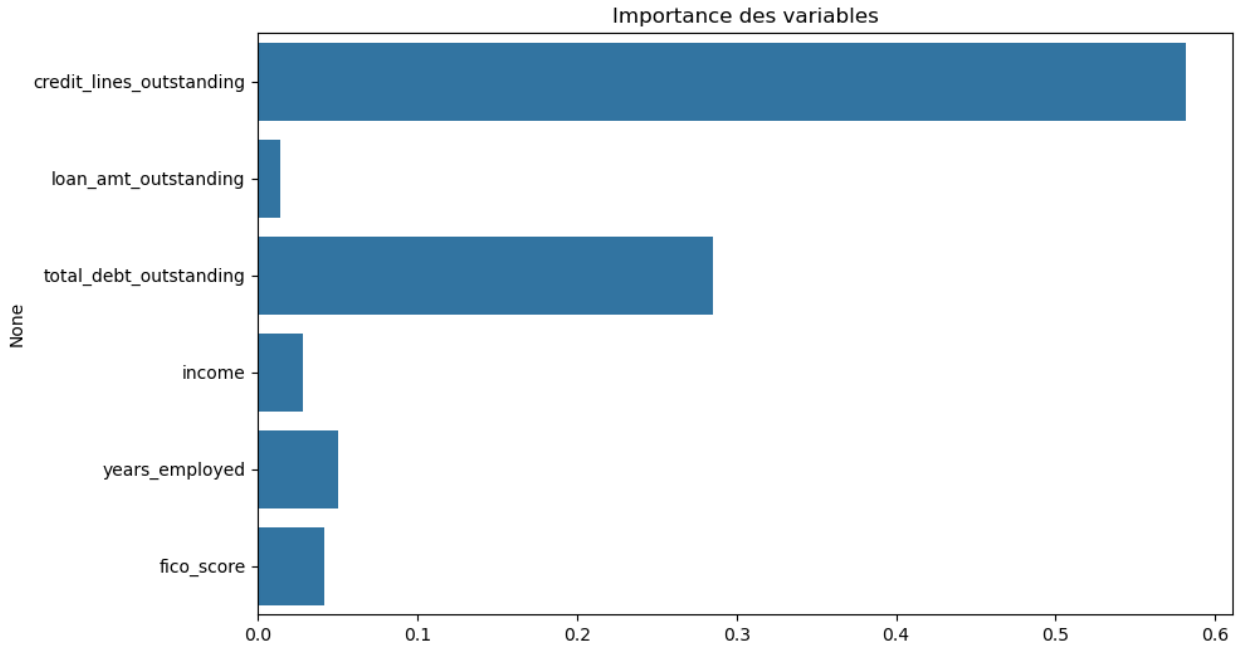
```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# 1. Chargement du bon fichier CSV depuis la base de données de
Jupyter
df = pd.read_csv("Task 3 and 4_Loan_Data.csv")

# 2. Prétraitement
if 'customer_id' in df.columns:
    df = df.drop(columns=['customer_id'])

df = df.dropna()

X = df.drop(columns=['default'])
y = df['default']

# 3. Split & Standardisation
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# 4. Modélisation avec RandomForest
model = RandomForestClassifier(random_state=42)
model.fit(X_train_scaled, y_train)

# 5. Évaluation
y_pred = model.predict(X_test_scaled)
print("Classification report :\n", classification_report(y_test,
y_pred))
print("Confusion matrix :\n", confusion_matrix(y_test, y_pred))

# 6. Fonction de calcul de perte attendue
def expected_loss(loan_amt_outstanding, credit_lines_outstanding,
total_debt_outstanding, income, years_employed, fico_score):
    input_data = pd.DataFrame([[
        credit_lines_outstanding,
        loan_amt_outstanding,
        total_debt_outstanding,
        income,
        years_employed,
        fico_score
    ]], columns=X.columns)
```

```
    input_scaled = scaler.transform(input_data)
    pd_default = model.predict_proba(input_scaled)[0][1]
    loss_given_default = 0.90
    return pd_default * loss_given_default * loan_amt_outstanding

# 7. Exemple
example_loss = expected_loss(
    loan_amt_outstanding=10000,
    credit_lines_outstanding=3,
    total_debt_outstanding=25000,
    income=60000,
    years_employed=5,
    fico_score=650
)
print(f"Exemple de perte attendue : ${example_loss:.2f}")

# 8. Importance des variables
importances = model.feature_importances_
plt.figure(figsize=(10,6))
sns.barplot(x=importances, y=X.columns)
plt.title("Importance des variables")
plt.show()
```

```
Classification report :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1652
           1       0.99      0.98      0.98       348

    accuracy                           0.99      2000
   macro avg       0.99      0.99      0.99      2000
weighted avg       0.99      0.99      0.99      2000

Confusion matrix :
 [[1649    3]
 [   8  340]]
Exemple de perte attendue : $3330.00
```


Importance des variables
credit_lines_outstanding
loan_amt_outstanding
total_debt_outstanding
income
years_employed
fico_score
None
0.0
0.1
0.2
0.3
0.4
0.5
0.6