Q.2

```
import pandas as pd
from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Dataset
data = {
    'Annual_Income': [50, 30, 60, 25, 55, 35, 65, 28, 70],
    'Credit_Score': [700, 650, 720, 600, 710, 640, 730, 610, 750],
    'Loan_Default': [0, 1, 0, 1, 0, 1, 0, 1, 0]
}

# Create DataFrame
df = pd.DataFrame(data)

# Features and Labels
X = df[['Annual_Income', 'Credit_Score']]
y = df['Loan_Default']

# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardizing the data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# SVM Classifier
classifier = svm.SVC(kernel='linear')
classifier.fit(X_train, y_train)

# Predictions
y_pred = classifier.predict(X_test)

# Model Evaluation
print(f"Accuracy: {accuracy_score(y_test, y_pred) * 100:.2f}%")
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))

# Predict for a new customer
new_data = pd.DataFrame({'Annual_Income': [75], 'Credit_Score': [760]})
```

```
new_data = scaler.transform(new_data)
prediction = classifier.predict(new_data)

if prediction[0] == 1:
    print("The customer is likely to default on the loan.")
else:
    print("The customer is not likely to default on the loan.")
```

Q.3

```
import pandas as pd
from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Dataset
data = {
    'Age': [25, 45, 35, 50, 40, 55, 30, 60, 38],
    'BMI': [22, 28, 24, 30, 26, 32, 23, 34, 25],
    'Insurance_Claim': [0, 1, 0, 1, 0, 1, 0, 1, 0]
}

# Create DataFrame
df = pd.DataFrame(data)

# Features and Labels
X = df[['Age', 'BMI']]
y = df['Insurance_Claim']

# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardizing the data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# SVM Classifier
classifier = svm.SVC(kernel='linear')
classifier.fit(X_train, y_train)
```

```
# Predictions
y_pred = classifier.predict(X_test)

# Model Evaluation
print(f"Accuracy: {accuracy_score(y_test, y_pred) * 100:.2f}%")
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))

# Predict for a new person
new_data = pd.DataFrame({'Age': [48], 'BMI': [29]})
new_data = scaler.transform(new_data)
prediction = classifier.predict(new_data)

if prediction[0] == 1:
    print("The person is likely to claim insurance.")
else:
    print("The person is not likely to claim insurance.")
```