# COURSE CODE : INT-254

# PROJECT REPORT

# ON

# Divorce Prediction Project

Submitted in partial fulfilment of the requirements for the assignment

Of

**B. Tech** In

**Computer Science & Engineering** (Hons.)

Submitted to:

**LOVELY PROFESSIONAL UNIVERSITY**

**Submitted by:**

J.Nithin (12007854)

https://github.com/EkantYadav/Divorce-Prediction

2. Ekant (12020586)

Github link-https://github.com/EkantYadav/Divorce-Prediction

**Under the Guidance of**

**Dr.Dhanpratap Singh**

# CANDIDATE'S DECLARATION

We hereby certify that the work, which is being presented in this report entitled, **Divorce Predction**, in partial fulfillment of the requirements for the degree of **Bachelor of Technology, INT-254** submitted in the **Computer Science and Engineering** , Lovely Professional University, Phagwara, Punjab; by **J.Nithin(12007854), Ekant(12020586),** is the authentic record of our own work carried out under the supervision of **Dr. Dhanpratap Singh**, **Professor, Computer Science and Engineering**, Lovely Professional University, Phagwara, Punjab.

We further declare that the matter embodied in this report has not been submitted by us for the award of any other degree.

**Candidate(s) Signature**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

**Signature of Students** **Signature of Supervisor**

**Nithin(12007854).** **Dr.DhanpratapSingh**

**Ekant(12020586).**

**Date :**

# ACKNOWLEDGMENT

It is our pleasure to acknowledge the contributions of all who have helped us and supported us during this Project report.

First, we thank God for helping us in one way or another and providing strength and endurance to us. We wish to express my sincere gratitude and indebtedness to our supervisor Dr. **Dhanpratap Singh**, Professor, Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab; for his intuitive and meticulous guidance and perpetual inspiration in completion of this report. In spite of his busy schedule, he rendered help whenever needed, giving useful suggestions and holding informal discussions. His invaluable guidance and support throughout this work cannot be written down in few words. We also thank her for providing facilities for my work in the department.

We are also humbly obliged by the support of our group members and friends for their love and caring attitude. The sentimental support they rendered to us is invaluable and everlasting. They have helped us through thick and thin and enabled us to complete the work with joy and vigor. We thank the group members for entrusting in each other and following directions, without them this report would never have been possible.

We are also thankful to our parents, elders and all family members for their blessing, motivation and inspiration throughout our work and bearing with us even during stress and bad temper. They have always provided us a high moral support and contributed in all possible ways in completion of this Capstone report

Thank You!

# ABSTRACT

The number of divorce cases are increasing very rapidly all over the world. In the last few decades, the number of divorces have gone up from 1 in 1000 to 13 in 1000 in India. Due to this reason, it is a major concern for marriage counsellors and therapists.

Therefore, an effective divorce prediction technique is needed that helps a marriage counsellor or a therapist to identify how severe a case is. In this work, the authors present a study on divorce case prediction using the existing machine learning algorithms.

This project after the completion will predict whether the divorce will occur or not. This can help the people to analyse how tense the situation is between a couple and hence help them accordingly to their condition.

# TABLE OF CONTENTS

**Title** **PageNo.**

**S No.** **Title** **Page no.**

A **Divorce prediction problem** In this dataset, we have 170 samples with 54 input features, which are :

1.When one of our apologies apologizes when our discussions go in a bad direction, the issue does not extend.

2. I know we can ignore our differences, even if things get hard sometimes.

3. When we need it, we can take our discussions with my wife from the beginning and correct it.

4. When I argue with my wife, it will eventually work for me to contact him.

5. The time I spent with my wife is special for us.

6. We don't have time at home as partners.

7. We are like two strangers who share the same environment at home rather than family.

8. I enjoy our holidays with my wife.

9. I enjoy traveling with my wife.

10. My wife and most of our goals are common.

11. I think that one day in the future, when I look back, I see that my wife and I are in harmony with each other.

12. My wife and I have similar values in terms of personal freedom.

13. My husband and I have similar entertainment.

14. Most of our goals for people (children, friends, etc.) are the same.

15. Our dreams of living with my wife are similar and harmonious

16. We're compatible with my wife about what love should be

17. We share the same views with my wife about being happy in your life

18. My wife and I have similar ideas about how marriage should be

19. My wife and I have similar ideas about how roles should be in marriage

20. My wife and I have similar values in trust

21. I know exactly what my wife likes.

22. I know how my wife wants to be taken care of when she's sick.

23. I know my wife's favorite food.

24. I can tell you what kind of stress my wife is facing in her life.

25. I have knowledge of my wife's inner world.

26. I know my wife's basic concerns.

27. I know what my wife's current sources of stress are.

28. I know my wife's hopes and wishes.

29. I know my wife very well.

30. I know my wife's friends and their social relationships.

31. I feel aggressive when I argue with my wife.

32. When discussing with my wife, I usually use expressions such as

33. I can use negative statements about my wife's personality during our discussions.

34. I can use offensive expressions during our discussions.

35. I can insult our discussions.

36. I can be humiliating when we argue.

37. My argument with my wife is not calm.

38. I hate my wife's way of bringing it up.

39. Fights often occur suddenly.

40. We're just starting a fight before I know what's going on.

41. When I talk to my wife about something, my calm suddenly breaks.

42. When I argue with my wife, it only snaps in and I don't say a word.

43. I'm mostly thirsty to calm the environment a little bit.

44. Sometimes I think it's good for me to leave home for a while.

45. I'd rather stay silent than argue with my wife.

46. Even if I'm right in the argument, I'm thirsty not to upset the other side.

47. When I argue with my wife, I remain silent because I am afraid of not being able to control my anger.

48. I feel right in our discussions.

49. I have nothing to do with what I've been accused of.

50. I'm not actually the one who's guilty about what I'm accused of.

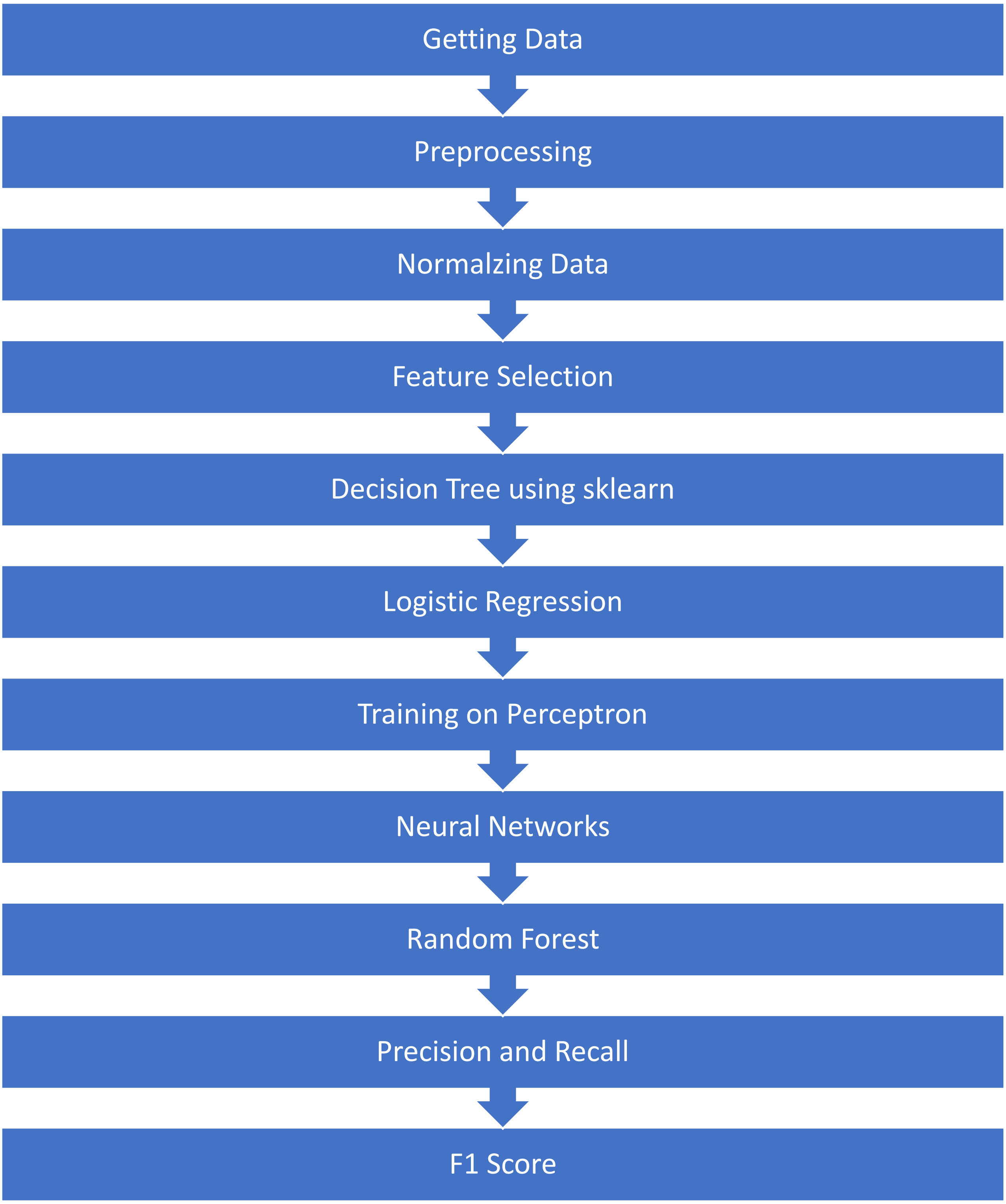
51. I'm not the one who's wrong about problems at home.

52. I wouldn't hesitate to tell her about my wife's inadequacy.

53. When I discuss it, I remind her of my wife’s inadequate issues.

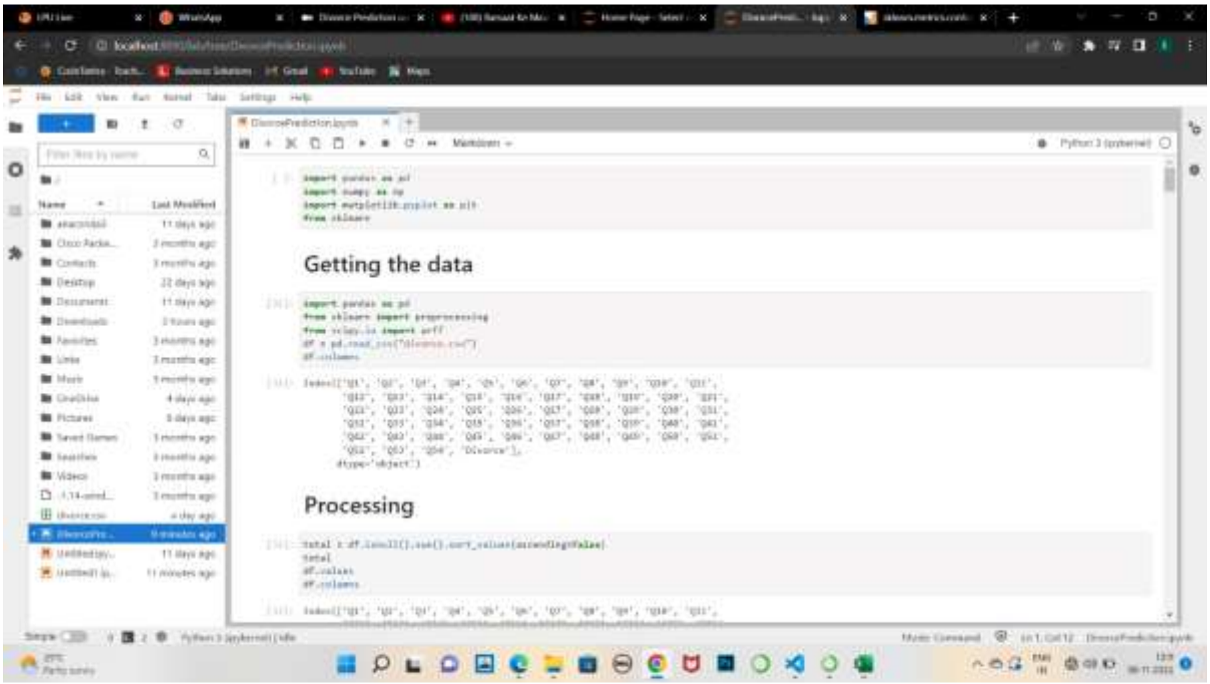
54. I’m not afraid to tell her about my wife’s incompetence.

## Flow Chart of Work Flow



Getting Data:

```
data = pd.read_csv('../input/divorce-prediction/divorce_data.csv', delimiter=';'
Data
```



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn
```

Getting the data

```
import pandas as pd
from sklearn import preprocessing
from scipy.io import arff
df = pd.read_csv("divorce.csv")
df.columns
```

```
Index(['Q1', 'Q2', 'Q3', 'Q4', 'Q5', 'Q6', 'Q7', 'Q8', 'Q9', 'Q10', 'Q11',
       'Q12', 'Q13', 'Q14', 'Q15', 'Q16', 'Q17', 'Q18', 'Q19', 'Q20', 'Q21',
       'Q22', 'Q23', 'Q24', 'Q25', 'Q26', 'Q27', 'Q28', 'Q29', 'Q30', 'Q31',
       'Q32', 'Q33', 'Q34', 'Q35', 'Q36', 'Q37', 'Q38', 'Q39', 'Q40', 'Q41',
       'Q42', 'Q43', 'Q44', 'Q45', 'Q46', 'Q47', 'Q48', 'Q49', 'Q50', 'Q51',
       'Q52', 'Q53', 'Q54', 'Divorce'],
      dtype='object')
```

Processing

```
total = df.isnull().sum().sort_values(ascending=False)
total
df.values
df.columns
```

```
Index(['Q1', 'Q2', 'Q3', 'Q4', 'Q5', 'Q6', 'Q7', 'Q8', 'Q9', 'Q10', 'Q11',
```

# Preprocessing

## Check for null values

```
total = df.isnull().sum().sort_values(ascending=False)
```

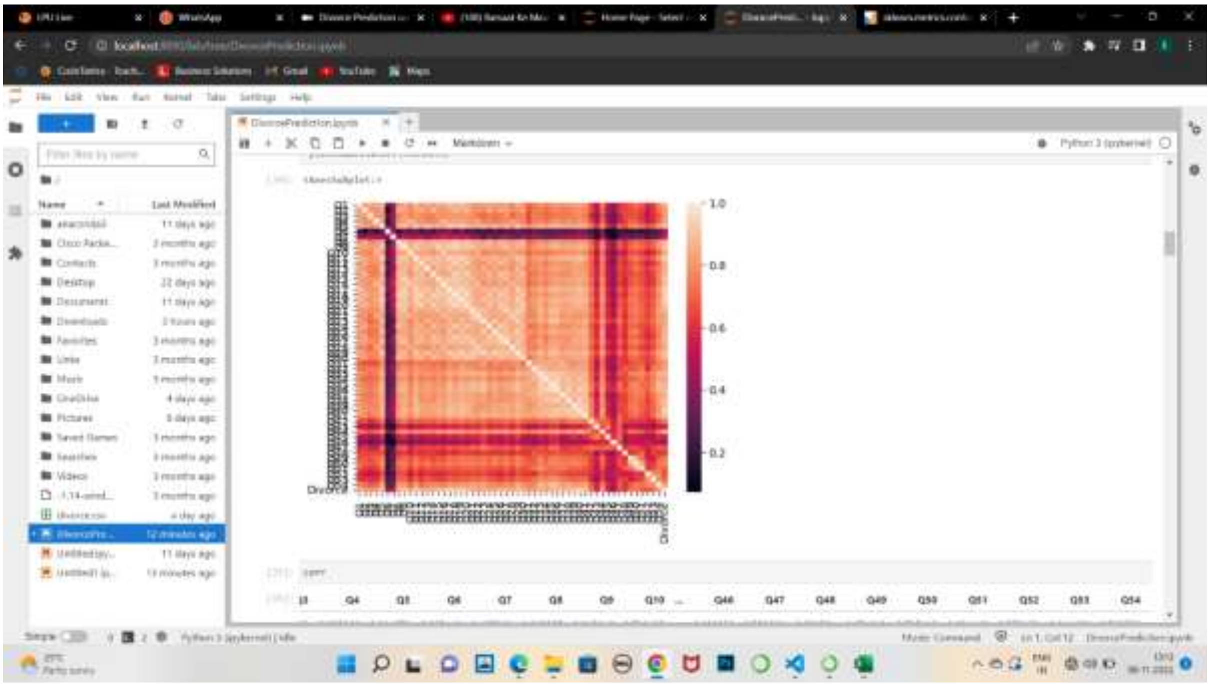
## Normalize data

```
from sklearn import preprocessingx = df.values #returns a numpy
array
print(x)
min_max_scaler = preprocessing.MinMaxScaler()
x_scaled = min_max_scaler.fit_transform(x)
df = pd.DataFrame(x_scaled)
df
```

## Feature Selection

We can select the important features and observe how much a feature is effecting our prediction.For this,we use the pearson correlation matrix.

```
import seaborn as sns# load the R package ISLR# load the Auto dataset
auto_df = df# calculate the correlation matrix
corr = auto_df.corr()# plot the heatmap
sns.heatmap(corr,
 xticklabels=corr.columns,
 yticklabels=corr.columns)
```

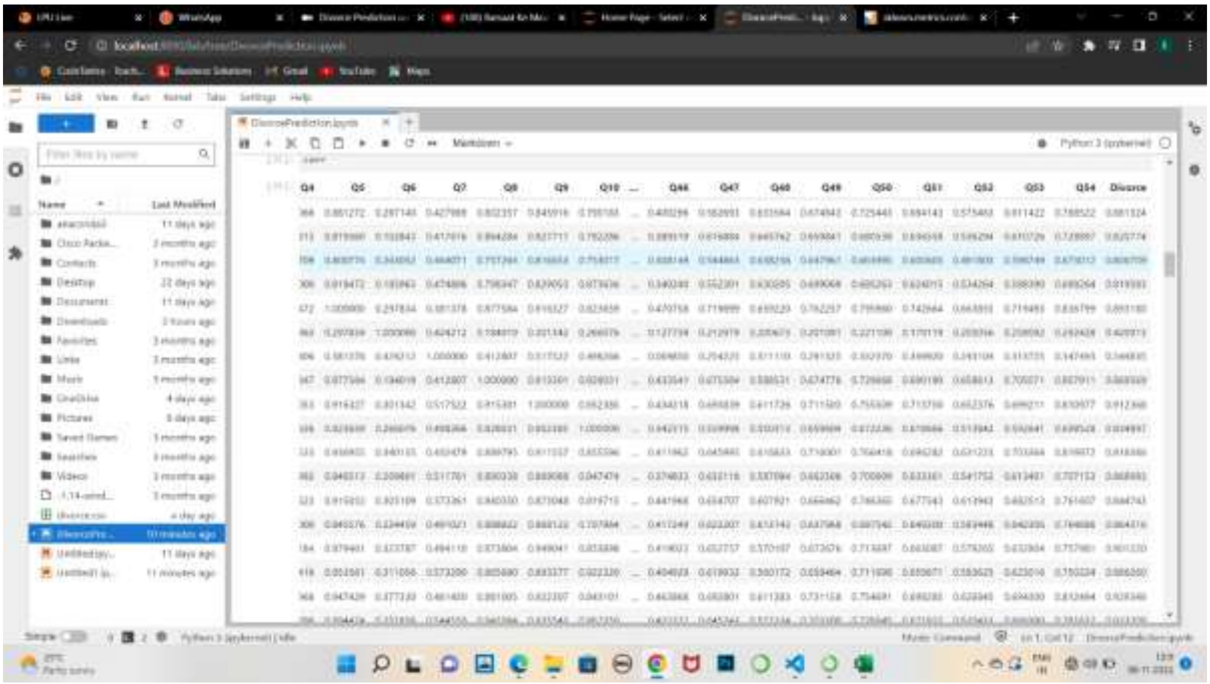


```
<AxesSubplot:>
```

```
corr
```

```
3    Q4    Q5    Q6    Q7    Q8    Q9    Q10  ...  Q46   Q47   Q48   Q49   Q50   Q51   Q52   Q53   Q54
```

```
Corr
```



Bigger numbers mean more correlation,irrespective of the sign.

## Store the predicting class into another variable and drop it from the Table

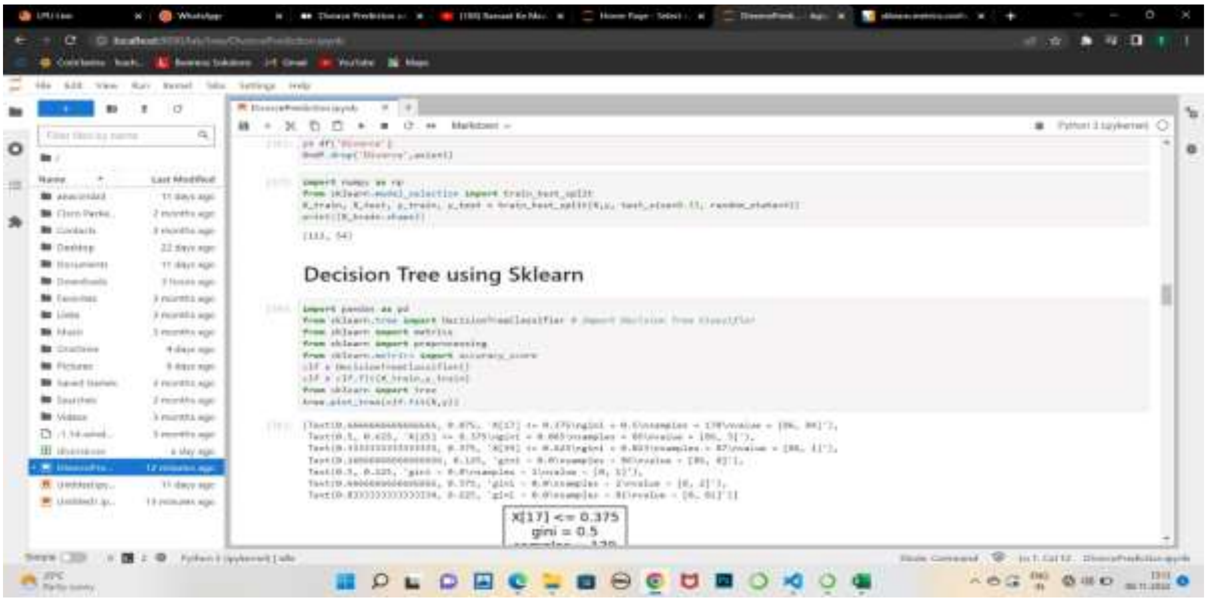
```
y= df['Class']
X=df.drop('Class',axis=1)
```

## Split Dataset into Training and Testing sets

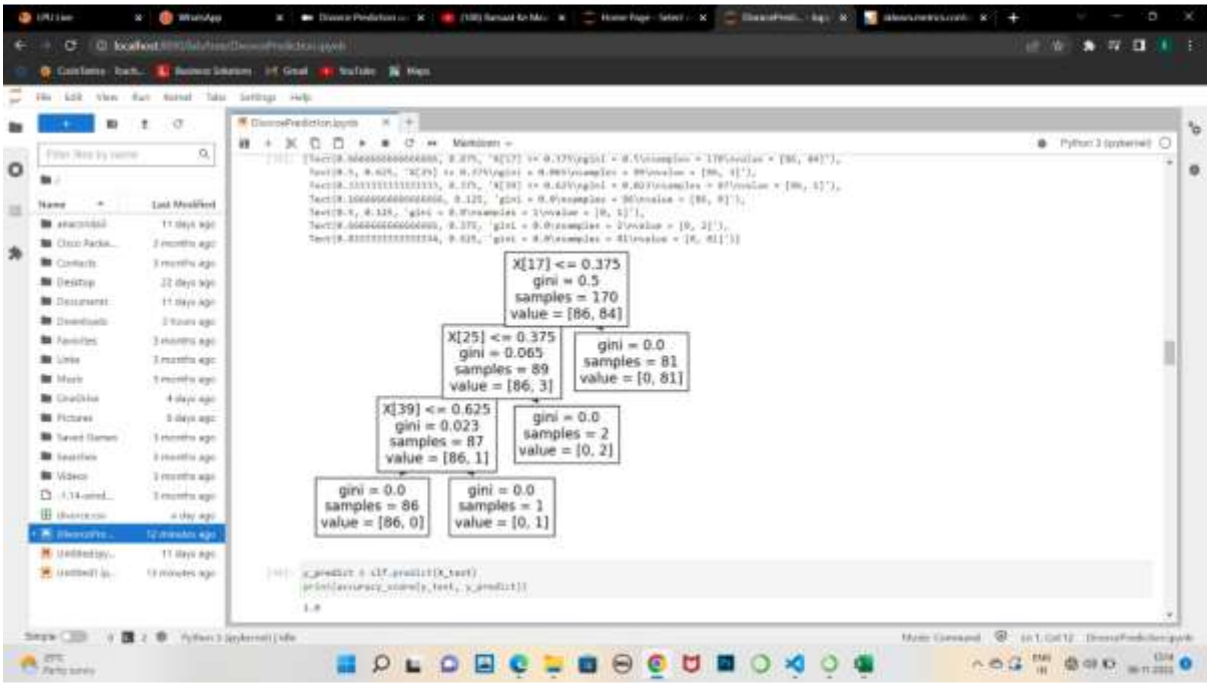
```
import numpy as np
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,
test_size=0.33, random_state=42)
print((X_train.shape))
```

## Decision Tree using Sklearn

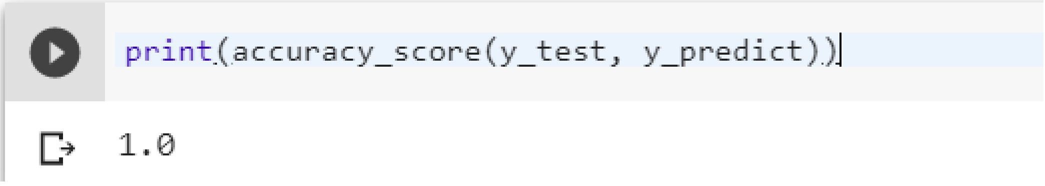
```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier # Import
Decision Tree Classifier
from sklearn import metrics
from sklearn import preprocessing
from sklearn.metrics import accuracy_scoreclf =
DecisionTreeClassifier()
clf = clf.fit(X_train,y_train)from sklearn import tree
tree.plot_tree(clf.fit(X,y))
```



Decision Tree using Sklearn



```
y_predict = clf.predict(X_test)
print(accuracy_score(y_test, y_predict))
```



```
print(accuracy_score(y_test, y_predict))
```

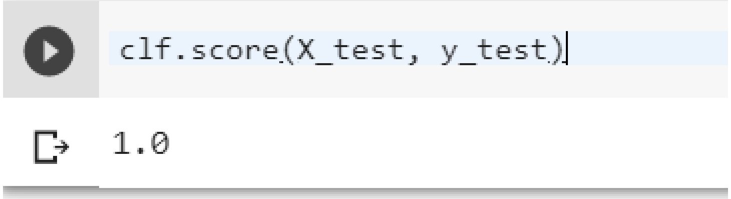
```
1.0
```

Resulting accuracy score after being trained on Decision Tree

# Logistic Regression

In statistics, the **logistic model** (or **logit model**) is a statistical model that models the probability of an event taking place by having the log-odds for the event be a linear combination of one or more independent variables. In regression analysis, **logistic regression**[1] (or **logit regression**) is estimating the parameters of a logistic model (the coefficients in the linear combination).

```
from sklearn.datasets import load_iris
from sklearn.linear_model import LogisticRegression
X, y = load_iris(return_X_y=True)
clf = LogisticRegression(random_state=0,
solver='lbfgs').fit(X_train, y_train)clf.score(X_test, y_test)
```

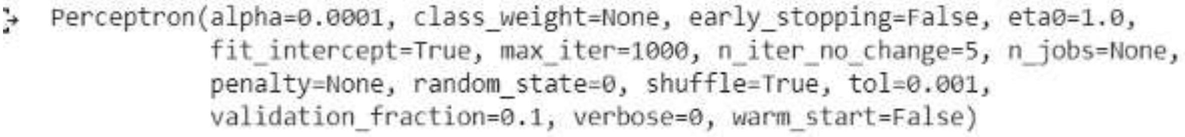


```
clf.score(X_test, y_test)
```

```
1.0
```

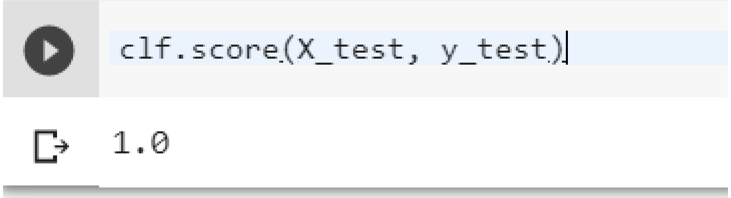
# Training on Perceptron

```
from sklearn.datasets import load_digits
from sklearn.linear_model import Perceptronclf =
Perceptron(tol=1e-3, random_state=0)
clf.fit(X_train, y_train)
```



```
Perceptron(alpha=0.0001, class_weight=None, early_stopping=False, eta0=1.0,
           fit_intercept=True, max_iter=1000, n_iter_no_change=5, n_jobs=None,
           penalty=None, random_state=0, shuffle=True, tol=0.001,
           validation_fraction=0.1, verbose=0, warm_start=False)
```

```
lf.score(X_test, y_test)
```



```
clf.score(X_test, y_test)
```

```
1.0
```

# Neural Networks

A neural network is **a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates**

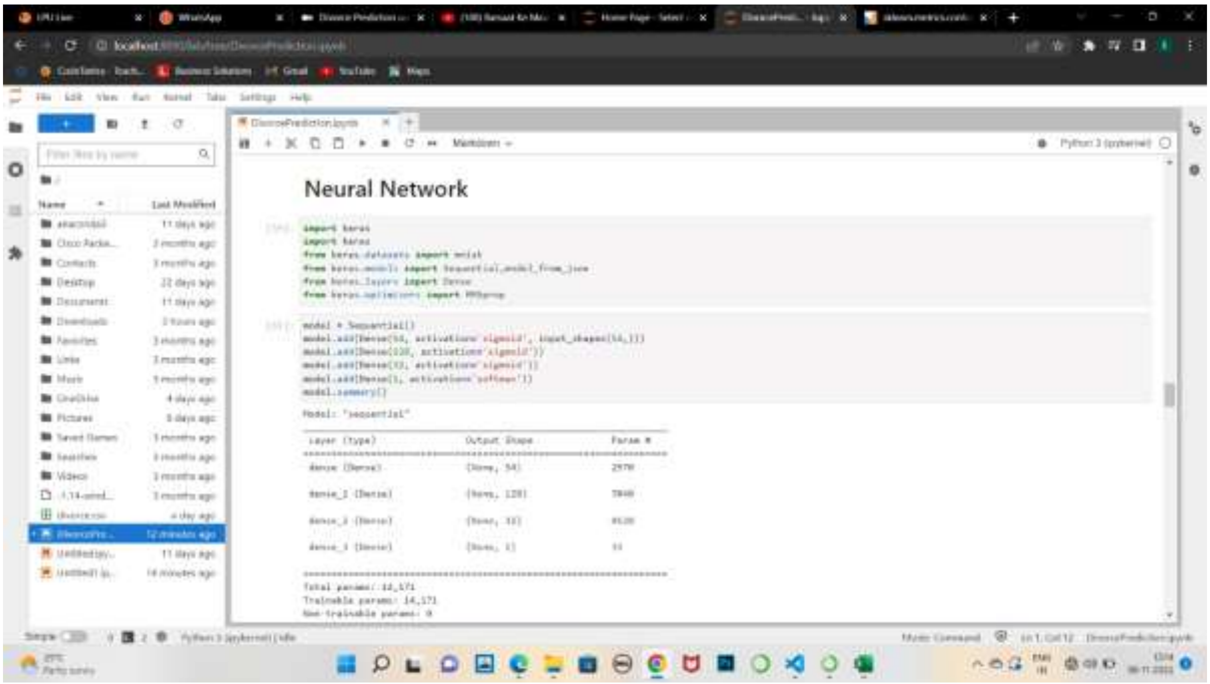
## First we import all the libraries

```
import keras
import keras
```

```
from keras.datasets import mnist
from keras.models import Sequential,model_from_json
from keras.layers import Dense
from keras.optimizers import RMSprop
```

## Now,we build our model

```
model = Sequential()
model.add(Dense(54, activation=’sigmoid’, input_shape=(54,)))
model.add(Dense(128, activation=’sigmoid’))
model.add(Dense(32, activation=’sigmoid’))
model.add(Dense(1, activation=’softmax’))model.summary()
```



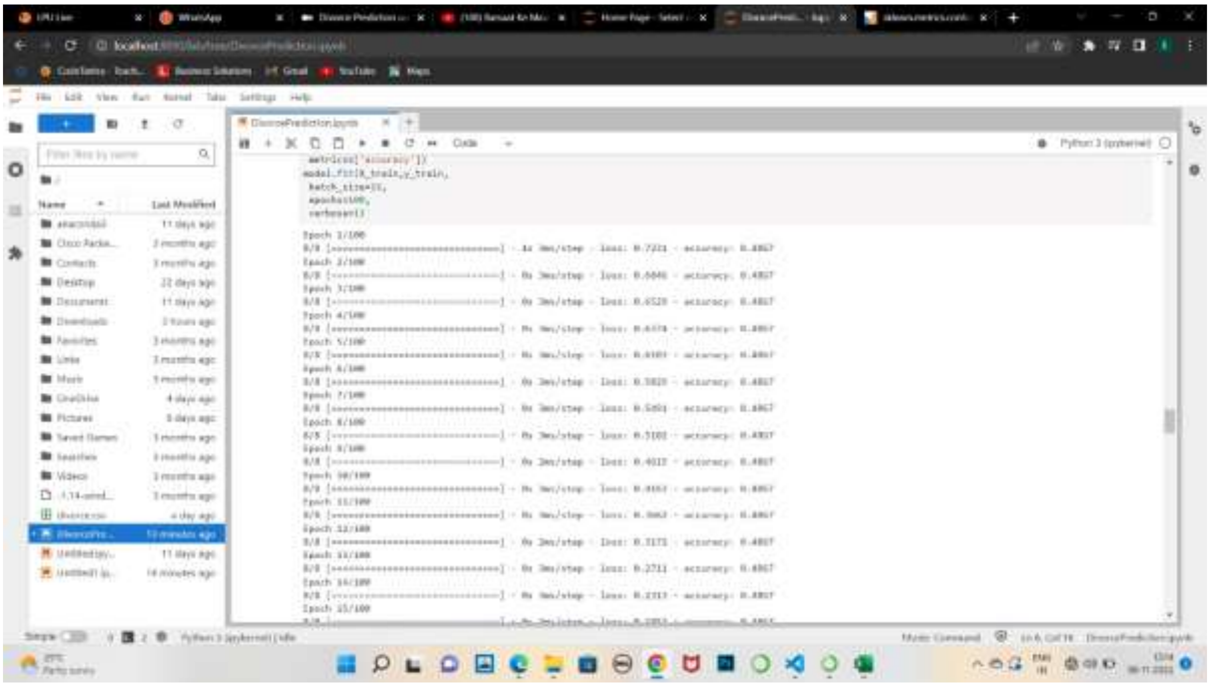
Neural Network

```
import keras
import keras
from keras.datasets import mnist
from keras.models import Sequential,model_from_json
from keras.layers import Dense
from keras.optimizers import RMSprop
```

```
model = Sequential()
model.add(Dense(54, activation='sigmoid', input_shape=(54,)))
model.add(Dense(128, activation='sigmoid'))
model.add(Dense(32, activation='sigmoid'))
model.add(Dense(1, activation='softmax'))
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 54)                2970

 dense_1 (Dense)             (None, 128)               7040

 dense_2 (Dense)             (None, 32)                4128

 dense_3 (Dense)             (None, 1)                 33

=================================================================
Total params: 14,171
Trainable params: 14,171
Non-trainable params: 0
```

```
model.compile(loss=’binary_crossentropy’,
 optimizer=RMSprop(),
 metrics=[‘accuracy’])model.fit(X_train,y_train,
 batch_size=16,
 epochs=100,
 verbose=1)
```

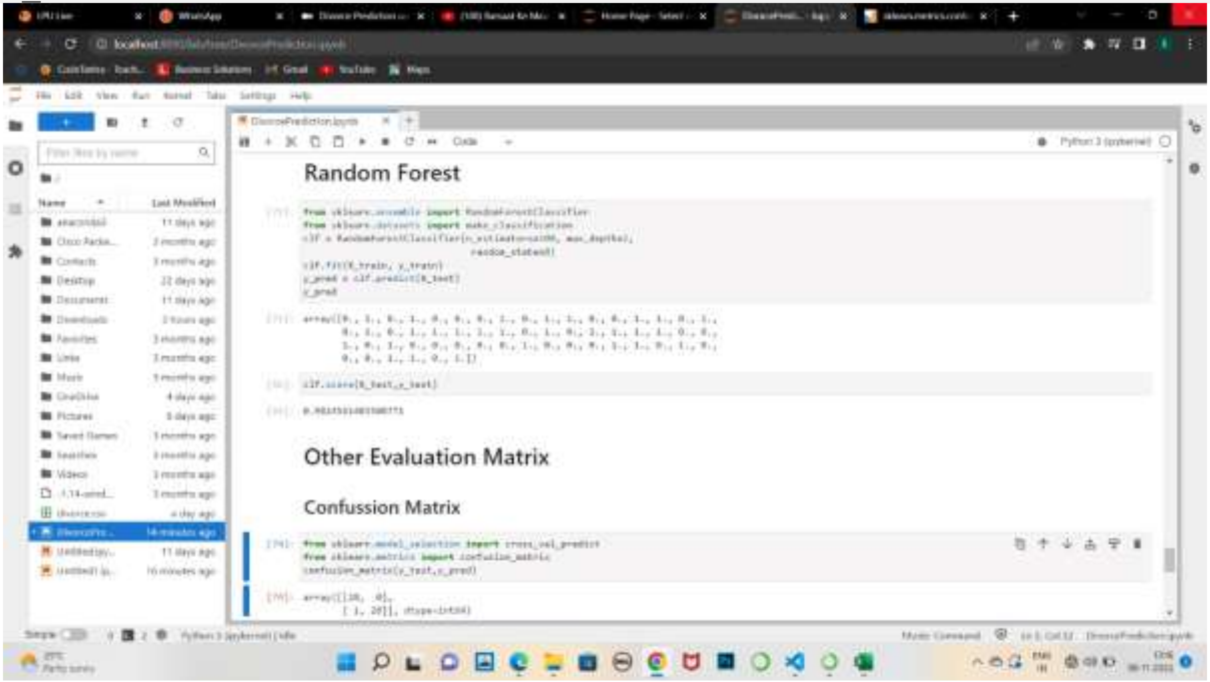


```
metrics=['accuracy'])
model.fit(X_train,y_train,
 batch_size=[illegible],
 epochs=100,
 verbose=1)
```

```
Epoch 1/100
8/8 [==============================] - 1s 3ms/step - loss: 0.7221 - accuracy: 0.4867
Epoch 2/100
8/8 [==============================] - 0s 3ms/step - loss: 0.6846 - accuracy: 0.4867
Epoch 3/100
8/8 [==============================] - 0s 3ms/step - loss: 0.6528 - accuracy: 0.4867
Epoch 4/100
8/8 [==============================] - 0s 3ms/step - loss: [illegible] - accuracy: [illegible]
Epoch 5/100
8/8 [==============================] - 0s 3ms/step - loss: [illegible] - accuracy: [illegible]
Epoch 6/100
8/8 [==============================] - 0s 3ms/step - loss: 0.5820 - accuracy: 0.4867
Epoch 7/100
8/8 [==============================] - 0s 3ms/step - loss: [illegible] - accuracy: [illegible]
Epoch 8/100
8/8 [==============================] - 0s 3ms/step - loss: 0.5102 - accuracy: 0.4867
Epoch 9/100
8/8 [==============================] - 0s 2ms/step - loss: 0.4612 - accuracy: 0.4867
Epoch 10/100
8/8 [==============================] - 0s 3ms/step - loss: [illegible] - accuracy: [illegible]
Epoch 11/100
8/8 [==============================] - 0s 3ms/step - loss: [illegible] - accuracy: [illegible]
Epoch 12/100
8/8 [==============================] - 0s 2ms/step - loss: 0.3171 - accuracy: 0.4867
Epoch 13/100
8/8 [==============================] - 0s 3ms/step - loss: 0.2711 - accuracy: 0.4867
Epoch 14/100
8/8 [==============================] - 0s 3ms/step - loss: 0.2313 - accuracy: 0.4867
Epoch 15/100
```

# Random Forest

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import make_classificationclf =
RandomForestClassifier(n_estimators=100, max_depth=2,
                              random_state=0))clf.fit(X_train,
y_train)
```



## Random Forest

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import make_classification
clf = RandomForestClassifier(n_estimators=100, max_depth=2,
                             random_state=0)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
y_pred
```

```
array([[illegible]])
```

```
clf.score(X_test,y_test)
```

```
[illegible]
```

## Other Evaluation Matrix

### Confussion Matrix

```
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

```
array([[28,  0],
       [ 1, 20]], dtype=int64)
```

The first row is about the not-divorced-predictions: 28 **couples were correctly classified as not divorced** (called true negatives) and 0 **where wrongly classified as not divorced** (false positives).

The second row is about the divorced-predictions: 1 couples where wrongly classified as divorced (false negatives) and 28 were correctly classified as divorced (true positives).

A confusion matrix gives you a lot of information about how well your model does, but there’s a way to get even more, like computing the classifiers precision.

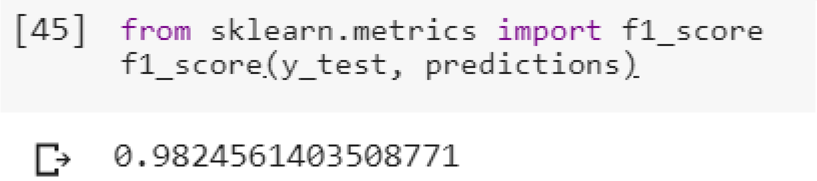
## Precision and Recall

Recall literally is how many of the true positives were recalled. Precision is how many of the returned hits were true positives.

```
from sklearn.metrics import precision_score,
recall_scoreprint(“Precision:”, precision_score(y_test,
predictions))
print(“Recall:”,recall_score(y_test, predictions))
```

## F1 Score

```
from sklearn.metrics import f1_score
f1_score(y_test, predictions)
```



```
[45] from sklearn.metrics import f1_score
     f1_score(y_test, predictions)

     0.9824561403508771
```

Thus ,we have successfully discussed about the various machine learning approaches through which divorces could be predicted and have also seen different methods which measure how well we have predict it.

# S.W.O.T-

## **Strength**-

If we have a dataset of various causes of divorce. Then using this predictive model one can easily predict the chances of divorce. Accuracy of this model is very good.

## **Weakness**-

Neural network model shows very less accuracy  for this dataset.

### **Opportunities**-

The Four Horsemen are four communication habits that increase the likelihood of divorce, according to research by psychologist and renowned marriage researcher John Gottman, Ph. D. Those four behaviors are **criticism, defensiveness, stonewalling, and contempt**

### **Threats**-

There is no threats in this prediction model.

## CONCLUSION:

The prediction of divorce by using machine learning and ensemble learning techniques is the core motive of this research study. The findings of our study are based on key indicators for divorce and the factors that are most significant when predicting divorce. The support vector machine (SVM), passive aggressive classifier, and neural network (MLP) are applied to predict divorce. The cross-validation and performance evaluation techniques are manipulated to evaluate the proposed models. Our EL proposed technique achieved the highest accuracy of 100%. In the context of limitations and future directions, we will try to enhance the questionnaire dataset by adding more questions to get more clarified results and also apply the data augmentation techniques. To reduce overfitting, we will explore different deep learning models.