MACHINE LEARNING

FOR

AFFINE CIPHER

CRYPTOGRAPY – 18CSC442

*Aim:*

To generate a Machine Learning algorithm for affine cipher.

*Affine Cipher:*

Affine cipher is a mono-alphabetic substitution cipher, where each letter in an alphabet is mapped to its numeric equivalent

(A to Z ~ 0 to 25), which is encrypted using a simple mathematical equation and converted to its alphabetic equivalent.

Here a and b are constants and can be considered as keys. If we are only using alphabet set, then n=26.

*Intro:*

Our aim is to create a machine learning algorithm for affine cipher.

For any machine learning algorithms, the important part is data, with which it can learn and train itself. So, the major part to create a data set, which follows affine cipher.

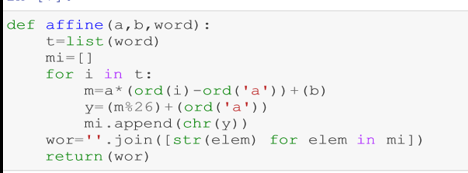
1. Data:

First, we need to consider random words to encrypt them and convert data into information. Here, we are taking random words from module random\_word in python.



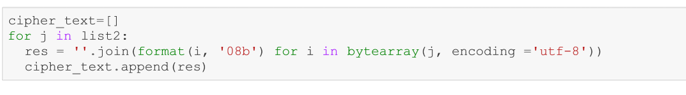


This is the data of words we required. Now we need to encrypt the words using affine cipher function.

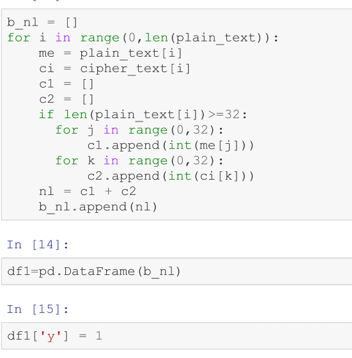


Now, we have two lists one with plain text of words and second is encrypted version of words which is cipher text.





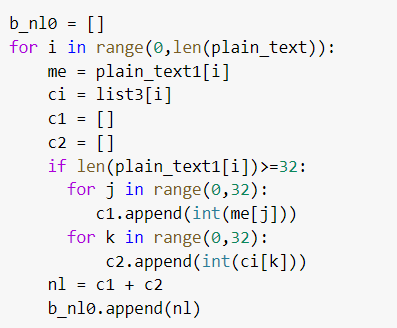
With binary encoder format change the value of each string in both lists to their corresponding 8-bit binary version of each alphabet.



Consider first 32- bits of each element in both lists. Join them and form a 64- bit binary value. Now we have a list with length of each element as 64. Now convert the list into matrix with m x n, where n=64. So, we create a data frame with 64 columns and 65th column will be label/output which is given as 1. Here, we are considering supervised machine learning algorithms, so we need an output parameter.

Next, we must take plaintexts and convert them into binary format.

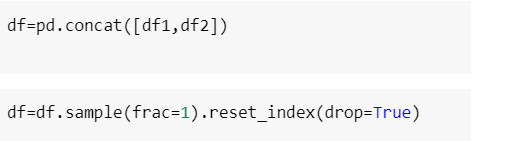


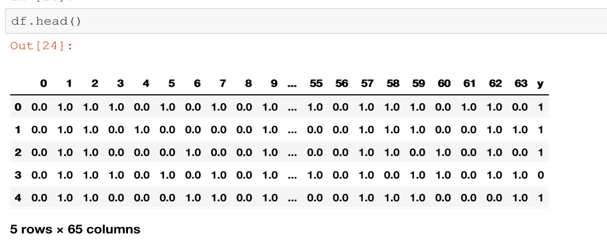


Length of each element of this list is 32 bits. Now, create a list with each element having length as 32- bits considering random of 0’s and 1’s. So, now we have list of 64-bits same as previous. Here label this as 0. So, convert this matrix into a data frame and its label values are 0.



Now, join both data frames with label=1 and label=0 row-wise. We will have our required complete data frame. But as the label is order wise shuffle rows.





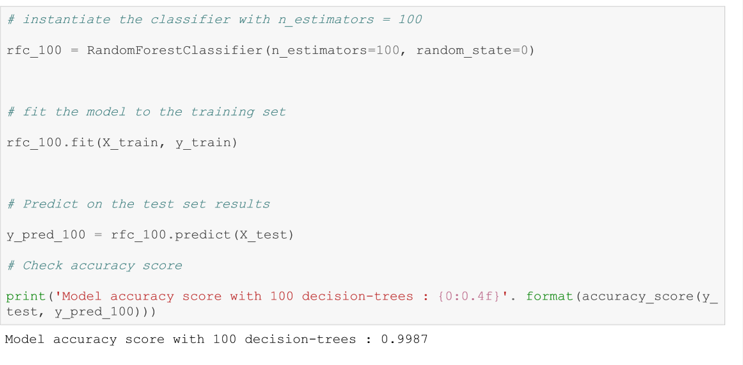
Now, we are ready with data. We must train and test the data with machine learning algorithms.

2. Machine Learning

Splitting the data into training and testing:

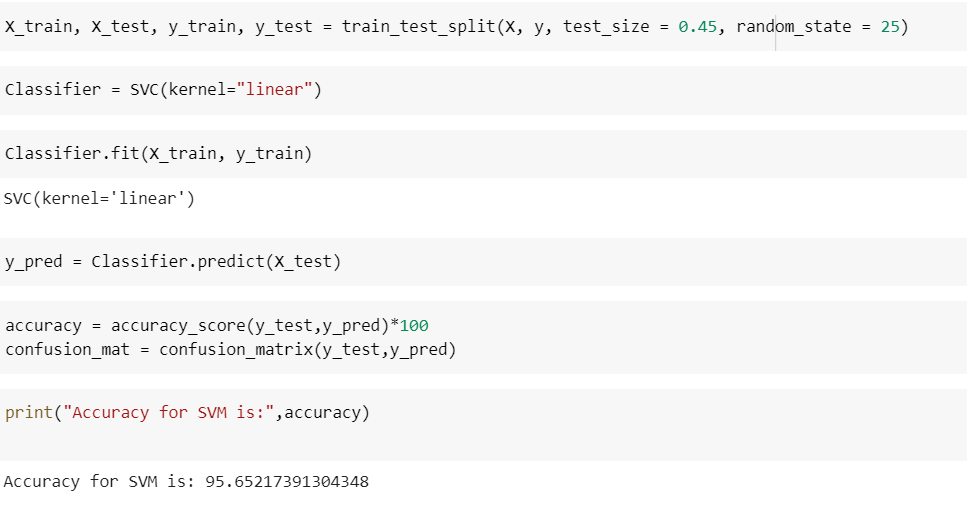


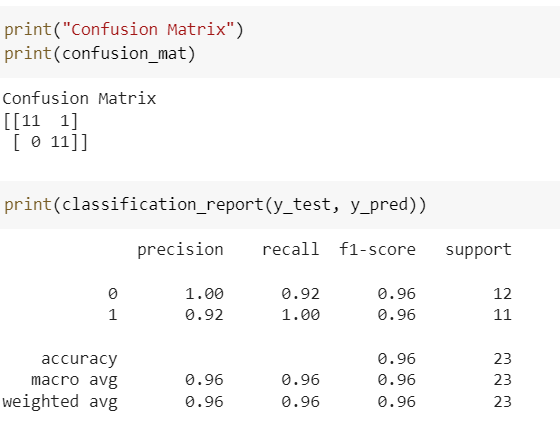
1. Rando Forest Classifier:





2. SVM





*Comparison:*

|  |  |
| --- | --- |
| Model | Accuracy |
| Random Forest Model | 99.87 |
| SVM | 99.874 |

*Inference:*

Here, we are getting similar accuracy rates for both models. So, we can either consider one or we can change number of decision trees and increase the volume of data. This model is particularly trained for same key values, but we can give different key values and evaluate. But accuracy of that condition will be low. Here, we generated machine learning algorithms for affine cipher.