PRAKRITI - DATA ANALYTICS FOOD FOR FUTURE

Submitted by,

Team Name - **AbracaData**

Tushar Mohta
Aayush Poddar
Srimahn V

IIT Kharagpur

PROBLEM STATEMENT - Food For Future

- There has been substantial food demand growth in the recent years and this has posed a crucial challenge of catering to all food demand in the near future.
- Innovation and intelligence with proper usage of available resources is the need of the hour
- To overcome this challenge, Eating and Diet habits of general people needs to be observed

According to the given Problem Statement, an Exploratory Data Analysis(EDA) approach comprising of analysis and visualisation of dataset is taken. Finally on the processed dataset, we will apply various classification models to predict Course type and then compare between our accuracies across these models.

Name	Ingradients	Diet	Preparation Time	Cooking Time	Flavor	Course	State	Region	Price/unit
Balu shahi	Maida flour, yogurt, (	vegetarian	45	25	sweet	dessert	West Bengal	East	260
Boondi	Gram flour, ghee, su	vegetarian	80	30	sweet	dessert	Rajasthan	West	270
Gajar ka halwa	Carrots, milk, sugar, g	vegetarian	15	60	sweet	dessert	Punjab	North	450
Ghevar	Flour, ghee, kewra, n	vegetarian	15	30	sweet	dessert	Rajasthan	West	460
Gulab jamun	Milk powder, plain fl	vegetarian	15	40	sweet	dessert	West Bengal	East	300
Imarti	Sugar syrup, lentil flo	vegetarian	10	50	sweet	dessert	West Bengal	East	270
Jalebi	Maida, corn flour, ba	vegetarian	10	50	sweet	dessert	Uttar Pradesh	North	260
Kaju katli	Cashews, ghee, card	vegetarian	10	20	sweet	dessert			400
Kalakand	Milk, cottage cheese	vegetarian	20	30	sweet	dessert	West Bengal	East	450
Kheer	Milk, rice, sugar, drie	vegetarian	10	40	sweet	dessert			500
Laddu	Gram flour, ghee, su	vegetarian	10	40	sweet	dessert			450
Lassi	Yogurt, milk, nuts, su	vegetarian	5	5	sweet	dessert	Punjab	North	260
Nankhatai	Refined flour, besan,	vegetarian	20	30	sweet	dessert			250
Petha	Firm white pumpkin,	vegetarian	10	30	sweet	dessert	Uttar Pradesh	North	200

Sample of data set and its features

DATA PRE-PROCESSING

Missing values

Simply dropping all incomplete rows will result in around 30% data loss.

#	Column	Non-Null Count	Dtype
0	Name	255 non-null	object
1	Ingradients	255 non-null	object
2	Diet	255 non-null	object
3	Preparation Time	225 non-null	float64
4	Cooking Time	227 non-null	float64
5	Flavor	226 non-null	object
6	Course	255 non-null	object
7	State	231 non-null	object
8	Region	241 non-null	object
9	Price/unit	255 non-null	int64

#	Column	Non-Null Count
0	Name	180 non-null
1	Ingradients	180 non-null
2	Diet	180 non-null
3	Preparation Time	180 non-null
4	Cooking Time	180 non-null
5	Flavor	180 non-null
6	Course	180 non-null
7	State	180 non-null
8	Region	180 non-null
9	Price/unit	180 non-null

Filling preparation and cooking times

- Missing preparation and cooking times are filled with the median after grouping by region and diet
- Median is used to avoid outlier effect
- Grouping by price or other features is not done, as they are uncorrelated

	Diet	Region	Preparation Time	Cooking Time
0	non vegetarian	East	12.5	37.5
1	non vegetarian	North	120.0	35.0
2	non vegetarian	North East	10.0	22.5
3	non vegetarian	South	10.0	60.0
4	non vegetarian	West	10.0	40.0
5	vegetarian	Central	10.0	45.0
6	vegetarian	East	20.0	42.5
7	vegetarian	North	15.0	40.0
8	vegetarian	North East	10.0	30.0
9	vegetarian	South	10.0	30.0
10	vegetarian	West	10.0	30.0

Missing flavor values

- Flavor is missing in 27 rows
- Dataset is dominated by spicy and sweet flavor, while sour and bitter have negligible appearance
- Missing flavor values are filled by randomly drawing values from a list of flavors, which emulates the frequency of available flavor categories

```
def fill_freq(freq):
  dct={}
  lst=[]
  curr=0
  for idx,flav in enumerate(freq.index):
    dct[curr]=flav
    ct=freq[idx]
    lst=lst+[curr]*ct
    curr+=1
  rnd.shuffle(lst)
  return [dct,lst]

pair=fill_freq(data['Flavor'].value_counts().sort_values(ascending=True))

for idx,x in enumerate(data["Flavor"].isna()):
  if(x):
    temp=pair[1][rnd.randint(0,len(pair[1])-1)]
    data['Flavor'][idx]=pair[0][temp]
```

Ingredients feature extraction

- ➔ Initially, there are 352 unique ingredient values
- ➔ Same ingredients are present with a variety of names

	red chili	red chilli	red chillies
count	1	4	1

- ➔ Similar ingredients are matched using Fuzzy string matching algorithm
- ➔ Some matches:

```
'biryani masala': 'garam masala',
'biryani masala powder': 'garam masala',
'black sesame seeds': 'sesame seeds',
'boiled potatoes': 'potatoes',
'cardamom pods': 'cardamom',
'carrots': 'carrot',
'cashews': 'cashew nuts',
'chana daal': 'chana dal',
```

Ingredients feature extraction (cont.)

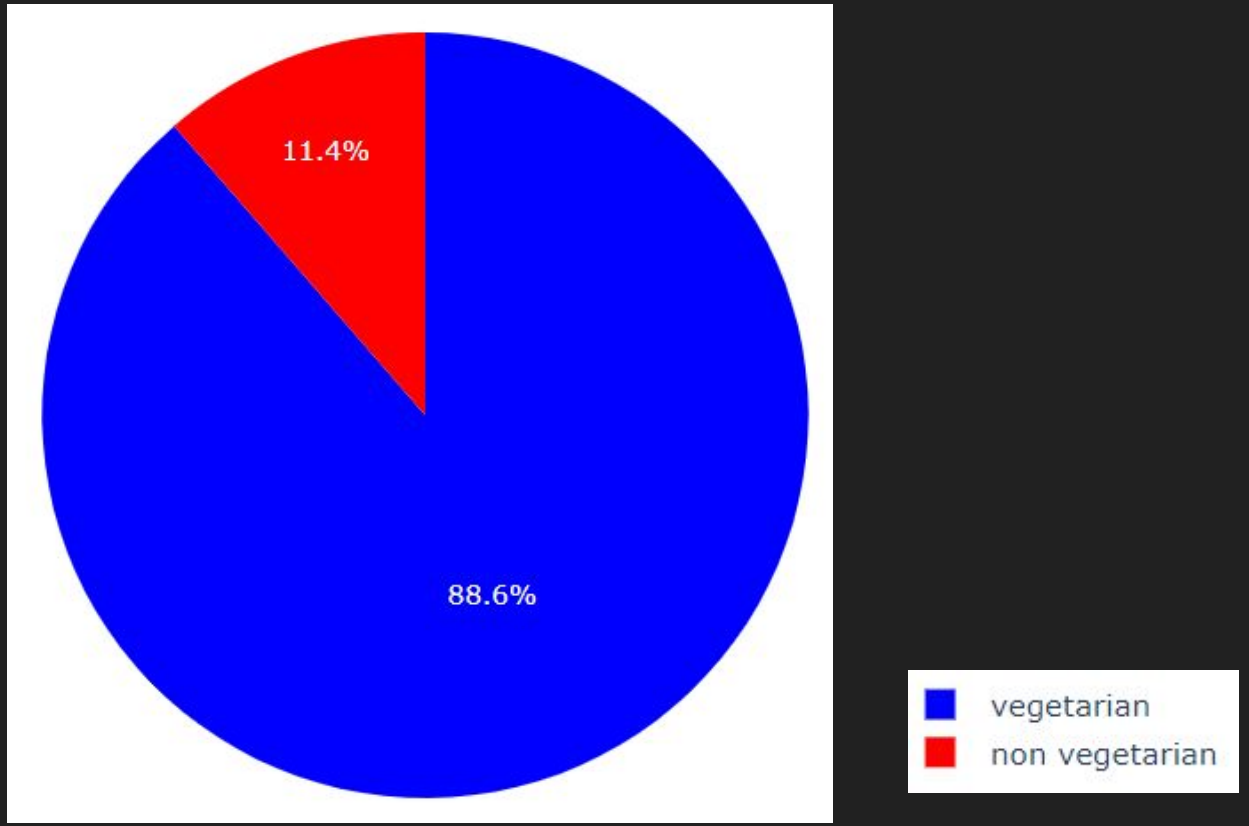
- ➔ After matching, there are 270 unique ingredients
- ➔ Ingredients are then one-hot encoded

	maida flour	yogurt	oil	sugar	gram flour	ghee	carrot	milk	cashew nuts
0	1	1	1	1	0	0	0	0	0
1	0	0	0	1	1	1	0	0	0
2	0	0	0	1	0	1	1	1	1

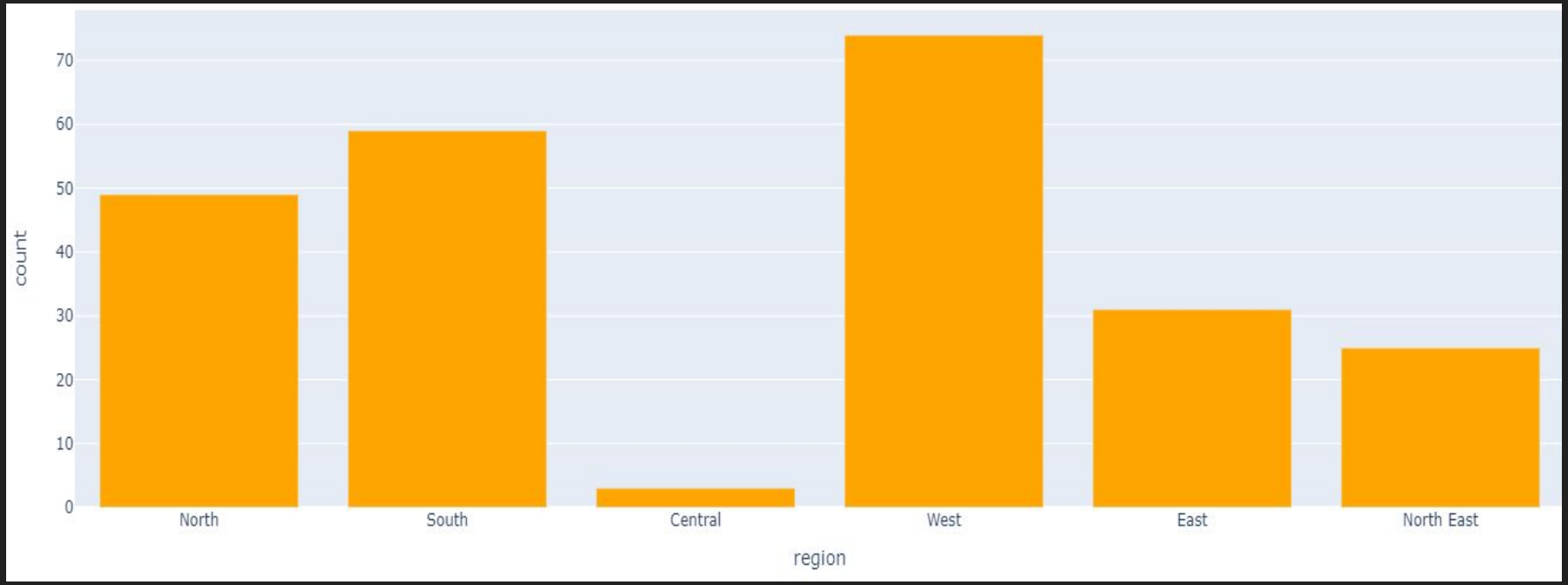
- ➔ PCA was applied with 90% explained variance retained
- ➔ PCA reduced the number of components to only 99
- ➔ The dataset is now ready for modelling!

VISUALISATIONS

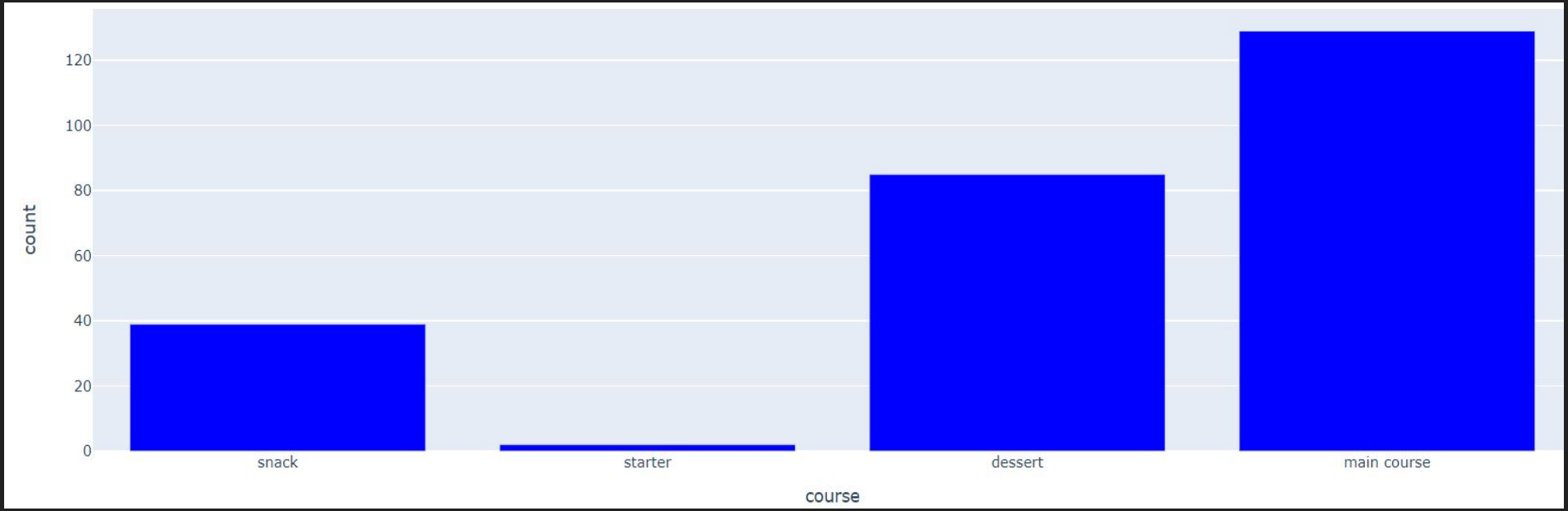
Veg vs Non-veg



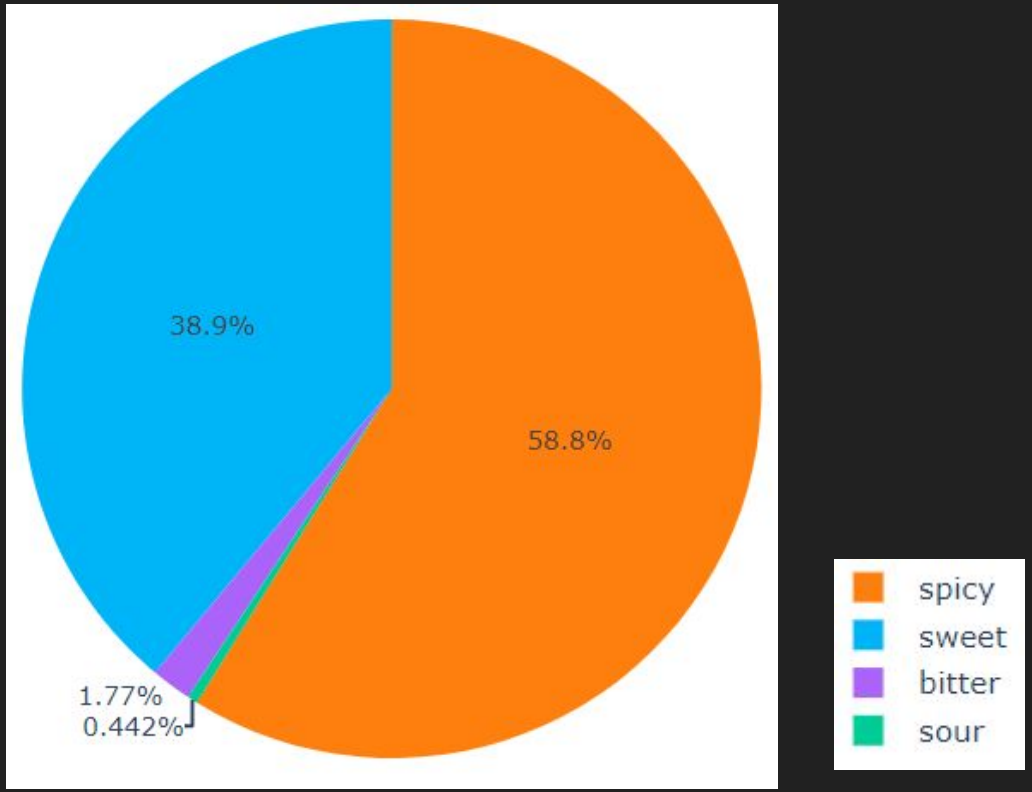
Dishes from different regions



Dishes of different courses

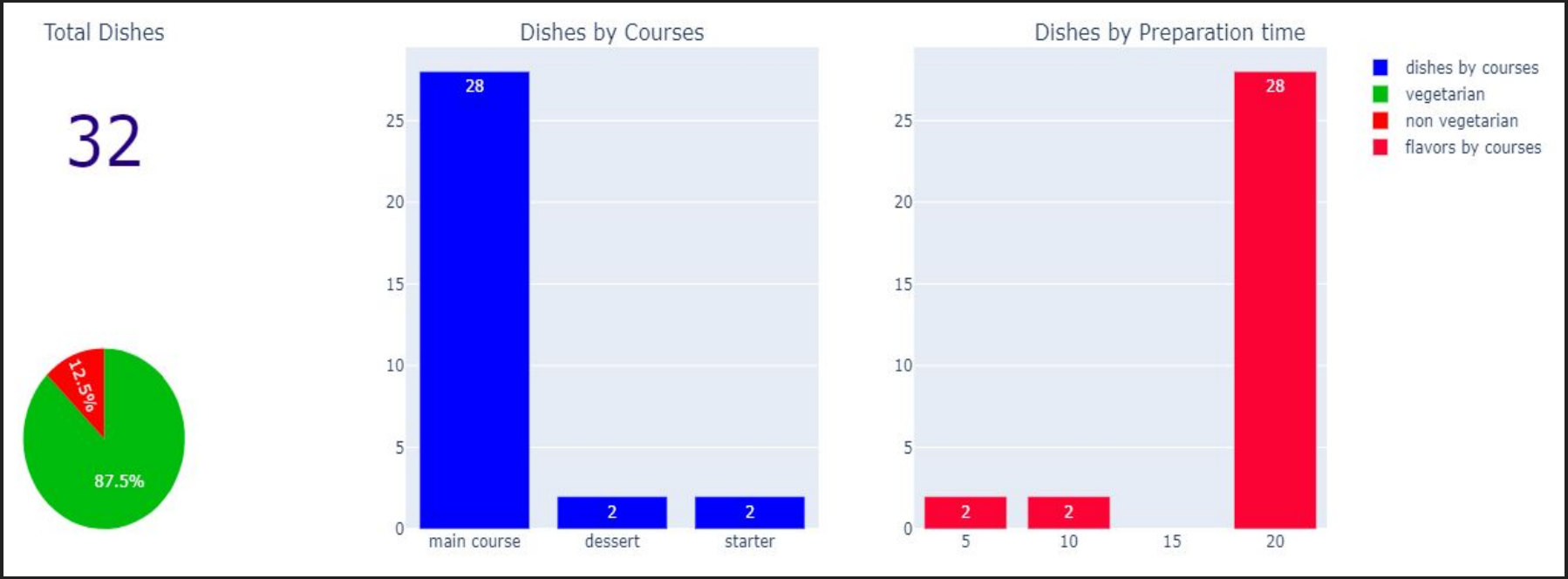


Distribution of flavors



West Bengal Food Infographic

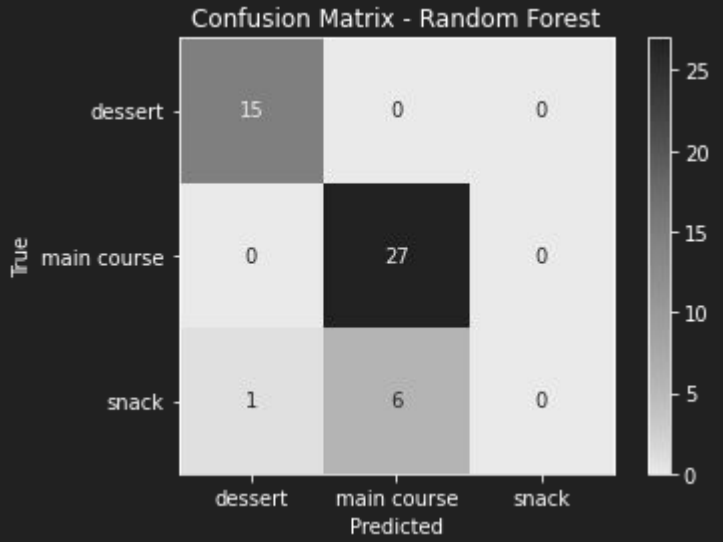
Punjab Food Infographic



PREDICTING COURSE

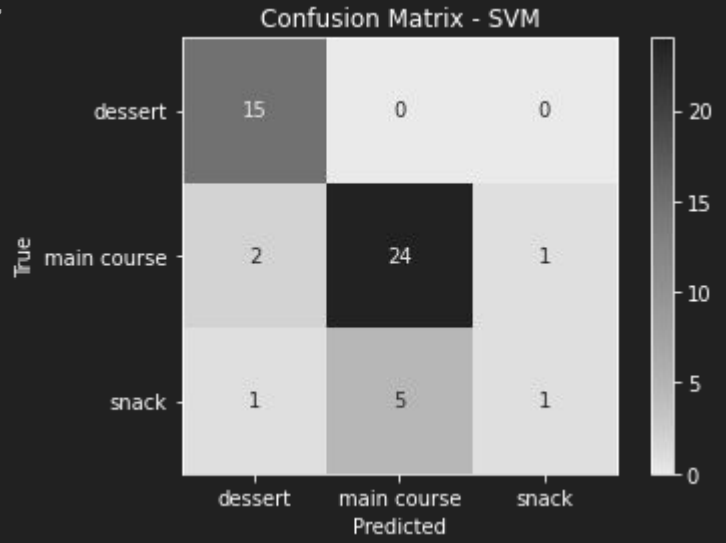
Random forest

- Overall accuracy = 85.72%
- Producer accuracy = 85.72%
- User accuracy = 73.78%
- Kappa coefficient = 0.73



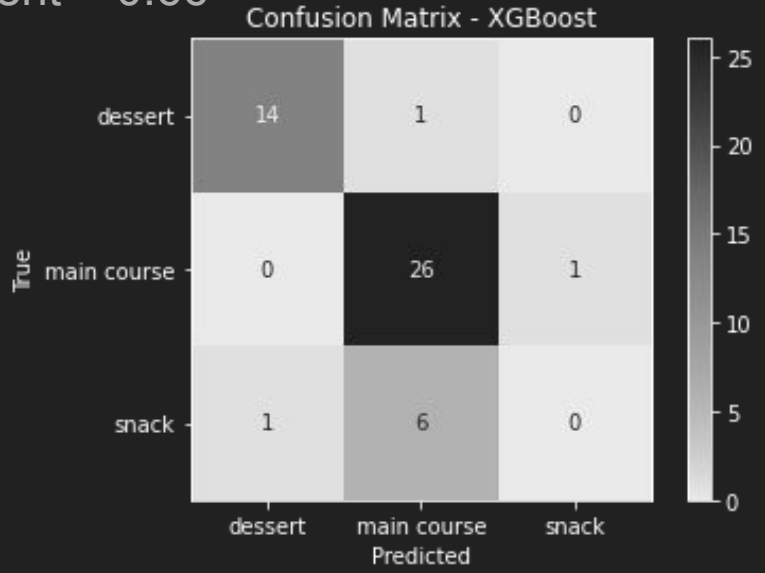
SVM

- Gaussian kernel works best
- Overall accuracy = 83.67%
- Producer accuracy = 83.67%
- User accuracy = 80.01%
- Kappa coefficient = 0.67



XGBoost

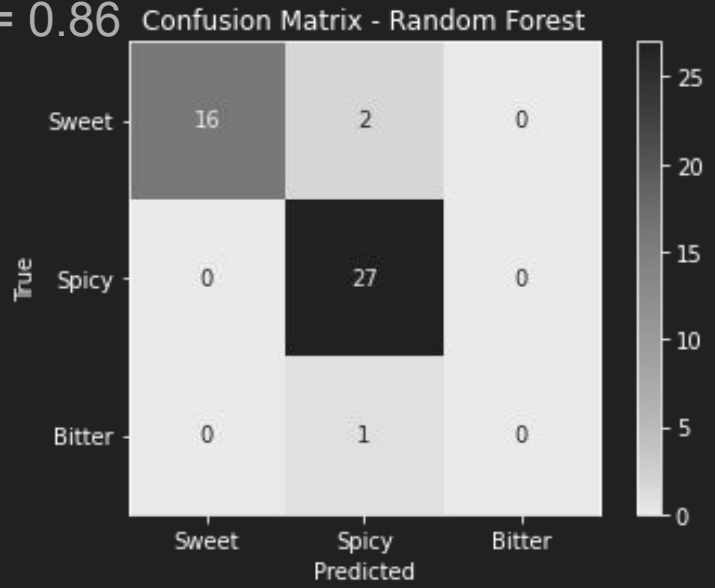
- Overall accuracy = 81.63%
- Producer accuracy = 81.63%
- User accuracy = 73.13%
- Kappa coefficient = 0.66



PREDICTING FLAVOR

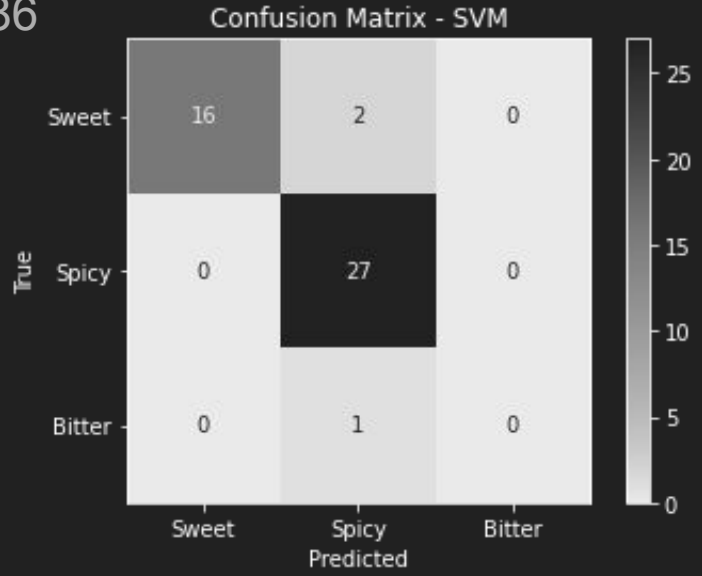
Random Forest

- Overall accuracy = 93.47%
- Producer accuracy = 93.47%
- User accuracy = 91.95%
- Kappa coefficient = 0.86



SVM

- Gaussian kernel works best
- Overall accuracy = 93.47%
- Producer accuracy = 93.47%
- User accuracy = 91.96%
- Kappa coefficient = 0.86



XGBoost

- Overall accuracy = 93.47%
- Producer accuracy = 93.47%
- User accuracy = 91.96%
- Kappa coefficient = 0.86

Thank You!