

Pandas provide extremely streamlined forms of data representation. It is built on top of Numpy.

In [1]:

```
import pandas as pd
```

In [2]:

```
dataset=pd.read_csv("store_data.csv",header=None)
dataset.head()
```

Out[2]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole wheat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	green tea	honey	salad	mineral water	salmon	antioxydant juice
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [3]:

```
#converting the dataset into a list of lists
ds= dataset.sample(n=200)
ds=ds.values.tolist()
```

In [4]:

```
lists=[ [item for item in record if isinstance(item,str)] for record in ds]
lists[:5]
```

Out[4]:

```
[['vegetables mix', 'asparagus'],
 ['milk'],
 ['frozen vegetables'],
 ['cake', 'green tea', 'strawberries'],
 ['chocolate']]
```

Create an empty dictionary. Traverse through each item in itemsets and create a new key with it's value as 0 if there is no key for the item in the dictionary. If the key is already existed in the dictionary, increment it's value.

In [5]:

```
d={}
for record in lists:
    for item in record:
        d[item]=d.get(item,0)+1
print(d)
items =list(d.keys())
```

```
{'vegetables mix': 10, 'asparagus': 1, 'milk': 27, 'frozen vegetables': 22, 'cake': 19, 'green tea': 33, 'strawberries': 4, 'chocolate': 42, 'fresh tuna': 4, 'mineral water': 46, 'french wine': 5, 'cereals': 5, 'barbecue sauce': 3, 'eggplant': 3, 'escalope': 16, 'shrimp': 16, 'tomatoes': 13, 'honey': 9, 'spaghetti': 37, 'turkey': 19, 'olive oil': 16, 'meatballs': 7, 'energy bar': 6, 'french fries': 43, 'melons': 4, 'low fat yogurt': 21, 'grated cheese': 8, 'black tea': 7, 'chicken': 10, 'frozen smoothie': 11, 'herb & pepper': 11, 'ground beef': 16, 'pepper': 5, 'tomato sauce': 6, 'chutney': 1, 'eggs': 26, 'oil': 9, 'oatmeal': 4, 'brownies': 11, 'whole wheat rice': 10, 'avocado': 8, 'cottage cheese': 1, 'tomato juice': 5, 'flax seed': 2, 'fromage blanc': 5, 'salmon': 16, 'protein bar': 3, 'clothes accessories': 4, 'cookies': 15, 'burgers': 17, 'shallot': 2, 'pancakes': 17, 'toothpaste': 2, 'almonds': 7, 'cooking oil': 8, 'champagne': 12, 'energy drink': 7, 'yams': 1, 'whole wheat pasta': 9, 'soup': 11, 'asparagus': 1, 'babies food': 1, 'red wine': 7, 'body spray': 4, 'parmesan cheese': 5, 'gums': 4, 'mayonnaise': 2, 'tea': 1, 'zucchini': 2, 'bramble': 1, 'fresh bread': 5, 'hot dogs': 2, 'butter': 3, 'nonfat milk': 1, 'magazines': 1, 'candy bars': 4, 'muffins': 6, 'bug spray': 1, 'pasta': 4, 'soda': 1, 'white wine': 4, 'shampoo': 2, 'light mayo': 3, 'green beans': 1, 'napkins': 1, 'ham': 3, 'whole wheat flour': 3, 'corn': 1, 'chocolate bread': 1, 'yogurt cake': 3, 'cauliflower': 2, 'spinach': 2, 'mint': 2, 'mint green tea': 1, 'extra dark chocolate': 1, 'chili': 2, 'pet food': 1, 'gluten free bar': 1, 'blueberries': 1, 'dessert wine': 2, 'antioxydant juice': 2, 'rice': 1}
```

In [6]:

```
#SUPPORT says how popular an itemset is.
#CONFIDENCE says how likely an item Y is purchased if X is purchased.
MIN_SUP=5
MIN_CONF=25
```

In [7]:

```
for item in items:
    if(d[item]<MIN_SUP):
        del d[item]
print('The set of single items having min support:\n\n',d)
items=list(d.keys())
```

The set of single items having min support:

```
{'vegetables mix': 10, 'milk': 27, 'frozen vegetables': 22, 'cake': 19, 'green tea': 33, 'chocolate': 42, 'mineral water': 46, 'french wine': 5, 'cereals': 5, 'escalope': 16, 'shrimp': 16, 'tomatoes': 13, 'honey': 9, 'spaghetti': 37, 'turkey': 19, 'olive oil': 16, 'meatballs': 7, 'energy bar': 6, 'french fries': 43, 'low fat yogurt': 21, 'grated cheese': 8, 'black tea': 7, 'chicken': 10, 'frozen smoothie': 11, 'herb & pepper': 11, 'ground beef': 16, 'pepper': 5, 'tomato sauce': 6, 'eggs': 26, 'oil': 9, 'brownies': 11, 'whole wheat rice': 10, 'avocado': 8, 'tomato juice': 5, 'fromage blanc': 5, 'salmon': 16, 'cookies': 15, 'burgers': 17, 'pancakes': 17, 'almonds': 7, 'cooking oil': 8, 'champagne': 12, 'energy drink': 7, 'whole wheat pasta': 9, 'soup': 11, 'red wine': 7, 'parmesan cheese': 5, 'fresh bread': 5, 'muffins': 6}
```

In [8]:

```
from itertools import combinations
```

In [9]:

```
s=[]
s.append(d)
for k in range(2,len(items)):
    d1=dict()
    #iterating through each combination of k items
    for c in combinations(items,k):
        for record in lists:
            if set(c).issubset(set(record)):
                d1[c]=d1.get(c,0)+1
            if d1.get(c) and d1.get(c)<MIN_SUP:
                del(d1[c])
        if not d1:
            break
    s.append(d1)
print(s)
#basically we are storing the item sets of 1,2 3.. elements in s that show min support
```

```
[{'vegetables mix': 10, 'milk': 27, 'frozen vegetables': 22, 'cake': 19, 'green tea': 33, 'chocolate': 42, 'mineral water': 46, 'french wine': 5, 'cereals': 5, 'escalope': 16, 'shrimp': 16, 'tomatoes': 13, 'honey': 9, 'spaghetti': 37, 'turkey': 19, 'olive oil': 16, 'meatballs': 7, 'energy bar': 6, 'french fries': 43, 'low fat yogurt': 21, 'grated cheese': 8, 'black tea': 7, 'chicken': 10, 'frozen smoothie': 11, 'herb & pepper': 11, 'ground beef': 16, 'pepper': 5, 'tomato sauce': 6, 'eggs': 26, 'oil': 9, 'brownies': 11, 'whole wheat rice': 10, 'avocado': 8, 'tomato juice': 5, 'fromage blanc': 5, 'salmon': 16, 'cookies': 15, 'burgers': 17, 'pancakes': 17, 'almonds': 7, 'cooking oil': 8, 'champagne': 12, 'energy drink': 7, 'whole wheat pasta': 9, 'soup': 11, 'red wine': 7, 'parmesan cheese': 5, 'fresh bread': 5, 'muffins': 6}, {'milk', 'chocolate': 6, ('milk', 'mineral water'): 5, ('milk', 'spaghetti'): 15, ('milk', 'turkey'): 5, ('milk', 'olive oil'): 7, ('milk', 'low fat yogurt'): 6, ('milk', 'eggs'): 6, ('milk', 'salmon'): 5, ('frozen vegetables', 'chocolate'): 6, ('frozen vegetables', 'mineral water'): 6, ('frozen vegetables', 'tomatoes'): 5, ('cake', 'green tea'): 6, ('cake', 'chocolate'): 6, ('cake', 'mineral water'): 6, ('cake', 'spaghetti'): 7, ('cake', 'french fries'): 7, ('cake', 'eggs'): 5, ('cake', 'burgers'): 6, ('green tea', 'chocolate'): 5, ('green tea', 'mineral water'): 6, ('green tea', 'spaghetti'): 6, ('green tea', 'turkey'): 7, ('green tea', 'french fries'): 9, ('green tea', 'ground beef'): 5, ('green tea', 'burgers'): 5, ('green tea', 'pancakes'): 5, ('chocolate', 'mineral water'): 14, ('chocolate', 'shrimp'): 6, ('chocolate', 'spaghetti'): 12, ('chocolate', 'turkey'): 6, ('chocolate', 'olive oil'): 6, ('chocolate', 'french fries'): 15, ('chocolate', 'low fat yogurt'): 6, ('chocolate', 'black tea'): 5, ('chocolate', 'ground beef'): 7, ('chocolate', 'eggs'): 7, ('chocolate', 'salmon'): 7, ('chocolate', 'burgers'): 5, ('chocolate', 'pancakes'): 6, ('chocolate', 'soup'): 5, ('mineral water', 'escalope'): 5, ('mineral water', 'shrimp'): 7, ('mineral water', 'spaghetti'): 12, ('mineral water', 'turkey'): 6, ('mineral water', 'french fries'): 11, ('mineral water', 'low fat yogurt'): 6, ('mineral water', 'eggs'): 8, ('mineral water', 'brownies'): 6, ('mineral water', 'salmon'): 7, ('mineral water', 'spaghetti'): 5, ('shrimp', 'spaghetti'): 5, ('shrimp', 'french fries'): 5, ('shrimp', 'burgers'): 5, ('spaghetti', 'turkey'): 8, ('spaghetti', 'olive oil'): 7, ('spaghetti', 'french fries'): 8, ('spaghetti', 'low fat yogurt'): 10, ('spaghetti', 'ground beef'): 5, ('spaghetti', 'eggs'): 8, ('spaghetti', 'brownies'): 5, ('spaghetti', 'salmon'): 7, ('spaghetti', 'burgers'): 7, ('spaghetti', 'pancakes'): 6, ('spaghetti', 'almonds'): 5, ('spaghetti', 'soup'): 6, ('turkey', 'eggs'): 5, ('turkey', 'burgers'): 5, ('olive oil', 'low fat yogurt'): 7, ('olive oil', 'salmon'): 6, ('french fries', 'low fat yogurt'): 6, ('french fries', 'eggs'): 6, ('french fries', 'burgers'): 6, ('french fries', 'pancakes'): 7, ('french fries', 'champagne'): 6, ('low fat yogurt', 'eggs'): 7, ('eggs', 'pancakes'): 5, ('eggs', 'soup'): 5}, {'milk', 'chocolate', 'spaghetti': 5, ('milk', 'spaghetti', 'olive oil'): 5, ('cake', 'spaghetti', 'french fries'): 5, ('chocolate', 'spaghetti', 'french fries'): 5, ('chocolate', 'spaghetti', 'low fat yogurt'): 6, ('chocolate', 'spaghetti', 'eggs'): 5, ('chocolate', 'french fries', 'pancakes'): 5, ('spaghetti', 'turkey', 'eggs'): 5, ('spaghetti', 'olive oil', 'low fat yogurt'): 5}]
```

In [10]:

```

rules = dict()
for combo in s[-1]:
    for item in combo:
        c = list(combo)
        c.remove(item)
        l_c = len(c)
        c = c[0] if l_c == 1 else tuple(c)
        rule_1 = s[-1][combo]/s[0][item]*100
        rule_2 = s[-1][combo]/s[l_c-1][c]*100
        if rule_1>=MIN_CONF:
            rules[f"{item}->{c}"] = rule_1
        if rule_2>=MIN_CONF:
            rules[f"{c}->{item}"] = rule_2
print(rules)

```

```

{'('chocolate', 'spaghetti')->milk": 41.66666666666667, "('milk', 'spaghetti')->chocolate": 33.33333333333333, "('milk', 'chocolate')->spaghetti": 83.33333333333334, "('spaghetti', 'olive oil')->milk": 71.42857142857143, "('milk', 'olive oil')->spaghetti": 71.42857142857143, "olive oil->('milk', 'spaghetti')": 31.25, "('milk', 'spaghetti')->olive oil": 33.33333333333333, "cake->('spaghetti', 'french fries')": 26.31578947368421, "('spaghetti', 'french fries')->cake": 62.5, "('cake', 'french fries')->spaghetti": 71.42857142857143, "('cake', 'spaghetti')->french fries": 71.42857142857143, "('spaghetti', 'french fries')->chocolate": 62.5, "('chocolate', 'french fries')->spaghetti": 33.33333333333333, "('chocolate', 'spaghetti')->french fries": 41.66666666666667, "('spaghetti', 'low fat yogurt')->chocolate": 60.0, "('chocolate', 'low fat yogurt')->spaghetti": 100.0, "low fat yogurt->('chocolate', 'spaghetti')": 28.57142857142857, "('chocolate', 'spaghetti')->low fat yogurt": 50.0, "('spaghetti', 'eggs')->chocolate": 62.5, "('chocolate', 'eggs')->spaghetti": 71.42857142857143, "('chocolate', 'spaghetti')->eggs": 41.66666666666667, "('french fries', 'pancakes')->chocolate": 71.42857142857143, "('chocolate', 'pancakes')->french fries": 83.33333333333334, "pancakes->('chocolate', 'french fries')": 29.411764705882355, "('chocolate', 'french fries')->pancakes": 33.33333333333333, "('turkey', 'eggs')->spaghetti": 100.0, "turkey->('spaghetti', 'eggs')": 26.31578947368421, "('spaghetti', 'eggs')->turkey": 62.5, "('spaghetti', 'turkey')->eggs": 62.5, "('olive oil', 'low fat yogurt')->spaghetti": 71.42857142857143, "olive oil->('spaghetti', 'low fat yogurt')": 31.25, "('spaghetti', 'low fat yogurt')->olive oil": 50.0, "('spaghetti', 'olive oil')->low fat yogurt": 71.42857142857143}

```

In [11]:

```

import matplotlib.pyplot as plt
import numpy as np

```

In [12]:

```

rules=sorted(rules.items(),key=lambda x:x[1],reverse=True)

```

In [13]:

```
x=list(map(lambda x:x[0],rules))
y=list(map(lambda x:x[1],rules))
fig=plt.figure(figsize=(20,15))
plt.bar(x[:5],y[:5],width=0.2)
plt.ylabel("confidence")
plt.show()
```

