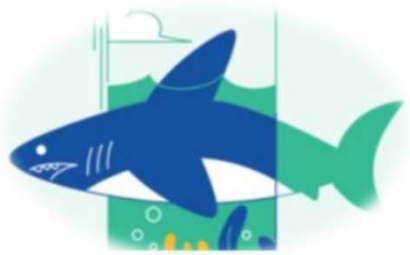
PERSONALIZED DIET RECOMMENDATION SYSTEM IN HEALTHCARE USING PREDICTIVE APPROACHES

BY: FRASON FRANCIS



Shark Hacks-3

INTRODUCTION

- Wide variety of ingredients, cultures and personal tastes makes decision about what to eat a great problem.

- Many diseases that were previously thought as hereditary are now seen to be connected to biological disfunction related to nutrition.

CONT....

- Being healthy and eating better is something the vast majority of the population wants and doing so usually requires great effort.

- The working prototype accomplishes a **Personalized Diet Recommendation System** with integration of **Machine Learning Algorithms** to recommend the right food at right time and with the right nutrition, calories, fat etc.

OBJECTIVE

- To establish working prototype of a Personalized Diet Recommendation System.

EXISTING SYSTEM

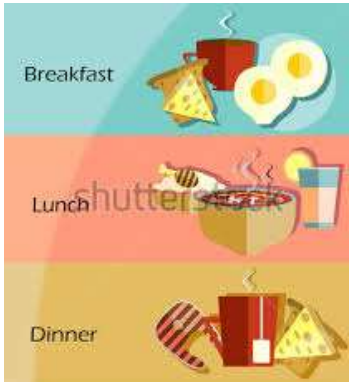
- The existing working model of the Diet Recommendation System gives recommendations concerning food based on user inputs in general life style on regular food timings.

CONT….

- The food is recommended based on the user habits of food intake at a particular term according to his/her tastes.

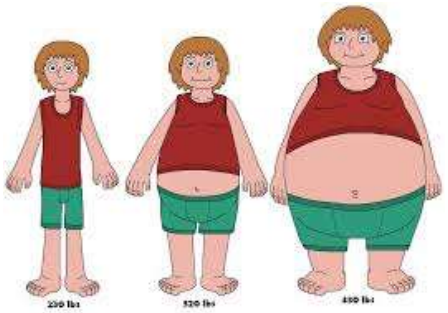
ISSUES

- The issues that are being faced by the current working model is it does not facilitate a user with the food classification based on the food timings on a daily basis other than general food timings.

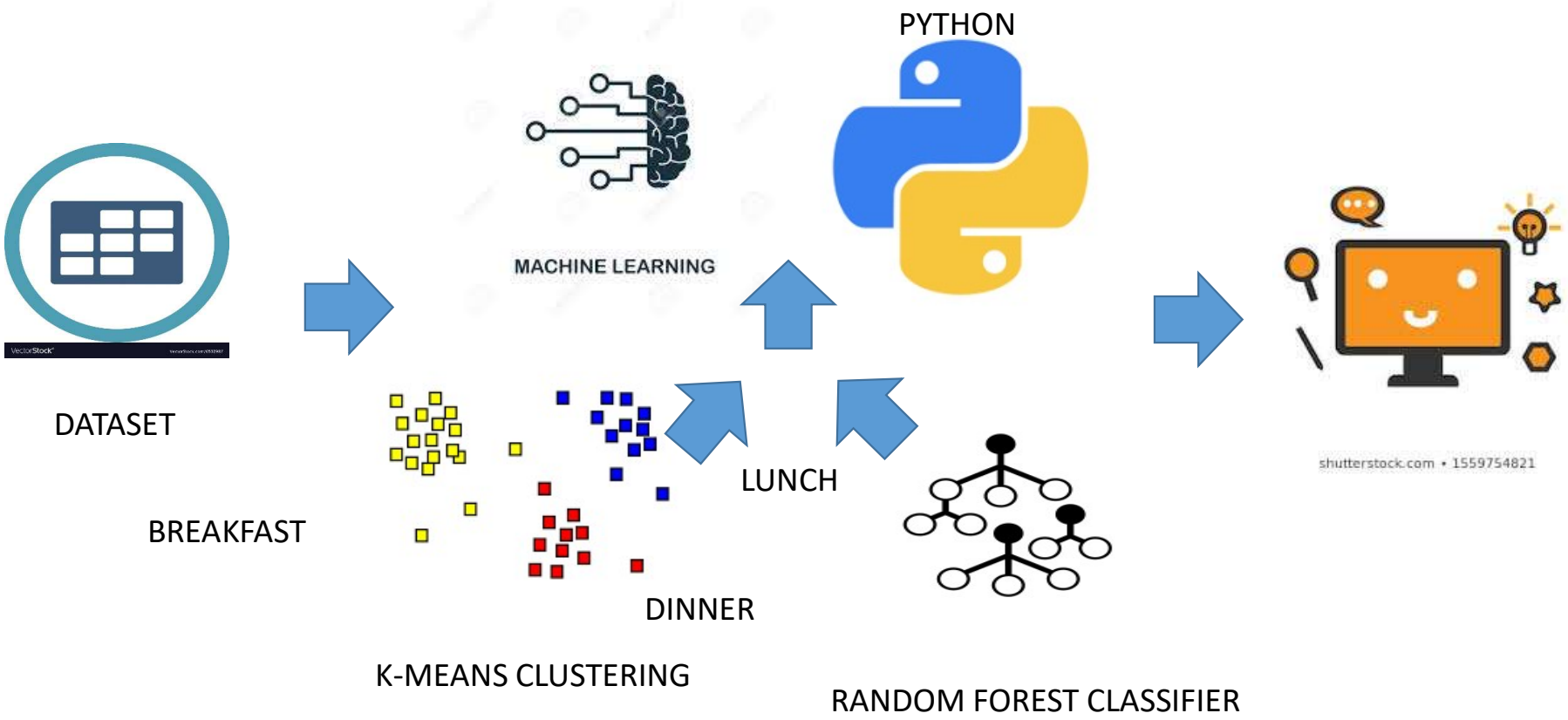


CONT….

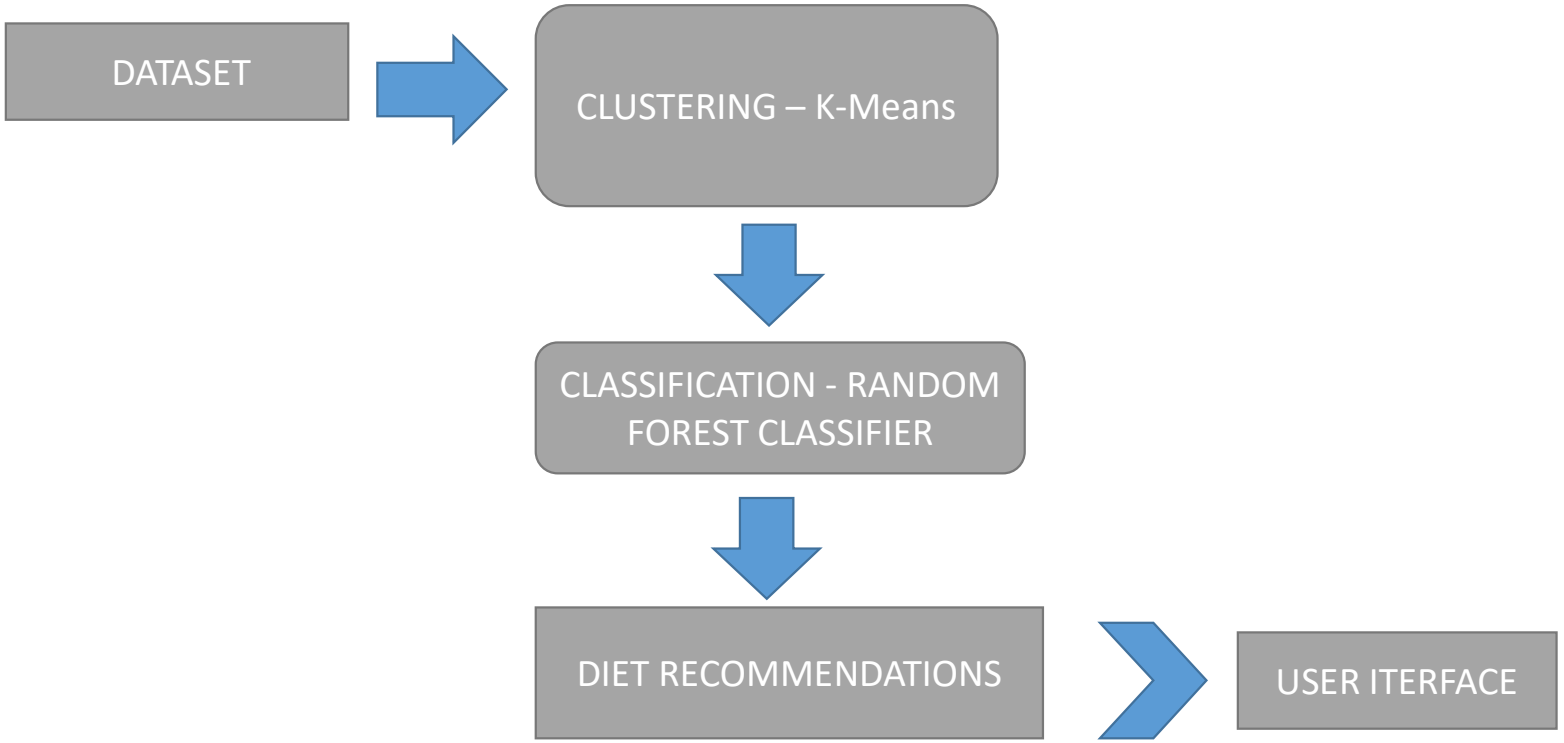
- The existing system struggles to provide a weight gain/loss scheme to a user based on his long term food habits.



SYSTEM ARCHITECTURE



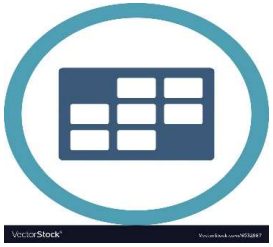
SYSTEM WORKFLOW



.

MODULES

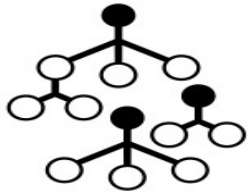
- **DATASET**



DATASET – Type of food with all nutritional values

CONT….

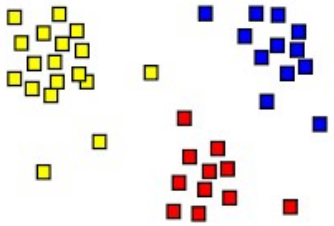
- **MACHINE LEARNING MODEL:**
  - **DATA CLASSIFICATION**



RANDOM FORETS CLASSIFICATION

CONT….

- **CLUSTERING**



K-MEANS CLUSTERING

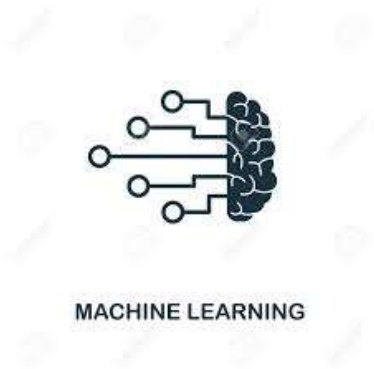
CONT....

- **GUI – PYTHON (using TKINTER)**

GUI

CONT….

- **TECH STACK AND TOOLS USED**



IMPLEMENTATION PROCEDURE

- For training of the system , the initial process involves the segregation of food items depending upon the meal for which they are consumed i.e Breakfast , Lunch and Dinner.

- The clustering of various nutrients depending upon which are essential for the weightloss , weightgain and healthy is performed.

- After the clustering is performed , using Random Forest classifier , the nearest food items are predicted which best suites for the appropriate diet.

CONT….

- As part of user interface , the inputs needed from the user are Age , Height , Weight and what the purpose for which the diet is required.

- Depending upon it , from the appropriate clustering , specific food items are classified and recommended to the user.

TECH STACK AND TOOLS

Python 3.6

Tested on Py-Charm IDE/Spyder

CODE:

```
Breakfastdata=data['Breakfast']
    BreakfastdataNumpy=Breakfastdata.to_numpy()

    Lunchdata=data['Lunch']
    LunchdataNumpy=Lunchdata.to_numpy()

    Dinnerdata=data['Dinner']
    DinnerdataNumpy=Dinnerdata.to_numpy()

    Food_itemsdata=data['Food_items']
    breakfastfoodseparated=[]
    Lunchfoodseparated=[]
    Dinnerfoodseparated=[]

    breakfastfoodseparatedID=[]
    LunchfoodseparatedID=[]
    DinnerfoodseparatedID=[]

    for i in range(len(Breakfastdata)):
      if BreakfastdataNumpy[i]==1:
        breakfastfoodseparated.append(Food_itemsdata[i])
        breakfastfoodseparatedID.append(i)
```

CONT....

```
for i in range(len(Breakfastdata)):
    if BreakfastdataNumpy[i]==1:
      breakfastfoodseparated.append(Food_itemsdata[i])
      breakfastfoodseparatedID.append(i)
    if LunchdataNumpy[i]==1:
      Lunchfoodseparated.append(Food_itemsdata[i])
      LunchfoodseparatedID.append(i)
    if DinnerdataNumpy[i]==1:
      Dinnerfoodseparated.append(Food_itemsdata[i])
      DinnerfoodseparatedID.append(i)

  #print ('BREAKFAST FOOD ITEMS')
  #print (breakfastfoodseparated)
  #print ('LUNCH FOOD ITEMS')
  #print (Lunchfoodseparated)
  #print ('DINNER FOOD ITEMS')
  #print (Dinnerfoodseparated)
```

CONT....

- # retrieving rows by loc method |
- LunchfoodseparatedIDdata = data.iloc[LunchfoodseparatedID]
- print(LunchfoodseparatedID)
- LunchfoodseparatedIDdata=LunchfoodseparatedIDdata.T
- val=list(np.arange(5,15))
- Valapnd=[0]+val
- LunchfoodseparatedIDdata=LunchfoodseparatedIDdata.iloc[Valapnd]
- LunchfoodseparatedIDdata=LunchfoodseparatedIDdata.T
- #print (LunchfoodseparatedIDdata)
- # retrieving rows by loc method
- breakfastfoodseparatedIDdata = data.iloc[breakfastfoodseparatedID]
- breakfastfoodseparatedIDdata=breakfastfoodseparatedIDdata.T
- val=list(np.arange(5,15))
- Valapnd=[0]+val
- breakfastfoodseparatedIDdata=breakfastfoodseparatedIDdata.iloc[Valapnd]
- breakfastfoodseparatedIDdata=breakfastfoodseparatedIDdata.T
- #print (breakfastfoodseparatedIDdata)

CONT....

```
age=int(e1.get())
    veg=float(e2.get())
    weight=float(e3.get())
    height=float(e4.get())
    bmi = weight/(height**2)
    agewiseinp=0

    for lp in range (0,80,20):
        test_list=np.arange(lp,lp+20)
        for i in test_list:
            if(i == age):
                print('age is between',str(lp),str(lp+10))
                tr=round(lp/20)
                agecl=round(lp/20)
```

CONT....

```
#conditions
print("Your body mass index is: ", bmi)
if ( bmi < 16):
    print("severely underweight")
    clbmi=4
elif ( bmi >= 16 and bmi < 18.5):
    print("underweight")
    clbmi=3
elif ( bmi >= 18.5 and bmi < 25):
    print("Healthy")
    clbmi=2
elif ( bmi >= 25 and bmi < 30):
    print("overweight")
    clbmi=1
elif ( bmi >=30):
    print("severely overweight")
    clbmi=0
```

CONT...

- import matplotlib.pyplot as plt
- Datacalorie=LunchfoodseparatedIDdata[1:,1:len(LunchfoodseparatedIDdata)]
- #print(Datacalorie)
- X = np.array(Datacalorie)
- kmeans = KMeans(n_clusters=3, random_state=0).fit(X)
- print ('## Prediction Result ##')
- print(kmeans.labels_)
- XValu=np.arange(0,len(kmeans.labels_))
- # fig,axs=plt.subplots(1,1,figsize=(15,5))
- # plt.bar(XValu,kmeans.labels_)
- lnchlbl=kmeans.labels_
- # plt.title("Predicted Low-High Weigted Calorie Foods")

RESULT

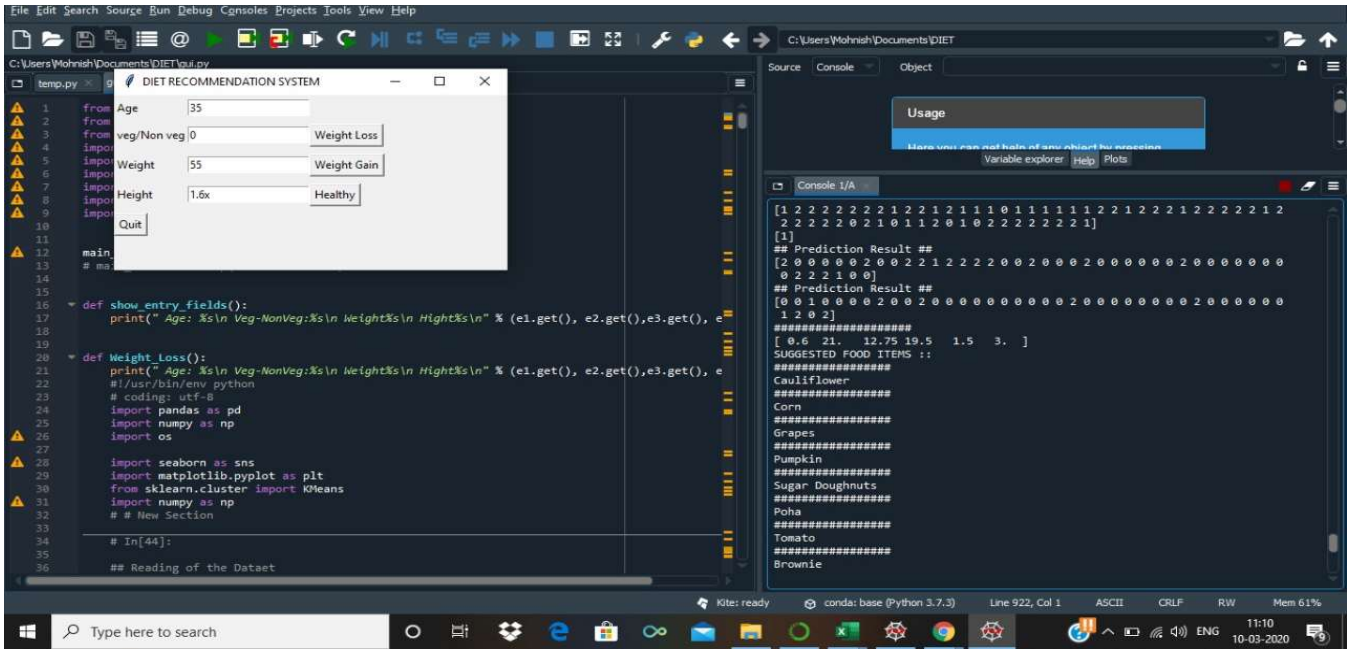
- A working prototype of a Diet Recommendation System is established.
- The module works on the basis of K-Means Clustering and Random Forest Classification Algorithms.
- Tkinter based GUI is implemented.

CONT….

- **Working GUI**

CONT....

- **Working Module**



CONT….

- Diet is recommended based on the user’s input and also the desired food list is displayed.
- BMI’ is also calculated and taken into consideration.
- Weight Loss/Gain/Healthy diet category is also predicted.

CONT....

- **FUTURE SCOPE:**
  - The module can be implemented as a cloud based application.
  - Packaged as a single entity, ready for production environment deployment .

CONCLUSION

- A Diet Recommendation System is implemented with the working functionalities like:
  - Desired food list prediction.
  - Weight category prediction.
  - BMI Calculation.

- Health is vital for an individual and can be achieved with this working module. Thus making life healthy.

✔ THANK YOU ☺