DNA type 1 occurs more on the higher part of the plot and DNA type 0 occurs more towards the bottom. This visualization again was only for experimenting.

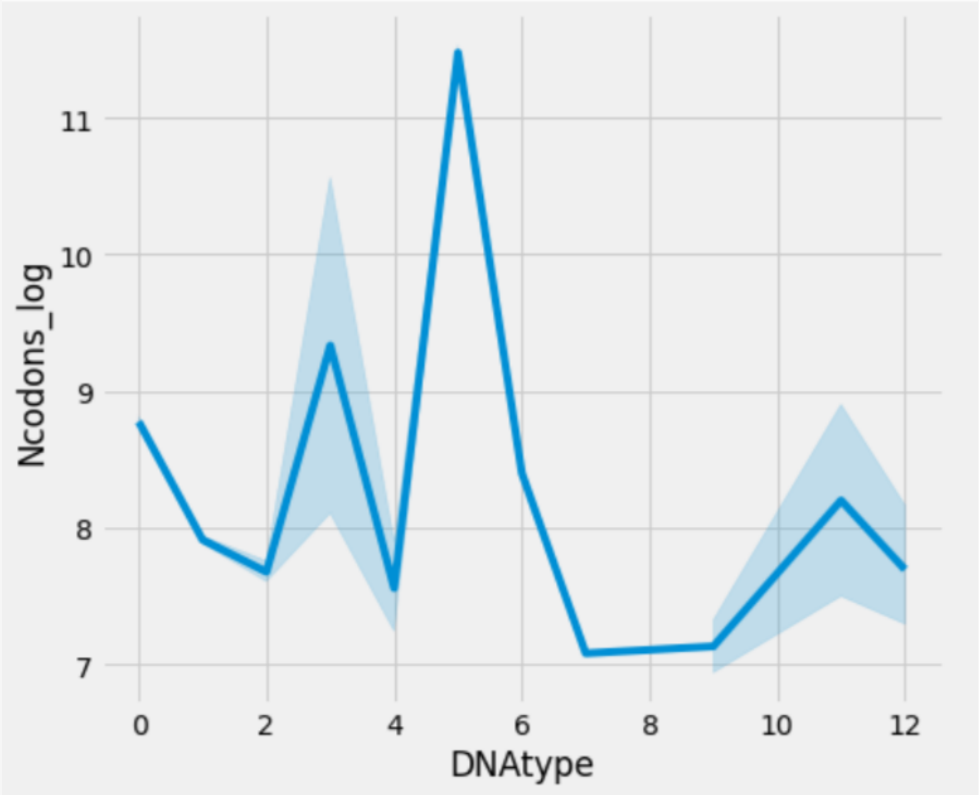
VISUALISATIONS



2D SCATTER PLOT
'UAA' AND 'AUU'

Setting hue as Kingdom, far more information can be observed with this plot as compared to the previous one. 'bct' which is green, for instance, covers the lower left side of the plot more. Most of the Kingdoms are found between 'UAA' value -9 to -4.5 and 'AUU' value -6 to -2.

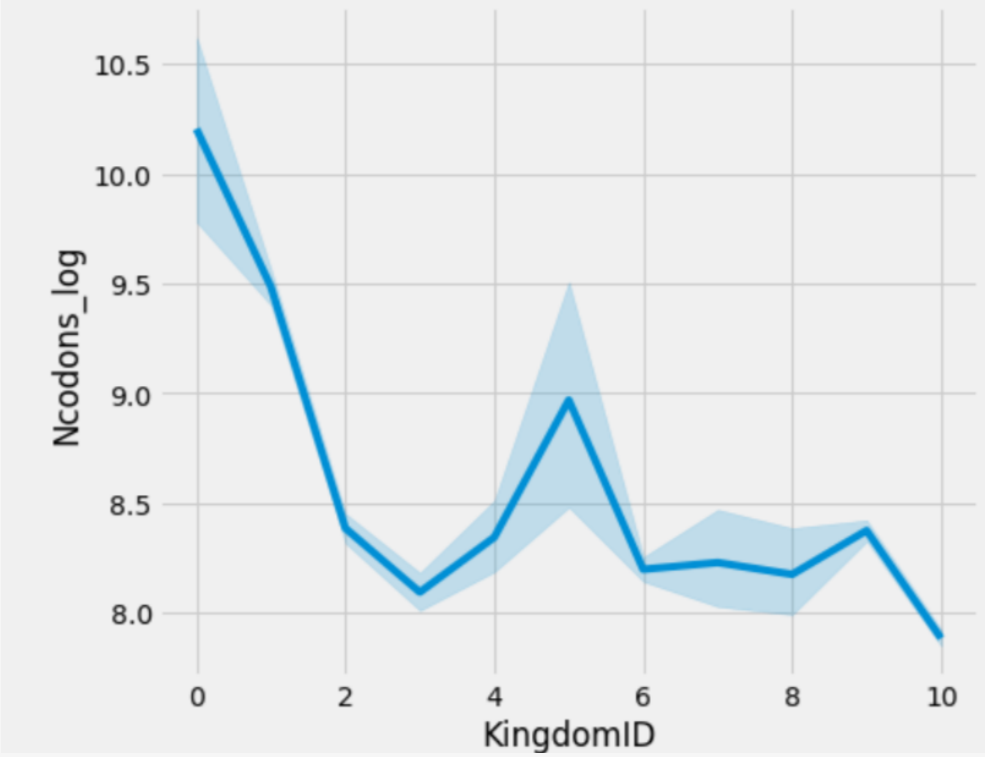
VISUALISATIONS



LINEPLOTS - DNA TYPE VS NCODONS_LOG

The above lineplot indicates that DNA type 5 has its maximum peak value of Ncodons_log. Its lowest point is for DNA type 7 at an Ncodons_log value that is a little greater than 7.

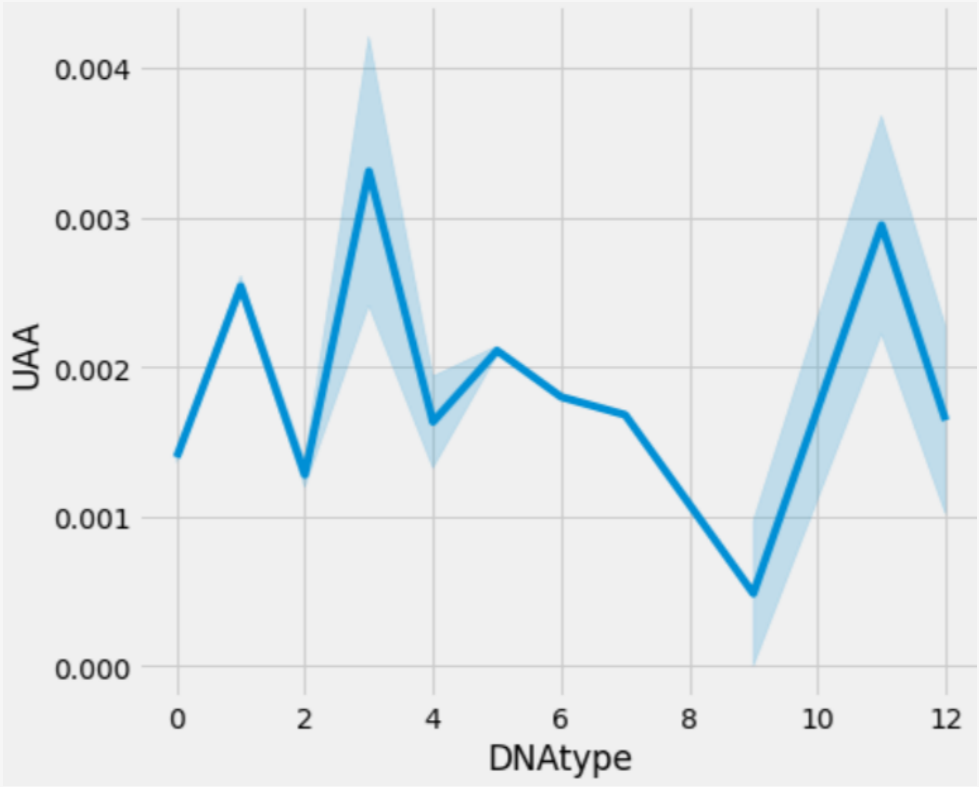
VISUALISATIONS



LINEPLOTS - KINGDOM ID VS NCODONS_LOG

The overall trend of this lineplot of Kingdom ID vs Ncodons_log is decreasing. Its peak is when Kingdom ID is 0 and the lowest point is for the Kingdom ID 10. The trend increases for Kingdom ID 5 then decreases again.

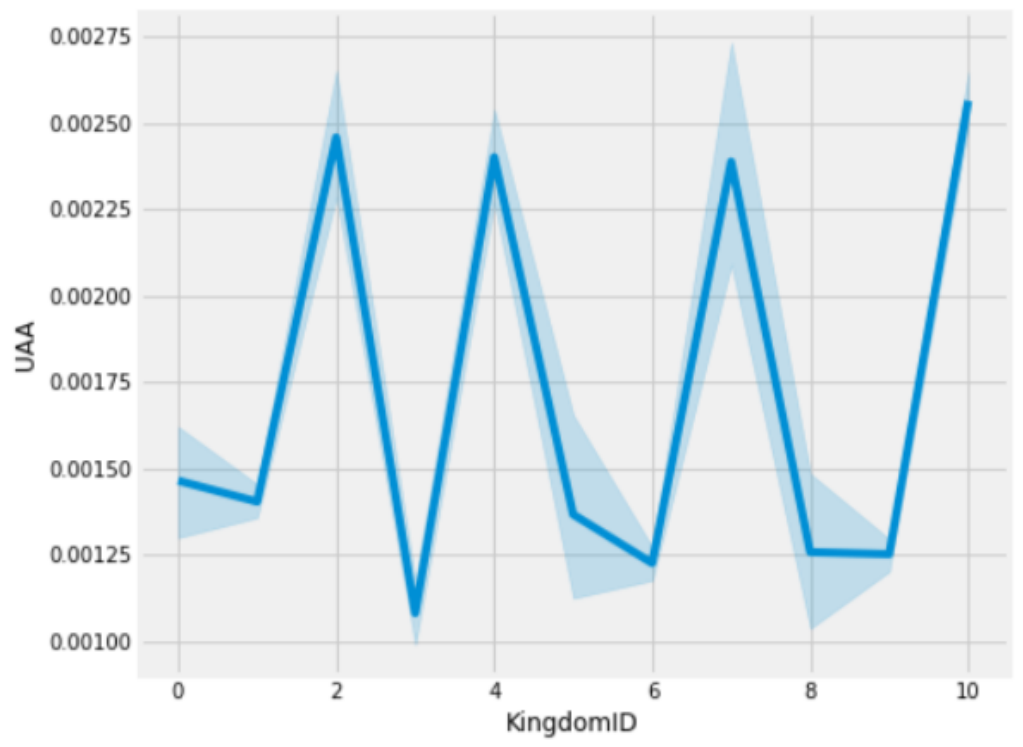
VISUALISATIONS



LINEPLOTS - DNA TYPE VS 'UAA'

For the following two lineplots, we have taken a codon sequence 'UAA'. The above lineplot shows the DNA type plotted against the sequence 'UAA'. 'UAA' occurs more in DNA type 3 and occurs least for DNA type 3.

VISUALISATIONS



LINEPLOTS - KINGDOM ID VS 'UAA'

The above lineplot shows the Kingdom ID plotted against the sequence 'UAA'.Looking at the overall graph, the line oscillates up and down, giving three peaks that have almost the same value. The final peak is the highest having a value a little more than 0.00225 for Kingdom ID 10.

MACHINE LEARNING ALGORITHMS

THIS SECTION INCLUDES THE METHODS WHICH WE HAVE USED TO PREDICT THE KINGDOM OF SPECIES

There are several machine learning algorithms that can be used for classification, but according to the best classifier paper, from the University of Santiago, Italy, which says SVM and decision trees are the best algorithms in general with no prerequisites.
In the case of decision trees, since they are highly prone to overfitting, we have used Random forests as a solution to avoid overfitting, in the datasets. Also, our data is huge in terms of computation, which takes a long time for a model to predict/ train using it, thus we have used PCA to reduce the data into principal components and use a small subset of it to train our model.

SUPPORT VECTOR MACHINE

RESULTS

In [35]:
```
svm_model = svm.SVC()
ten_cross_validation = cross_val_score(svm_model,
                                       data_scale,
                                       data_cleaned['KingdomID'],
                                       cv = 10 #this is the number of cross validations which are to be performed
                                      )
print(ten_cross_validation)
print(np.mean(ten_cross_validation))
```

```
[0.90867229 0.92555641 0.92325403 0.93323101 0.92785879 0.90636992
 0.93241167 0.92626728 0.92242704 0.91858679]
0.9224635220859814
```

As the above cell shows, the score obtained is 0.9224, which is quite a good accuracy for a machine learning model. And since the score is not too high, we can safely state that the model is not overfitting

10 FOLD CROSS VALIDATION

In [37]:
```
list_output = []
for i in range(1,25):


    pca_model = PCA(n_components = i)
    pca_dataset = pca_model.fit_transform(data_scale)

    values, mean = svm_model_function(pca_dataset, data_cleaned['KingdomID'])
    list_output.append(mean)
```

ACCURACY VS VAL ACCURACY

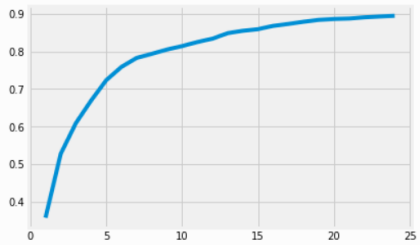
SUPPORT VECTOR MACHINE

RESULTS

```
In [38]: list_output
Out[38]: [0.3575156822315983,
          0.5274773564019226,
          0.6093927165598,
          0.6692734361092739,
          0.7237026571081977,
          0.7590177105179705,
          0.7829656364315835,
          0.7938678672518694,
          0.8053076145914014,
          0.8145203141044005,
          0.8252669899192812,
          0.8342495694091268,
          0.8489146516428472,
          0.8552880449582849,
          0.8595867034953015,
          0.8684152015966934,
          0.8734826755696709,
          0.8793947088899184,
          0.8843076299170176,
          0.8866882286299017,
          0.8879162820526425,
          0.891217301913462,
          0.8933668964330217,
          0.8951322895409742]
```

RESULTS AFTER PCA

```
In [39]: sns.lineplot(y = list_output,
                      x = [i for i in range(1, 25)])
Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x7ffaba4c1390>
```



Looking at the graph, we can analyse that the curve somewhat looks like a **logarithmic** graph, while using the PCA attribute reduction mechanism.

Even though the maximum value that we can get with addition of all the attributes that is about 65 of them, we get a score of about 0.92, and just by includin 25 of them we will get an accuracy of 0.8964, which is a pretty good accuracy considering the fact that we are using less than half of the original attributes.

PCA GRAPH

DECISION TREE

RESULTS

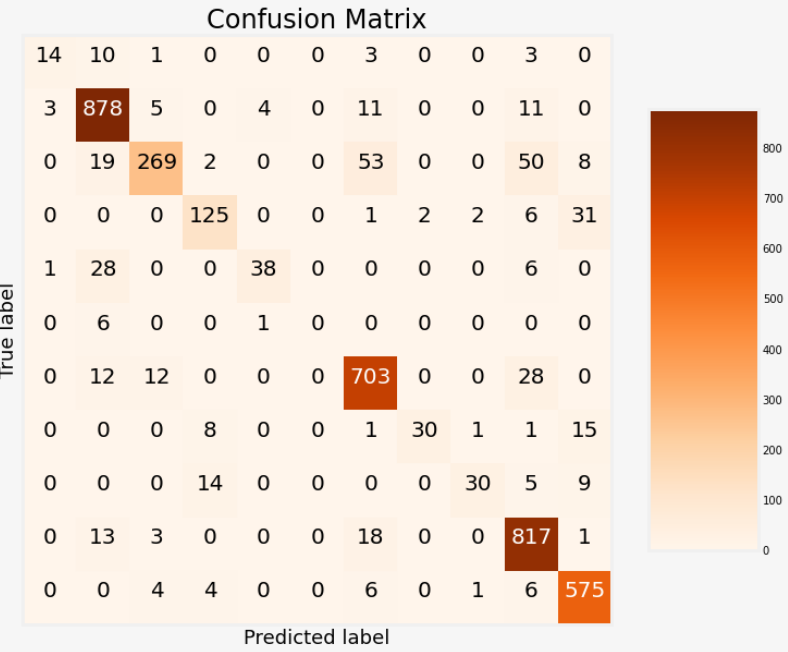
```
n_nodes = []
max_depths = []

for ind_tree in model.estimators_:
    n_nodes.append(ind_tree.tree_.node_count)
    max_depths.append(ind_tree.tree_.max_depth)

print(f'Average number of nodes {int(np.mean(n_nodes))}')
print(f'Average maximum depth {int(np.mean(max_depths))}')
```

```
Average number of nodes 2119
Average maximum depth 23
```

AVERAGE NODES



CONFUSION MATRIX

```
print(classification_report(y_test,rf_predictions))
```

```
              precision    recall  f1-score   support

           0       0.78      0.45      0.57        31
           1       0.91      0.96      0.94       912
           2       0.91      0.67      0.77       401
           3       0.82      0.75      0.78       167
           4       0.88      0.52      0.66        73
           5       0.00      0.00      0.00         7
           6       0.88      0.93      0.91       755
           7       0.94      0.54      0.68        56
           8       0.88      0.52      0.65        58
           9       0.88      0.96      0.92       852
          10       0.90      0.96      0.93       596

    accuracy                           0.89      3908
   macro avg       0.80      0.66      0.71      3908
weighted avg       0.89      0.89      0.88      3908
```

CLASSIFICATION REPORT

NEURAL NETWORKS

WE ARE TRYING TO PREDICT DNA TYPE WITH THE HELP OF NEURAL NETWORKS

Neural networks are amazing at classifying with a high amount of features.we used adam optimizer,ran upto 30 epochs with batch size of 64.we added early stopping feature with min mode to avoid overfitting the model. after ran 30 epochs validation accuracy ends at 98.37% and accuracy ends at 98.35%. results are pretty good but major classes falls under DNA type (0,1,2) due to very less data for remaining DNA types. solution is we have to collect more data or create a new target and pull all classes from 3-10 in that new target

65 / TANH

128 / TANH

64 / RELU

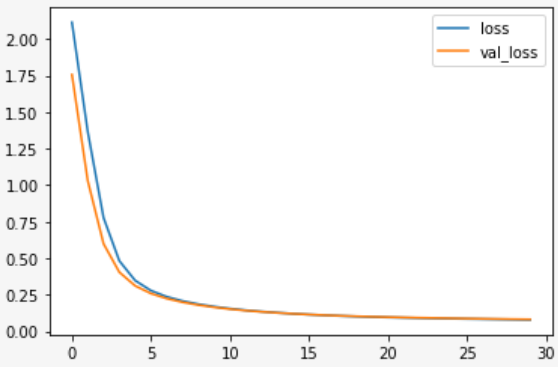
32 / RELU

11 / softmax

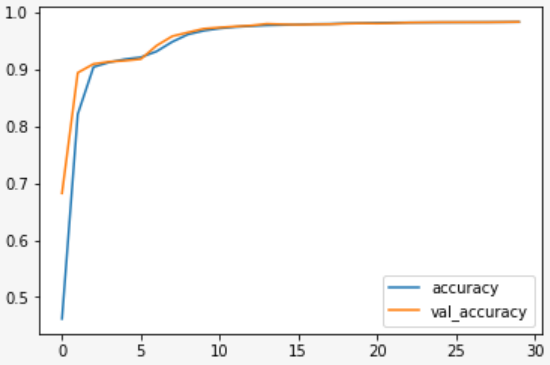
OUTPUT

NEURAL NETWORKS

RESULTS



LOSS VS VAL LOSS



ACCURACY VS VAL ACCURACY

TARGET

0	9265
1	2899
2	816
4	31
10	5
5	2
11	2
3	2
9	2
7	1
6	1

CONFUSION MATRIX

```
[[2274    2   10    0    0]
 [   6  730   13    0    0]
 [   7    4  200    0    0]
 [   0    0    1    0    0]
 [   3    1    6    0    0]]
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      2286
           1       0.99      0.97      0.98       749
           2       0.87      0.95      0.91       211
           3       0.00      0.00      0.00         1
           4       0.00      0.00      0.00        10

    accuracy                           0.98      3257
   macro avg       0.57      0.58      0.58      3257
weighted avg       0.98      0.98      0.98      3257
```

PAPER REFERENCES

RELEVANT PAPERS

Khomtchouk BB: 'Codon usage bias levels predict taxonomic identity and genetic composition'. bioRxiv, 2020, doi: 10.1101/2020.10.26.356295.

Manuel Fern´andez-Delgado, Eva Cernadas,Sen´en Barro,Dinani Amorim: 'Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?'. Journal of Machine Learning Research 15 (2014) 3133-3181

CITATION REQUEST

Khomtchouk BB: 'Codon usage bias levels predict taxonomic identity and genetic composition'. bioRxiv, 2020, doi: 10.1101/2020.10.26.356295.

Nakamura Y, Gojobori T, Ikemura T: 'Codon usage tabulated from international DNA sequence databases: status for the year 2000'. Nucleic Acids Research, 2000, 28:292