MAKERERE UNIVERSITY

SEMESTER ONE 2024/2025 ACADEMIC YEAR

SCHOOL COMPUTING AND INFORMATICS TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

MASTER OF SCIENCE IN COMPUTER SCIENCE

MCS 7103
Machine Learning

ASSIGNMENT ONE

NAKAYIZA SHAMIM

2024/HD05/21950U

2400721950

EXPLORATORY DATA ANALYSIS REPORT FOR PREDICTING THE MOST APPROPRIATE PRODUCTS WHILE PRESCRIBING MEDICINES.

The Problem

Poor prescription of medicines has led to issues like High rate of drug expiries as doctors tend to only prescribe medicines known to them, low sales since the unknown medicines to doctors are not sold to patients who need them, this makes it hard for the pharmacy business to grow.

Solution

Making prescriptions more efficient using machine learning hence solving the above problems.

Data

The data used was from my workplace, the type of machine learning applied is supervised learning, using classification data in a tabular format.

EXPLORATORY DATA ANALYSIS

Understanding the data.

1. **Question**: Do I have the data required to solve the problem?
   Answer: Yes I do have the dataset as demonstrated in the figure 1 below.

```
[15]: import pandas as pd
```

```
[ ]: # Accessing my data
```

```
[16]: data = pd.read_csv('/home/devsham/Documents/Muk/Prescription Data .csv')
```

```
[21]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 360 entries, 0 to 359
Data columns (total 14 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   Diagnosis                    354 non-null    object
 1   Age Range                    346 non-null    object
 2   age unit                     349 non-null    object
 3   PRODUCT DESCRIPTION/ BRAND   351 non-null    object
 4   ALTERNATIVE PRODUCT 1        321 non-null    object
 5   ALTERNATIVE PRODUCT 2        146 non-null    object
 6   APPROPRIATE ADD ON           72 non-null     object
 7   Comments                     66 non-null     object
 8   Age Range 1                  47 non-null     object
 9   Age Range 2                  15 non-null     object
 10  Age Range.1                  6 non-null      object
 11  Contraindications            3 non-null      object
 12  Unnamed: 12                  1 non-null      object
 13  Unnamed: 13                  1 non-null      object
dtypes: object(14)
memory usage: 39.5+ KB
```

Figure 1

2. **Question**: Are all the parameters Available for me to solve my problem?
   Answer: Yes. The parameters I need to solve my problem are available in my dataset and that is to say: *Diagnosis, Age Range, Product/Description/Brand and Alternative product 1, and 2. This means that the rest of the columns will be dropped since they are not required.*

```
[19]: data.head()
```

[19]:

	Diagnosis	Age Range	age unit	PRODUCT DESCRIPTION/ BRAND	ALTERNATIVE PRODUCT 1	ALTERNATIVE PRODUCT 2	APPROPRIATE ADD ON	Comments	Age Range 1	Age Range 2	Age Range.1	Contraindi
0	Dry Cough	>=12	years	Benylin Dry Cough (Dextromethorphan)	Delased dry cough (Diphen + Dextrom + Sodium C...	Zedex (Dextro, Bromhexin, Ammonium Chloride + ...	NaN	Sedation is a common side effect among options...	Recommended from age 2	NaN	NaN	
1	Dry Cough	>=12	years	Brochophane (Dextrom + Diphenhydramine + Ephe...	Menthodex (Ammonium chloride, Sodium Citrate, ...	NaN	NaN	Mixed coughs	From 2 years and above	NaN	NaN	Risk of hig
2	Dry Cough	2-5	years	Benylin Peadiatric (Dextromethorphan + Sodium ...	Delased Peadiatric (Sodium Citrate + Diphenhyd...	Piritex baby (Acetic acid 26.35mg/5mL)	NaN	Irritating / Allergic Coughs	Atleast 2 years for Benylin & Delased Paed	Pirtitex baby from 3 months	Piritex Junior from 1 year	
3	Dry Cough	>=12	years	Hydrllin DM (Diphen + Ammonium Chloride+ Ment...	Flugone DM (Chlorpheniramine, Dextro, Paraceta...	Koff-Go (Chlorpheniramine, Dextro & Phenylephr...	NaN	Hyryllin M can also work in productive cough	Flugone can be used from 1 year	Hyryllin M from two year	Koff-Go recommended from 2 years and above	
4	Dry Cough	2-5	years	Piritex Junior (Dextro, Pseudoephedrine, Chlor...	Contus Peadiatric linctus (Phenylephrine, Chl...	NaN	NaN	Dry cough + Nasal Decongestion + Anti-Allergy	Piritex Junior from 1 year	Contus Paed from 2 year	Rinalin recommended from 2 years	

Figure 2

3. **Question**: How much data do I have?
Answer: There are 359 records in my dataset as shown. Looking at the last rows, you find that most of the alternative fields have no data yet they are required in my training, but this is okay because it is not a must for all products to have alternative products.

```
[22]: data.tail() # Checking for the number of records I have in my dataset.
```

[22]:

	Diagnosis	Age Range	age unit	PRODUCT DESCRIPTION/ BRAND	ALTERNATIVE PRODUCT 1	ALTERNATIVE PRODUCT 2	APPROPRIATE ADD ON	Comments	Age Range 1	Age Range 2	Age Range.1	Contraindications	Unnamed: 12
355	Vaccination	0-5	years	MEASLES RUBELLA MR - UNEPI	MEASLES RUBELLA	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
356	Vaccination	0-5	years	ORAL POLIO - UNEPI	ORAL POLIO	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
357	Vaccination	All	years	YELLOW FEVER - UNEPI	STAMARIL (Multiple dose)	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
358	Vaccination	0-5	years	VITAMIN A UNEPI	RETINOLE	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
359	Vaccination	0-5	years	IPV - UNEPI	UROPOLIO	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Figure 3

In my data, I have 360 rows and 14 columns, but remember I am only considering only 6 columns because they are the ones that fit my training.

```
data.shape
```

```
(360, 14)
```

Figure 4

Getting a high level overview of the data, I see that I have 14 unique diagnoses, Meaning the sample space on the diagnoses is 14, with high blood pressure appearing most, the sample space also includes 12 unique age ranges and 336 unique products based on these number of records, I think that this is good for a start.

```
[28]: data.describe()
```

[28]:

	Diagnosis	Age Range	age unit	PRODUCT DESCRIPTION/ BRAND	ALTERNATIVE PRODUCT 1	ALTERNATIVE PRODUCT 2	APPROPRIATE ADD ON	Comments	Age Range 1	Age Range 2	Age Range.1	Contraindications	Unnar
count	354	346	349	351	321	146	72	66	47	15	6	3	
unique	14	12	3	336	317	146	72	62	36	15	6	3	
top	High blood Pressure	>= 12	years	CARBAMAZEPINE TABLETS 200 MG	Contus Peadiatric linctus (Phenylephrine, Chl...	Zedex (Dextro, Bromhexin, Ammonium Chloride + ...	Ambroxol capsules	High risk of liver damage (Do CBC,LFTs & RFTs ...	From 2 years and above	Piritex baby from 3 months	Piritex Junior from 1 year	Risk of high blood pressure	CADI 20MC II (
freq	126	189	336	3	2	1	1	3	8	1	1	1	

Figure 5

The 14 unique diagnoses focused on in this dataset are:

```
[39]: data['Diagnosis'].unique()

[39]: array(['Dry Cough', nan, 'Wet Cough', 'Cough / expectorants', 'Flue',
       'Decongestants', 'Mixed Cough and flue', 'Supplements',
       'Sickle Cell Aneamia', 'High blood Pressure', 'Diabetes Mellitus',
       'Blood Disorders', 'Psychosis', 'Obesity', 'Vaccination'],
      dtype=object)
```

Conclusion: According to this phase of understanding data, you find that data is not clean. The example is in the diagnoses listed above. One of them is nan, meaning that data needs cleaning.

Data Cleaning

4. **Question**: Is the data clean?
   Answer: No.

   First reason as to why our data is not clean is because it has none required fields as demonstrated in the first phase of understanding data.
   Therefore, we need to get rid of them as shown below. In data wrangling, I have been able to get rid of the none required fields as shown below, remaining with only the 6 required fields.

```
[ ]: # Phase 3 Cleaning the data

[ ]: # Dropping none required Fields

[45]: data_with_required_fields = data.drop(['APPROPRIATE ADD ON', 'Comments', 'Age Range 1', 'Age Range 1', 'Age Range 2', 'Contraindication

[47]: # Confirming if none required fields have been remove.
      data_with_required_fields.columns

[47]: Index(['Diagnosis', 'Age Range', 'age unit', 'PRODUCT DESCRIPTION/ BRAND',
             'ALTERNATIVE PRODUCT 1', 'ALTERNATIVE PRODUCT 2'],
            dtype='object')
```

Second reason: We have missing values that I need to get rid of, like diagnosis has 6, age range has 14 and many more as shown below. The reason as to why I need to get rid of them is because I do not need them.

```
[55]: # Looking for missing values.
      data_with_required_fields.isnull().sum()

[55]: Diagnosis                     6
      Age Range                    14
      age unit                     11
      PRODUCT DESCRIPTION/ BRAND    9
      ALTERNATIVE PRODUCT 1        39
      ALTERNATIVE PRODUCT 2       214
      dtype: int64
```

The figure below shows how I got rid of missing values.

5. **Question**: Has the Data been Cleaned?
   Answer: Yes.
   This is because missing values have been removed,no duplicates, no null records and also we only have our required fields as shown below

```
•[60]: # Getting rid of records with missing values.
       data_with_required_fields_and_no_missing_values = data_with_required_fields.dropna(subset=['Diagnosis',  'Age Range', 'age unit', 'PROD

•[62]: # Confirming if missing values have been remove.
       data_with_required_fields_and_no_missing_values.isnull().sum()

[62]:  Diagnosis                     0
       Age Range                     0
       age unit                      0
       PRODUCT DESCRIPTION/ BRAND    0
       ALTERNATIVE PRODUCT 1         0
       ALTERNATIVE PRODUCT 2         0
       dtype: int64

[66]:  # Check for duplicates
       duplicates = data_with_required_fields_and_no_missing_values.duplicated().sum()

[67]:  duplicates

[67]:  np.int64(0)
```

Relationships between the variables

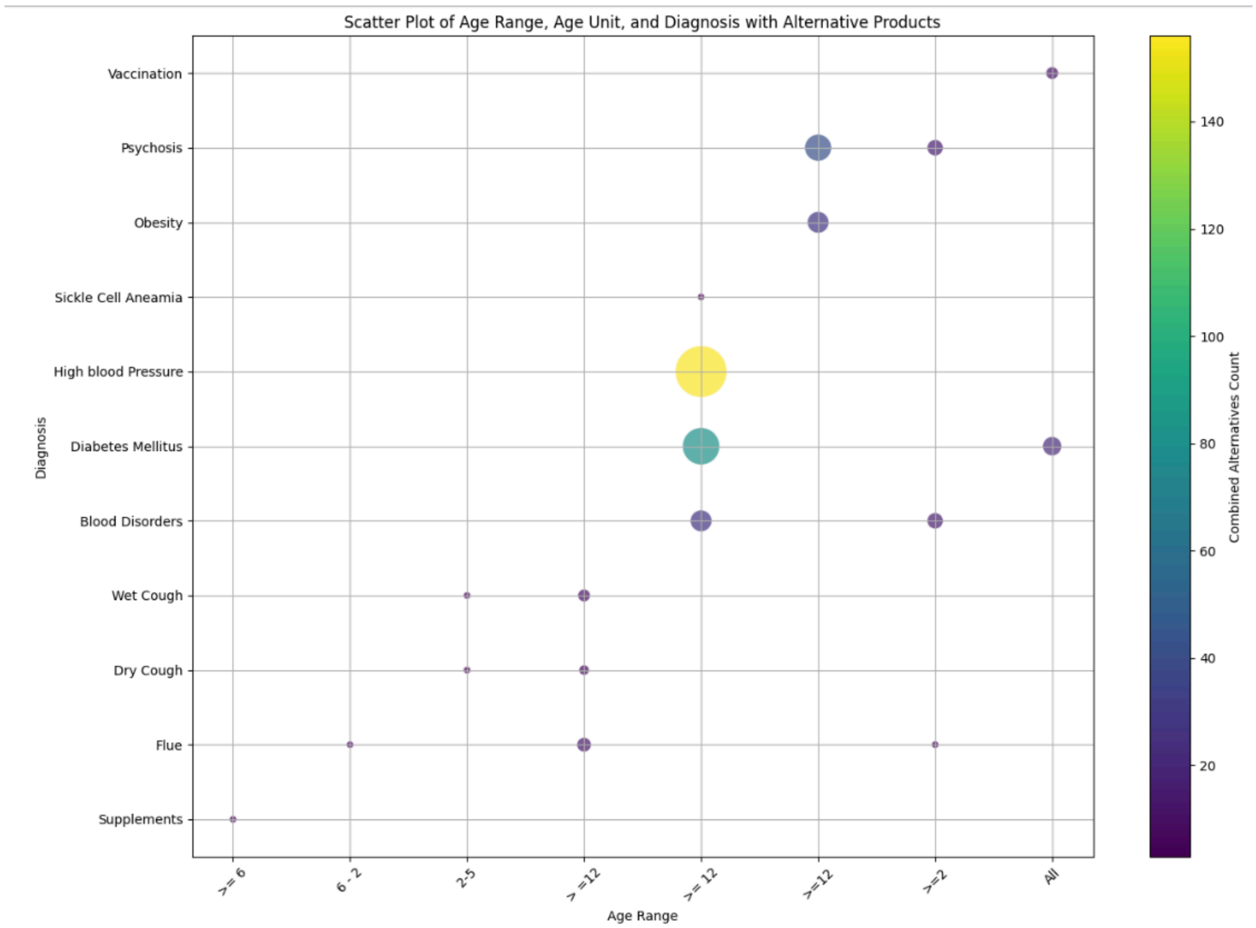
6. What are some of the insights can I draw from this data?
   I have come up with a pivot table to help me summarize products based on their diagnosis, age range and age unit, so as to find the patterns.

```
[72]: # Finding Relationships between the variables or Finding patterns
```

```
[ ]: # Check how products and their alternatives are distributed among diffent diagnosis.
```

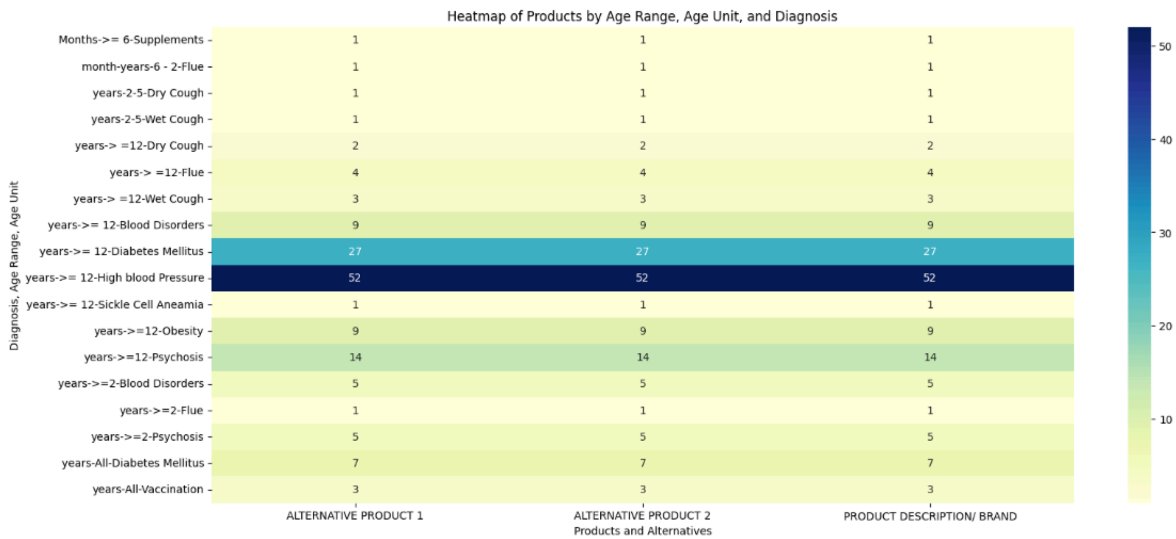
```
[42]: pivot_table = data_with_required_fields_and_no_missing_values.pivot_table(
          index=['age unit', 'Age Range', 'Diagnosis'],
          values=['ALTERNATIVE PRODUCT 1', 'ALTERNATIVE PRODUCT 2', 'PRODUCT DESCRIPTION/ BRAND'],
          aggfunc='count',
          fill_value=0
      )
```

Finding the distribution of products among different diagnoses and age ranges.



Finding the distribution of products and their alternatives among diagnosis, age range and age unit using a heatmap.

```
[50]: # Plotting heatmap to find the number of products and alternatives per diagnosis.
plt.figure(figsize=(18, 8))
sns.heatmap(pivot_table, annot=True, cmap='YlGnBu')
plt.title('Heatmap of Products by Age Range, Age Unit, and Diagnosis')
plt.xlabel('Products and Alternatives')
plt.ylabel('Diagnosis, Age Range, Age Unit')
plt.show()
```



7. What patterns am I seeing?
You find that the High blood pressure diagnosis has the most products and alternatives for people with age range 12 and above.

From the scatter plot, High blood pressure patients could be having enough medication in stock, but you find that we may need to stock more supplements, cough and flue medication.

Conclusions

The above grouping will help me determine the most appropriate products for prescription hence avoiding leaving out products unknown to doctors while prescribing therefore increasing sales, and reducing products expiries.

It is going to help the company know what to stock, based on the count of the items available.