Day 44

DIY

Q1. Problem Statement: Dimensionality Reduction

Write a Python program that reads the *dairy_product.csv* (provided on LMS) file into a DataFrame, the followinag are the tasks that are to be taken into consideration while reducing the dimensions of data.

1. Load the *mobile dairy_product.csv* data into a DataFrame
2. Find missing values and drop columns having more than 80% missing data
3. Do label encoding for categorical features
4. Extract independent variables (Xs) and dependent variables (Ys) into separate data objects and drop unwanted columns like “ID”
5. Print low-variance data
6. Use random forest and print important features as per their value
7. Based on the high correlation, drop highly correlated columns, as you can find.

Dataset:

	ID	FoodGroup	ShortDescrip	Descrip	CommonName	MfgName	ScientificName	Energy_kcal	Protein_g	Fat_g	...	Folate_USRDA	Niacin_USRDA	Riboflavin_USRDA	Thiamin_USRDA	Calcium_USRDA	Copper_USRDA	Magnesium_USRDA	Phosphorus_USRDA	Selenium_USRDA	Zinc_USRDA
0	1001	Dairy and Egg Products	BUTTER,WITH SALT	Butter, salted	NaN	NaN	NaN	717.0	0.85	81.11	...	0.0075	0.002625	0.026154	0.004167	0.020000	0.000000	0.004762	0.034286	0.018182	0.008182
1	1002	Dairy and Egg Products	BUTTER,WHIPPED,WITH SALT	Butter, whipped, with salt	NaN	NaN	NaN	717.0	0.85	81.11	...	0.0075	0.002625	0.026154	0.004167	0.020000	0.000018	0.004762	0.032857	0.018182	0.004545
2	1003	Dairy and Egg Products	BUTTER OIL,ANHYDROUS	Butter oil, anhydrous	NaN	NaN	NaN	876.0	0.28	99.48	...	0.0000	0.000188	0.003846	0.000833	0.003333	0.000001	0.000000	0.004286	0.000000	0.000909
3	1004	Dairy and Egg Products	CHEESE,BLUE	Cheese, blue	NaN	NaN	NaN	353.0	21.40	28.74	...	0.0900	0.063500	0.293846	0.024167	0.440000	0.000044	0.054762	0.552857	0.263636	0.241818
4	1005	Dairy and Egg Products	CHEESE,BRICK	Cheese, brick	NaN	NaN	NaN	371.0	23.24	29.68	...	0.0500	0.007375	0.270000	0.011667	0.561667	0.000027	0.057143	0.644286	0.263636	0.236364

5 rows × 45 columns

Sample Output:

5. Print low-variance data

ShortDescrip	6.185080e+06
Descrip	6.189878e+06
Energy_kcal	2.869260e+04
Protein_g	1.113149e+02
Fat_g	2.517411e+02
Carb_g	7.419631e+02
Sugar_g	1.850171e+02
Fiber_g	1.860775e+01
VitA_mcg	6.074054e+05
VitB6_mg	2.290714e-01
VitB12_mcg	1.865534e+01
VitC_mg	3.315774e+03
VitE_mg	1.481502e+01
Folate_mcg	3.480488e+04
Niacin_mg	2.337308e+01
Riboflavin_mg	2.023920e-01
Thiamin_mg	2.687065e-01
Calcium_mg	4.054732e+04
Copper_mcg	3.058670e-01
Iron_mg	3.279815e+01
Magnesium_mg	3.143672e+03
Manganese_mg	4.074828e+01
Phosphorus_mg	4.124686e+04
Selenium_mcg	8.004569e+02
Zinc_mg	1.127759e+01
VitA_USRDA	7.498833e-01
VitB6_USRDA	7.926347e-02
VitB12_USRDA	3.238774e+00
VitC_USRDA	4.093548e-01
VitE_USRDA	6.584452e-02
Folate_USRDA	2.175305e-01

6. Use random forest and print important features as per their value

ShortDescrip	0.070306
Descrip	0.080734
Energy_kcal	0.041400
Protein_g	0.040171
Fat_g	0.037234
Carb_g	0.068544
Sugar_g	0.024435
Fiber_g	0.023709
VitA_mcg	0.010369
VitB6_mg	0.018054
VitB12_mcg	0.026663
VitC_mg	0.020575
VitE_mg	0.010687
Folate_mcg	0.018990
Niacin_mg	0.020981
Riboflavin_mg	0.014503
Thiamin_mg	0.022342
Calcium_mg	0.018464
Copper_mcg	0.016315
Iron_mg	0.025404
Magnesium_mg	0.017926
Manganese_mg	0.022947
Phosphorus_mg	0.017631
Selenium_mcg	0.018731
Zinc_mg	0.026315
VitA_USRDA	0.012710
VitB6_USRDA	0.018897
VitB12_USRDA	0.023977
VitC_USRDA	0.021369
VitE_USRDA	0.010253
Folate_USRDA	0.018132

7. Based on high correlation, drop highly correlated columns as much as you can find.

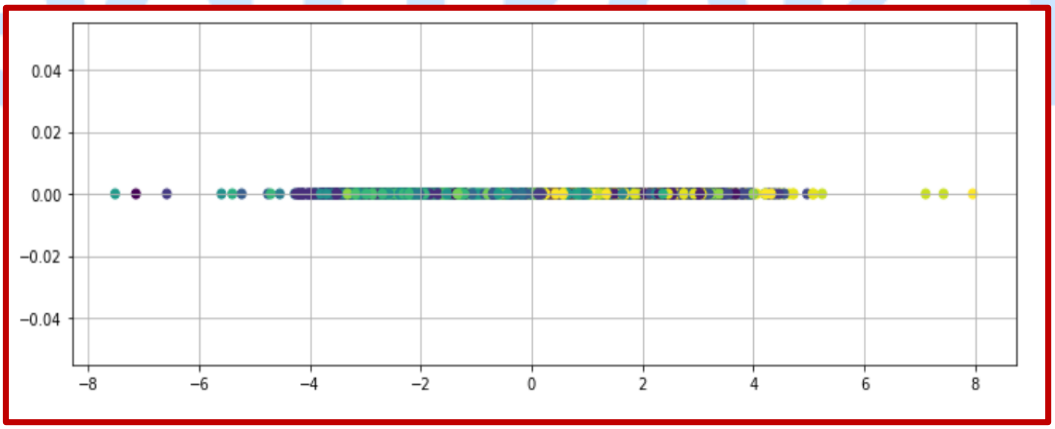
```
final shape of dataFrame is (8618, 39)
```

Q2. Problem Statement: Linear Discriminant Analysis

1. Use the above-separated data of X and Y
2. Standardizes the data
3. Perform LDA
4. Plot the graph of LDA for Test data using seaborn
5. Build a Random forest model and evaluate your data

Sample Output:

4. Plot the graph of LDA for Test data



5. Build a Random forest model and evaluate your data

```
Accuracy: 0.22621809744779584
```