Q1)

```
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori

df = pd.read_csv('GroceryStoreDataSet.csv')
print (df)

transactions = df[0].apply(lambda x: [item.strip() for item in x.split(',')])

te = TransactionEncoder()
te_ary = te.fit(transactions).transform(transactions)
df_encoded = pd.DataFrame(te_ary, columns=te.columns_)

frequent_itemsets = apriori(df_encoded, min_support=0.25,
use_colnames=True)

print("Frequent Itemsets:")
print(frequent_itemsets)


frequent_itemsets['length'] = frequent_itemsets['itemsets'].apply(lambda x:
len(x))
itemsets_length_1 = frequent_itemsets[frequent_itemsets['length'] == 1]

print(itemsets_length_1)
```

```
if itemsets_length_1.empty:
print("No itemsets of length 1 found. Try adjusting the min_support value.")
else:

most_frequent_item =
itemsets_length_1.loc[itemsets_length_1['support'].idxmax()]



item_name = list(most_frequent_item['itemsets'])[0]
item_support = most_frequent_item['support']

print(f"\nThe most frequently bought item is '{item_name}' with a support of
{item_support:.2f}")
```

Q2)

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import pandas as pd
```

```
data = pd.read_csv("accidentcleaned.csv");
data

features = ['Age', 'Speed']
target='Survival'

X= data[features]
y= data[target]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

model = LogisticRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test,y_pred)
print("Accuracy:", accuracy )

Survive = pd.DataFrame({
'Age': [39],
'Speed': [60]
})
predicted= model.predict(Survive)
print("Predicted Survival:", predicted[0])
```