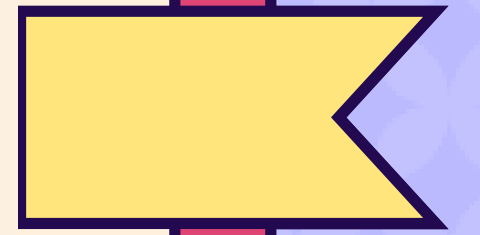
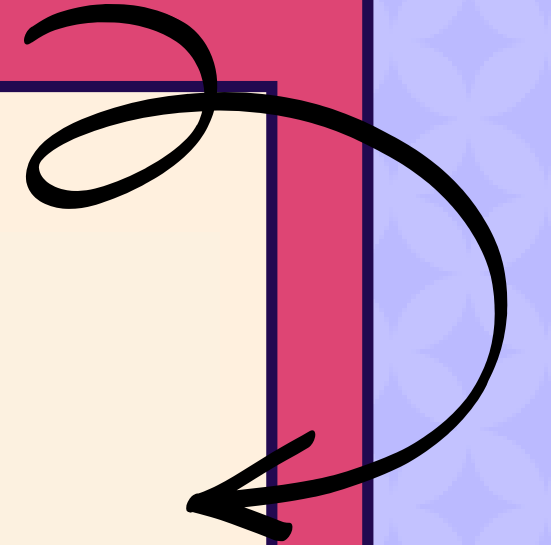
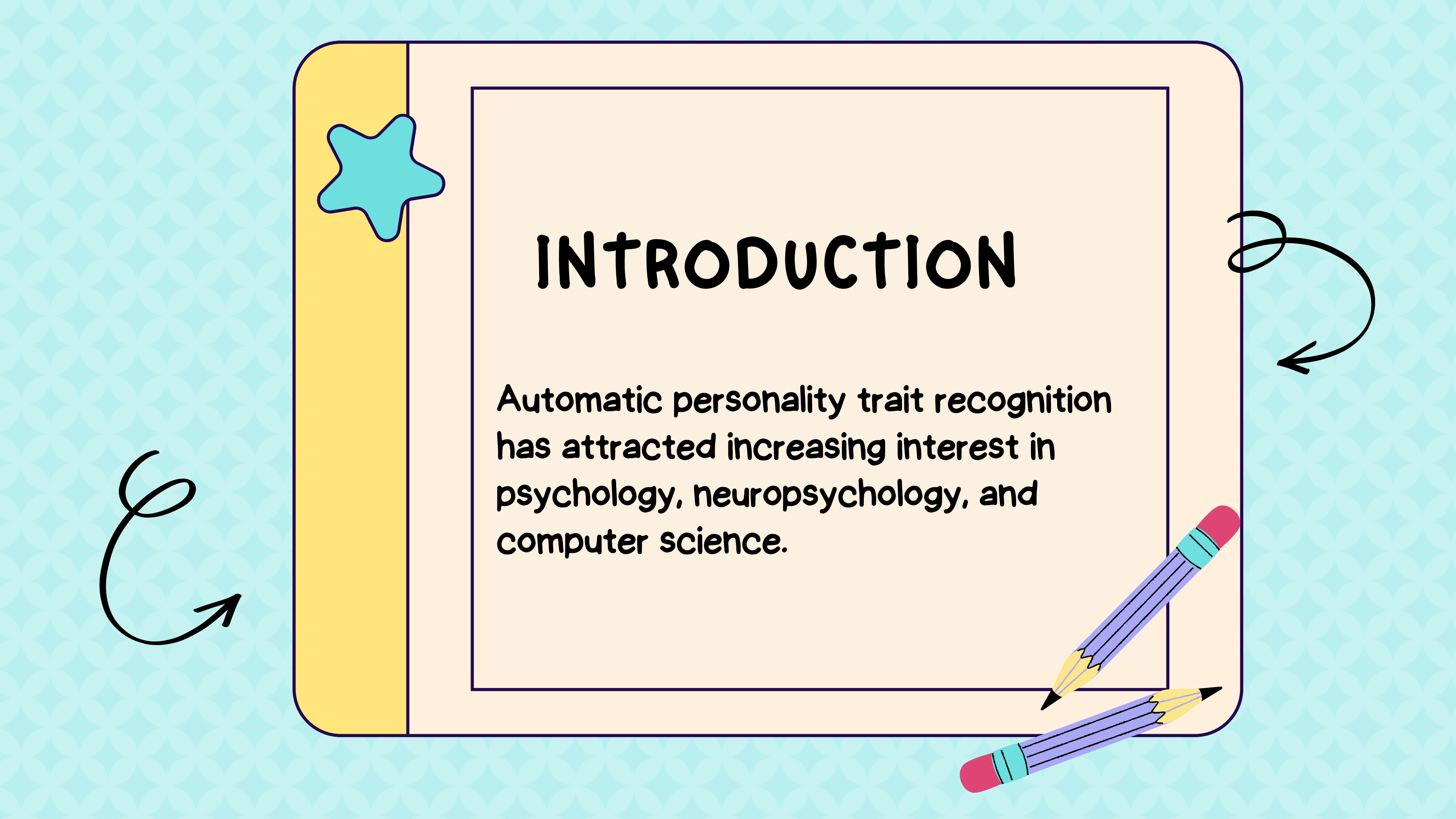




PERSONALITY DETECTION

**HAYAM MAHMOUD
DOHA AHMED
HADEEL HAMAM
ASMAA KHAMIS
MAHMOUD MOHAMED
AYMAN RAMADAN**





INTRODUCTION

Automatic personality trait recognition has attracted increasing interest in psychology, neuropsychology, and computer science.

APPLIED MACHINE LEARNING

1. Define problem
2. Summarize data
3. Prepare data
4. Preprocessing
5. Train the model
6. Select algorithm
7. Testing
8. Get accuracy ,prediction ,confusion matrix



DEFINE PROBLEM



Personality classification
based on a set of
questions whose answers
range from a range of
assessments.

SUMMARIZE DATA



get data ,features and classes

FEATURES KEYS

- 1.You regularly make new friends.
- 2.You spend a lot of your free time exploring various random topics that pique your interest
- 3.Seeing other people cry can easily make you feel like you want to cry too
- 4.You usually stay calm, even under a lot of pressure
- 5.At social events, you rarely try to introduce yourself to new people and mostly talk to the ones you already know
- 6.You prefer to completely finish one project before starting another.

- 7.You often make a backup plan for a backup plan
- 8.You are very sentimental.
- 9.You like to use organizing tools like schedules and lists.
- 10.Even a small mistake can cause you to doubt your overall abilities and knowledge.
- 11.You feel comfortable just walking up to someone you find interesting and striking up a conversation.
- 12.You are not too interested in discussing various interpretations and analyses of creative works.
- 13.You are more inclined to follow your head than your heart.

..... to 60

classes

Analysts



Architect

INTJ-A / INTJ-T

Imaginative and strategic thinkers,
with a plan for everything.



Logician

INTP-A / INTP-T

Innovative inventors with an
unquenchable thirst for
knowledge.



Commander

ENTJ-A / ENTJ-T

Bold, imaginative and strong-
willed leaders, always finding a
way – or making one.



Debater

ENTP-A / ENTP-T

Smart and curious thinkers who
cannot resist an intellectual
challenge.

Diplomats



Advocate

INFJ-A / INFJ-T

Quiet and mystical, yet very inspiring and tireless idealists.



Mediator

INFP-A / INFP-T

Poetic, kind and altruistic people, always eager to help a good cause.



Protagonist

ENFJ-A / ENFJ-T

Charismatic and inspiring leaders, able to mesmerize their listeners.



Campaigner

ENFP-A / ENFP-T

Enthusiastic, creative and sociable free spirits, who can always find a reason to smile.

Sentinels



Logistician

ISTJ-A / ISTJ-T

Practical and fact-minded individuals, whose reliability cannot be doubted.



Defender

ISFJ-A / ISFJ-T

Very dedicated and warm protectors, always ready to defend their loved ones.



Executive

ESTJ-A / ESTJ-T

Excellent administrators, unsurpassed at managing things – or people.



Consul

ESFJ-A / ESFJ-T

Extraordinarily caring, social and popular people, always eager to help.

Explorers



Virtuoso

ISTP-A / ISTP-T

Bold and practical experimenters,
masters of all kinds of tools.



Adventurer

ISFP-A / ISFP-T

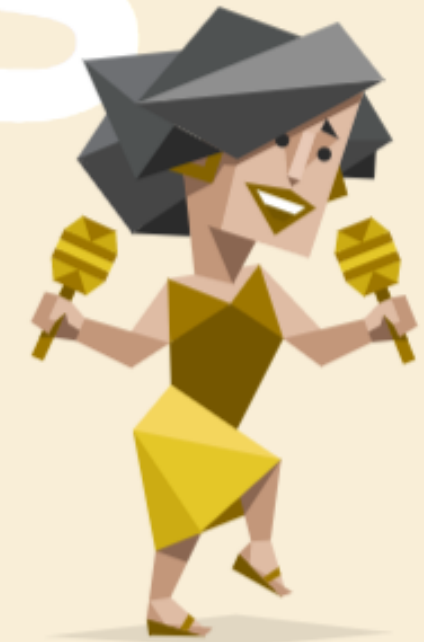
Flexible and charming artists,
always ready to explore and
experience something new.



Entrepreneur

ESTP-A / ESTP-T

Smart, energetic and very
perceptive people, who truly enjoy
living on the edge.



Entertainer

ESFP-A / ESFP-T

Spontaneous, energetic and
enthusiastic people – life is never
boring around them.

PREPARE DATA



get the dataset

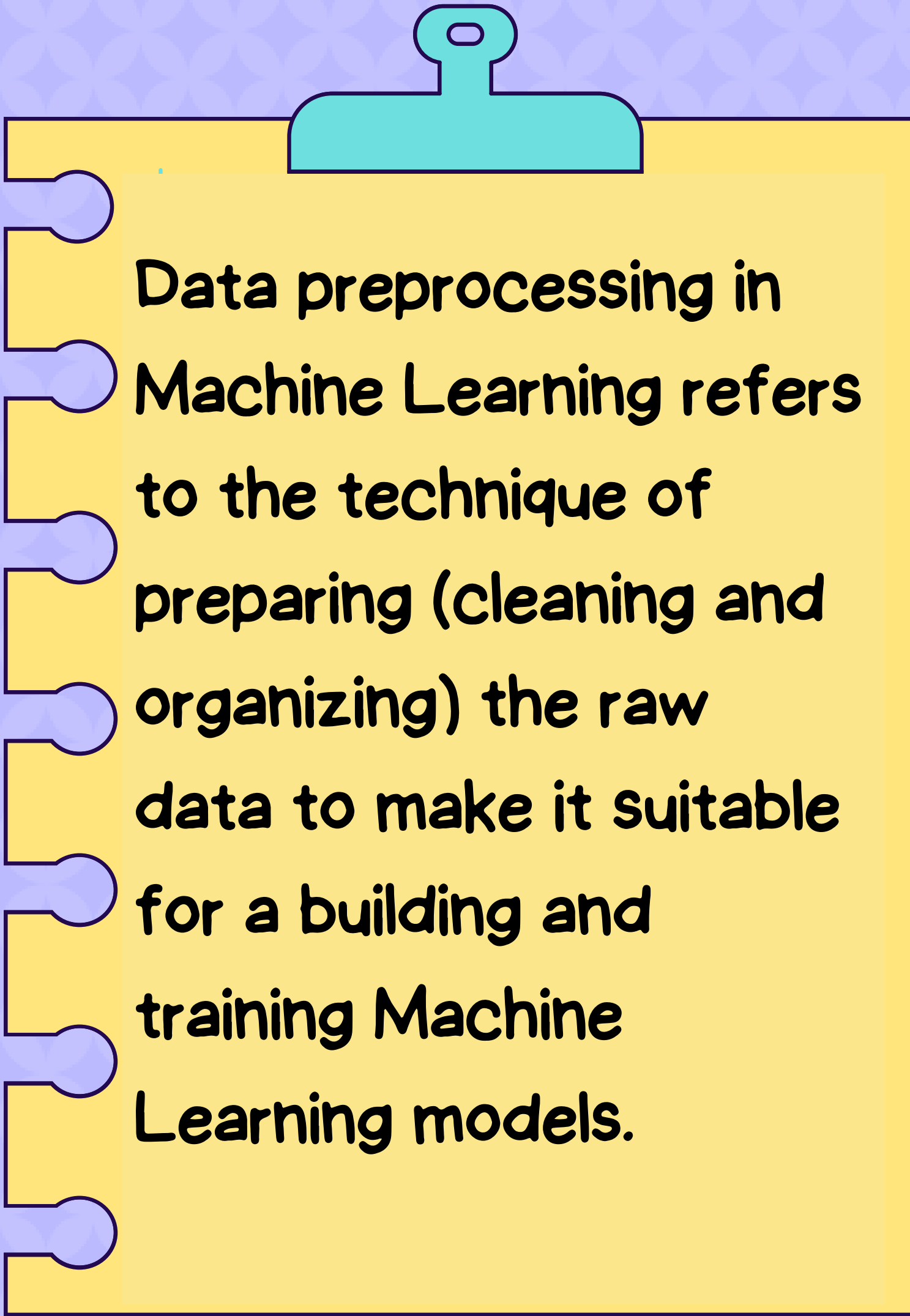


IMPORTING LIBRARIES

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g.
import matplotlib.pyplot as plt #visualization
```

READ THE DATASET

```
...  
  
#Reading the csv file using pandas  
df = pd.read_csv("perdet.csv")
```



Data preprocessing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for building and training Machine Learning models.

PREPROCESSING





```
#Showing if there are any null_values in the dataset  
df.isnull().sum()
```

TRAIN THE MODEL



A training model is a dataset that is used to train an ML algorithm. It consists of the sample output data and the corresponding sets of input data that have an influence on the output. The training model is used to run the input data through the algorithm to correlate the processed output against the sample output. The result from this correlation is used to modify the model.



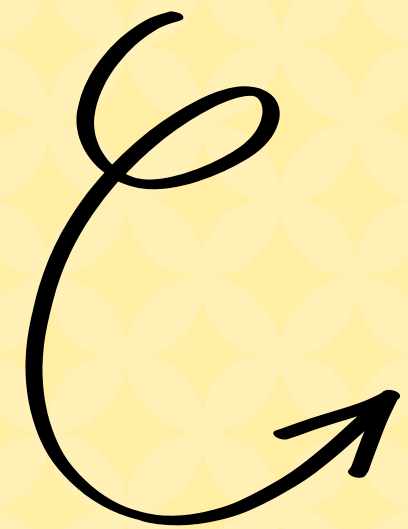
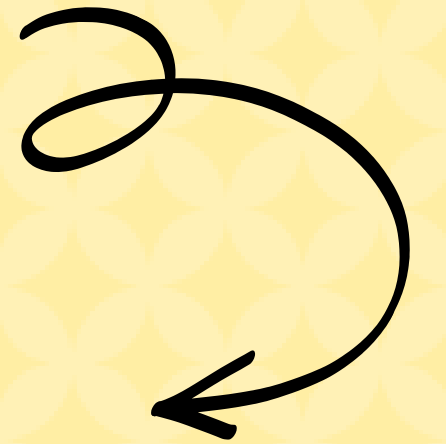
```
#Train,Test,Split  
from sklearn.model_selection import train_test_split  
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3, stratify =  
Y, random_state = 2)
```

SELECT ALGORITHM



- After categorizing the problem and understand the data, the next milestone is identifying the algorithms that are applicable and practical to implement in a reasonable time.

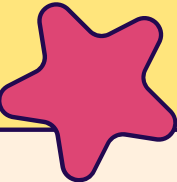
**KNEIGHBORS WHICH
WHAT WE CHOOSE...**





**WHY?
LET'S KNOW.**

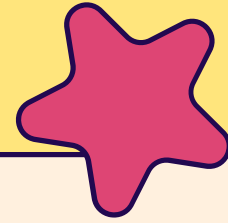


ADVANTAGES

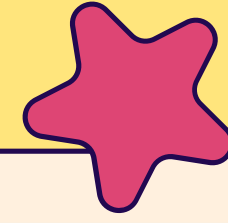
- 
- It is simple to implement.

- 
- It is robust to the noisy training data.

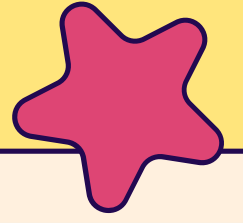
- 
- It can be more effective if the training data is large.



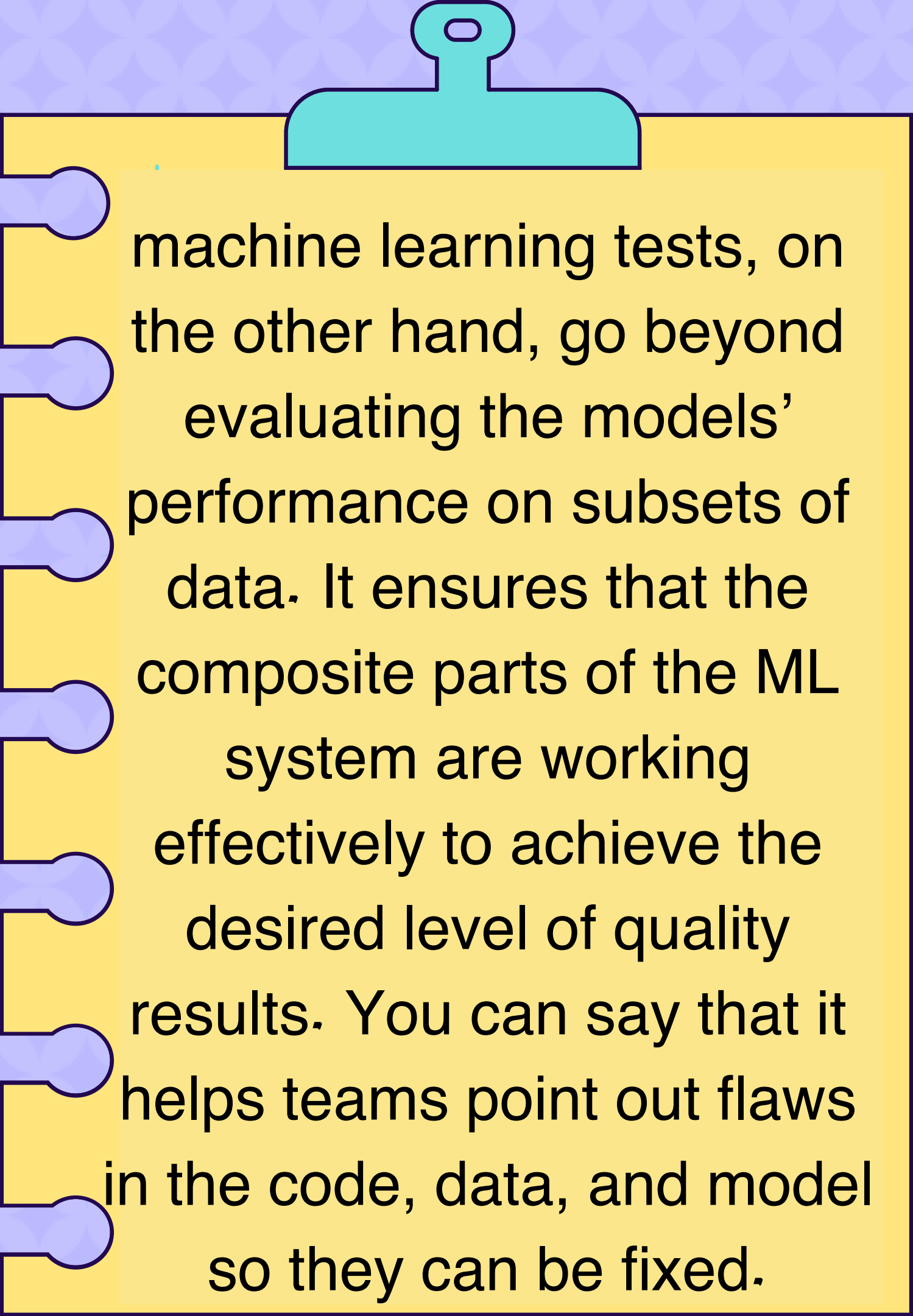
- Can learn non-linear decision boundaries when used for classification and regression. Can come up with a highly flexible decision boundary adjusting the value of K .



- No Training Time for classification/regression : The KNN algorithm has no explicit training step and all the work happens during prediction.



- Constantly evolves with new data: Since there is no explicit training step, as we keep adding new data to the dataset, the prediction is adjusted without having to retrain a new model.



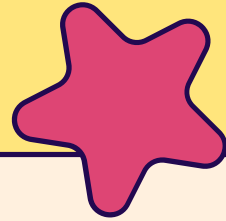
machine learning tests, on the other hand, go beyond evaluating the models' performance on subsets of data. It ensures that the composite parts of the ML system are working effectively to achieve the desired level of quality results. You can say that it helps teams point out flaws in the code, data, and model so they can be fixed.

TESTING

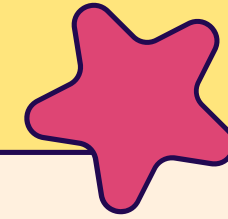


```
from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier()
model.fit(X_train, Y_train)
#Finding the accuracy score on training dataset
from sklearn.metrics import accuracy_score
X_train_prediction = model.predict(X_train)
train_data_accuracy = accuracy_score(X_train_prediction, Y_train)
train_data_accuracy
#Finding the accuracy score on test dataset
from sklearn.metrics import accuracy_score
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
test_data_accuracy
```

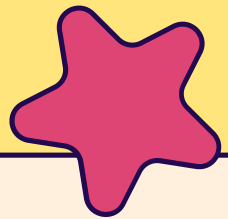
GET...ACCURACY,PREDICTION ,CONFUSION MATRIX



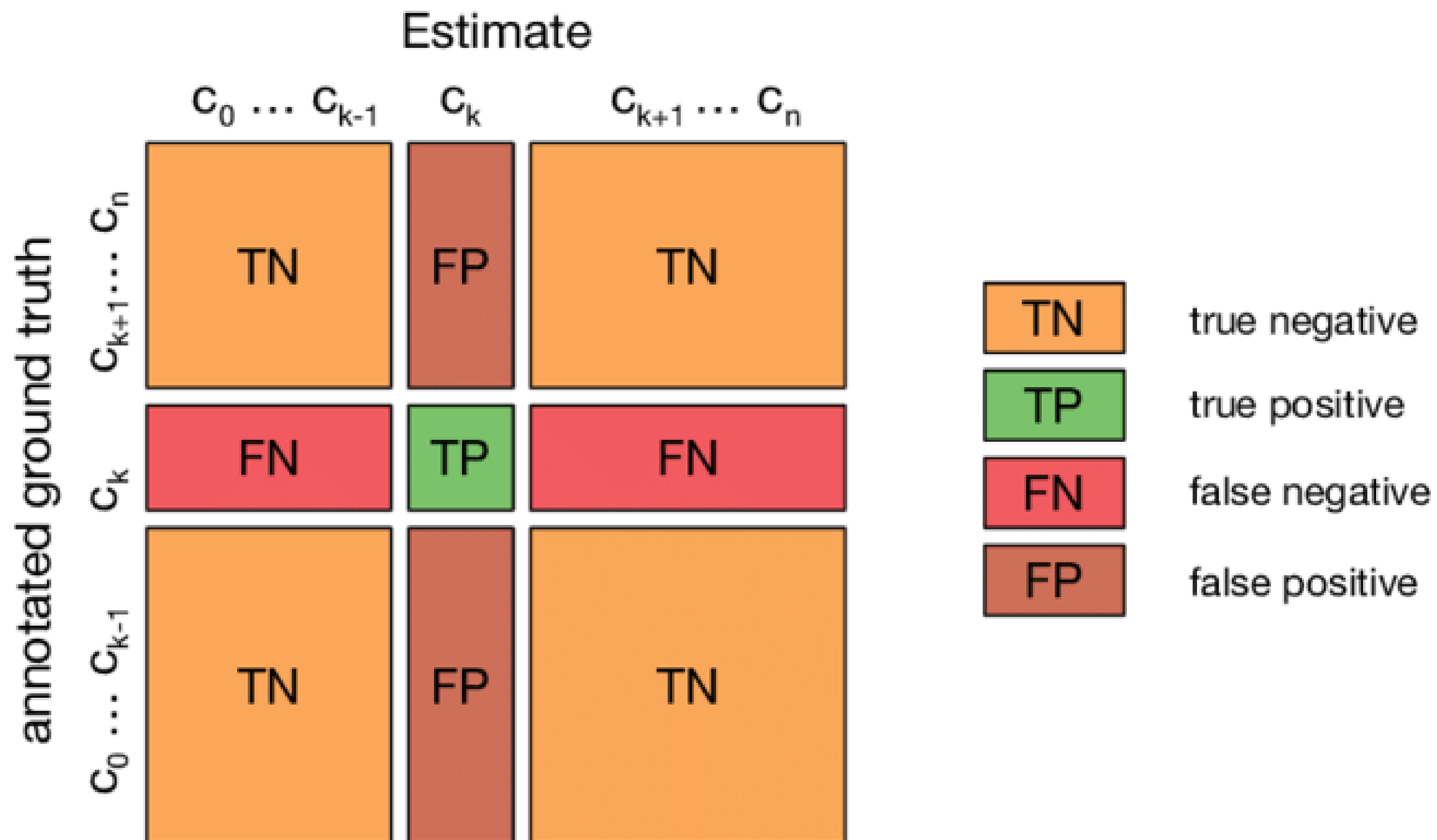
- **confusion matrix** is a tabular way of visualizing the performance of your prediction model.

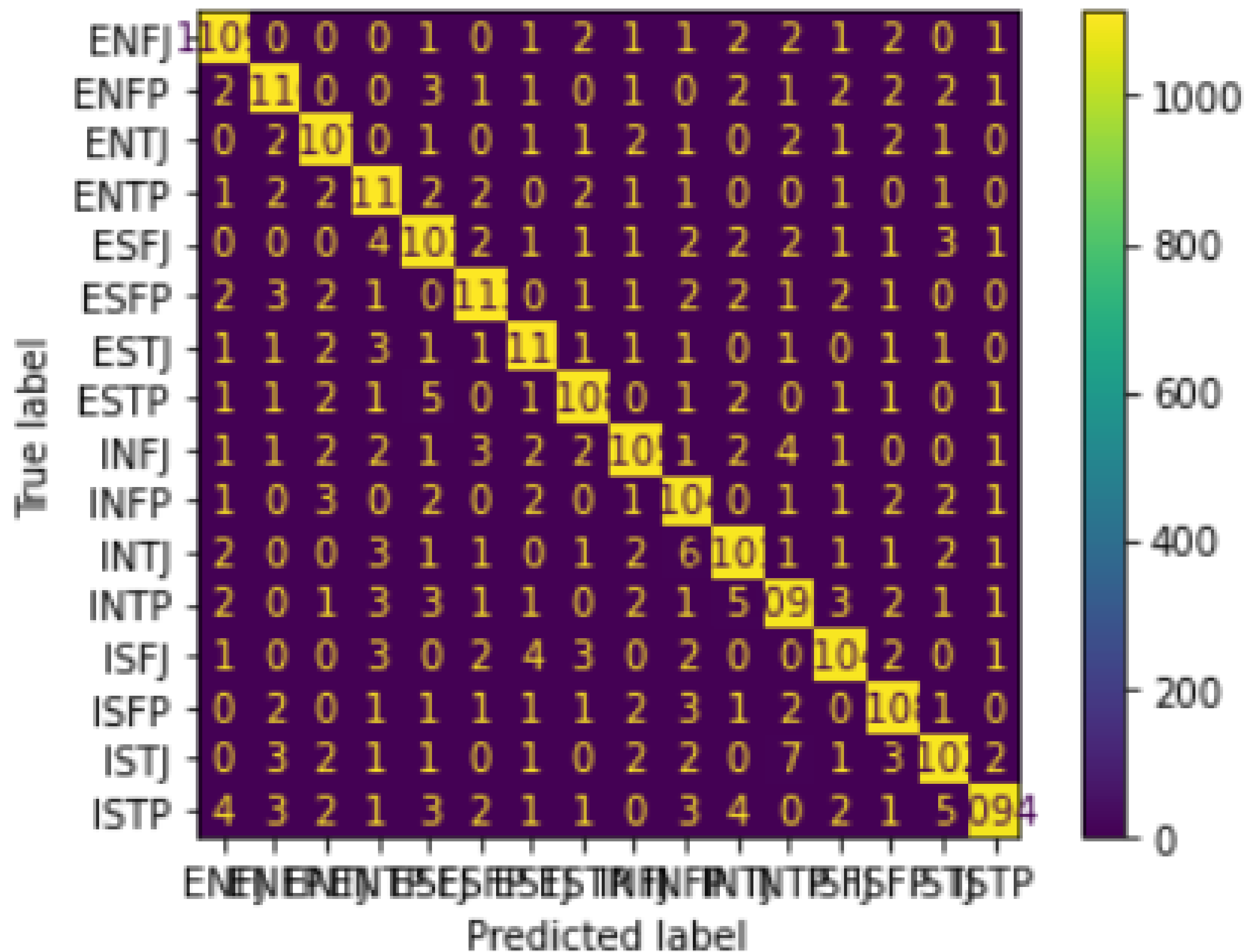



- **Accuracy** is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. .



- **Prediction** essentially means to predict a future outcome, similarly to what is accomplished in machine learning. It refers to the output of an algorithm post training on a historical data set.







```
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
X_test_prediction=model.predict(X_test)
test_data_accuracy=accuracy_score(X_test_prediction,Y_test)
print(accuracy_score(X_test_prediction,Y_test))
print(confusion_matrix(X_test_prediction,Y_test))
print(classification_report(X_test_prediction,Y_test))
```

0.9827777777777778

=====

```
[[1109  0  0  0  1  0  1  2  1  1  2  2  1  2
  0  1]
[  2 1110  0  0  3  1  1  0  1  0  2  1  2  2
  2  1]
[  0  2 1107  0  1  0  1  1  2  1  0  2  1  2
  1  0]
[  1  2  2 1113  2  2  0  2  1  1  0  0  1  0
  1  0]
[  0  0  0  4 1103  2  1  1  1  2  2  2  1  1
  3  1]
[  2  3  2  1  0 1113  0  1  1  2  2  1  2  1
  0  0]
[  1  1  2  3  1  1 1113  1  1  1  0  1  0  1
  1  0]
[  1  1  2  1  5  0  1 1108  0  1  2  0  1  1
  0  1]
[  1  1  2  2  1  3  2  2 1105  1  2  4  1  0
  0  1]
[  1  0  3  0  2  0  2  0  1 1104  0  1  1  2
  2  1]
[  2  0  0  3  1  1  0  1  2  6 1101  1  1  1
  2  1]
[  2  0  1  3  3  1  1  0  2  1  5 1096  3  2
  1  1]
[  1  0  0  3  0  2  4  3  0  2  0  0 1104  2
  0  1]
[  0  2  0  1  1  1  1  1  2  3  1  2  0 1108
  1  0]
[  0  3  2  1  1  0  1  0  2  2  0  7  1  3
 1102  2]
[  4  3  2  1  3  2  1  1  0  3  4  0  2  1
  5 1094]]
```

=====

=====				
	precision	recall	f1-score	support
ENFJ	0.98	0.99	0.99	1123
ENFP	0.98	0.98	0.98	1128
ENTJ	0.98	0.99	0.99	1121
ENTP	0.98	0.99	0.98	1128
ESFJ	0.98	0.98	0.98	1124
ESFP	0.99	0.98	0.98	1131
ESTJ	0.98	0.99	0.99	1128
ESTP	0.99	0.98	0.99	1125
INFJ	0.98	0.98	0.98	1128
INFP	0.98	0.99	0.98	1120
INTJ	0.98	0.98	0.98	1123
INTP	0.98	0.98	0.98	1122
ISFJ	0.98	0.98	0.98	1122
ISFP	0.98	0.99	0.98	1124
ISTJ	0.98	0.98	0.98	1127
ISTP	0.99	0.97	0.98	1126
accuracy			0.98	18000
macro avg	0.98	0.98	0.98	18000
weighted avg	0.98	0.98	0.98	18000

THANK YOU!