credit-risk-classification-model20-challenge-screenshots

Module 20 Challenge – credit-risk-classification

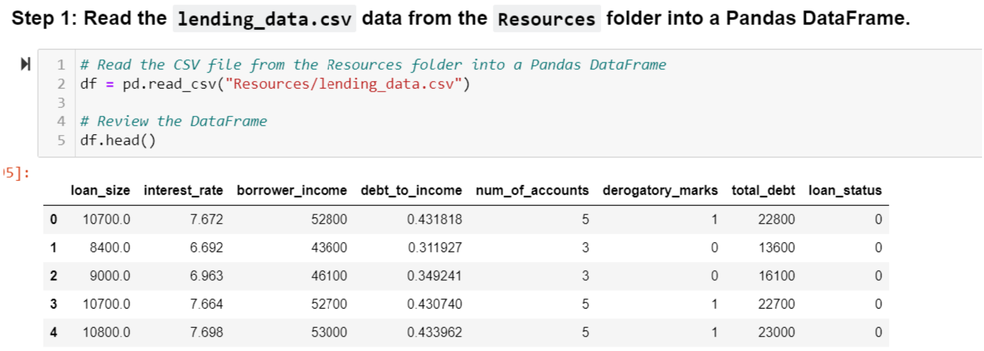
SMU DS – Raj Agrawal

Submitted on: 03-SEP-2023

Repository – https://github.com/RajAgrawal99/SMU_DS_Bootcamp_March2023_RA.git

Folder – credit-risk-classification-Challenge

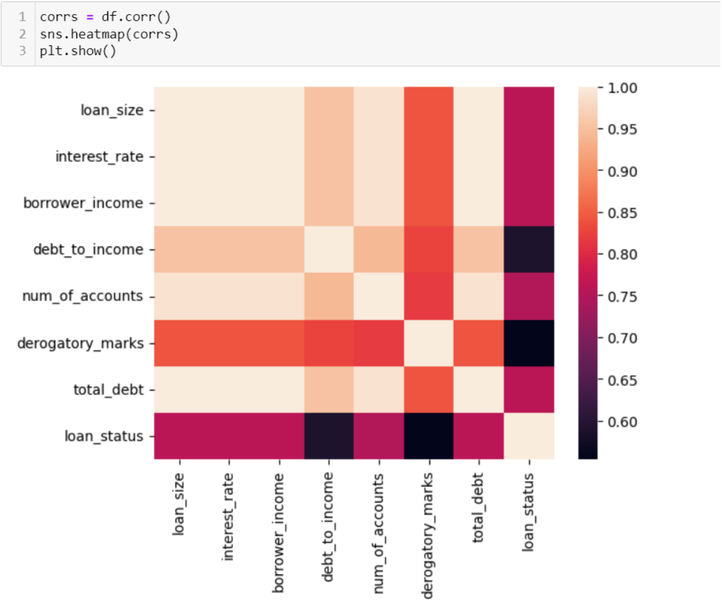
Data source - lending_data.csv



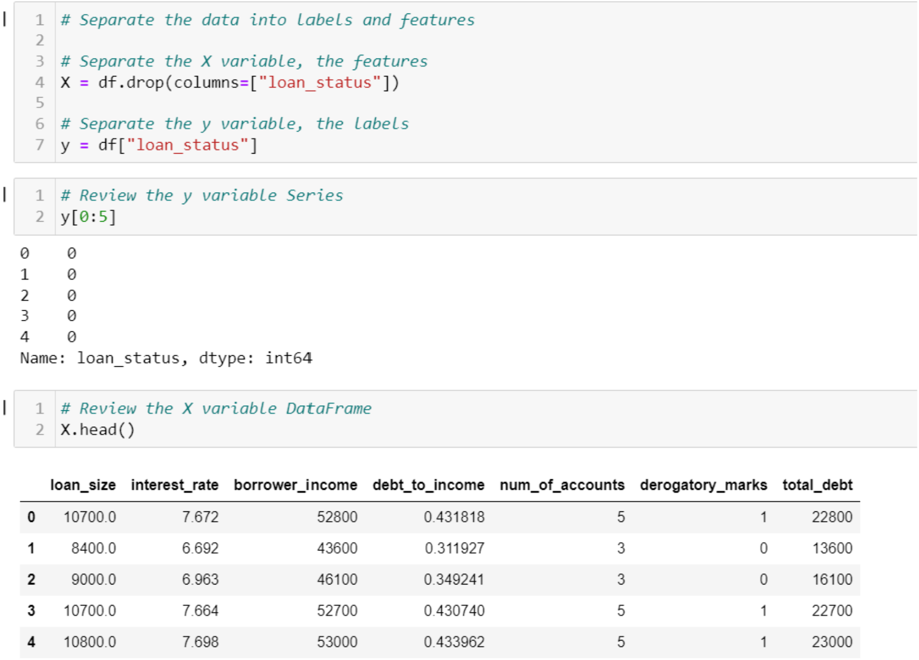
Step 1: Read the `lending_data.csv` data from the `Resources` folder into a Pandas DataFrame.

```python
# Read the CSV file from the Resources folder into a Pandas DataFrame
df = pd.read_csv("Resources/lending_data.csv")

# Review the DataFrame
df.head()
```

	loan_size	interest_rate	borrower_income	debt_to_income	num_of_accounts	derogatory_marks	total_debt	loan_status
0	10700.0	7.672	52800	0.431818	5	1	22800	0
1	8400.0	6.692	43600	0.311927	3	0	13600	0
2	9000.0	6.963	46100	0.349241	3	0	16100	0
3	10700.0	7.664	52700	0.430740	5	1	22700	0
4	10800.0	7.698	53000	0.433962	5	1	23000	0



```python
corrs = df.corr()
sns.heatmap(corrs)
plt.show()
```



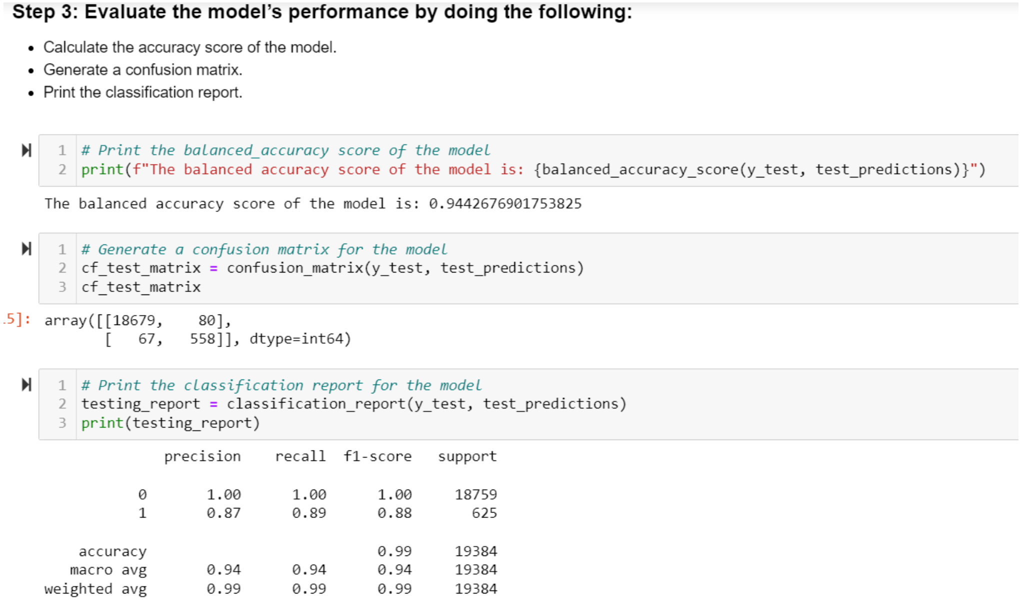
```python
# Separate the data into labels and features

# Separate the X variable, the features
X = df.drop(columns=["loan_status"])

# Separate the y variable, the labels
y = df["loan_status"]
```

```python
# Review the y variable Series
y[0:5]
```

```
0    0
1    0
2    0
3    0
4    0
Name: loan_status, dtype: int64
```

```python
# Review the X variable DataFrame
X.head()
```

	loan_size	interest_rate	borrower_income	debt_to_income	num_of_accounts	derogatory_marks	total_debt
0	10700.0	7.672	52800	0.431818	5	1	22800
1	8400.0	6.692	43600	0.311927	3	0	13600
2	9000.0	6.963	46100	0.349241	3	0	16100
3	10700.0	7.664	52700	0.430740	5	1	22700
4	10800.0	7.698	53000	0.433962	5	1	23000



Step 3: Evaluate the model’s performance by doing the following:

- Calculate the accuracy score of the model.
- Generate a confusion matrix.
- Print the classification report.

```python
# Print the balanced_accuracy score of the model
print(f"The balanced accuracy score of the model is: {balanced_accuracy_score(y_test, test_predictions)}")
```

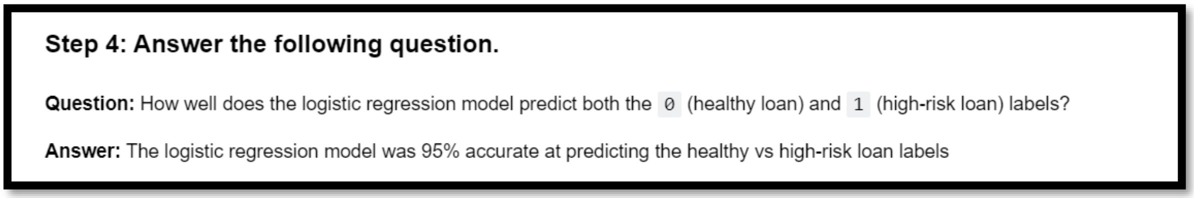
```
The balanced accuracy score of the model is: 0.9442676901753825
```

```python
# Generate a confusion matrix for the model
cf_test_matrix = confusion_matrix(y_test, test_predictions)
cf_test_matrix
```

```
array([[18679,    80],
       [   67,   558]], dtype=int64)
```

```python
# Print the classification report for the model
testing_report = classification_report(y_test, test_predictions)
print(testing_report)
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     18759
           1       0.87      0.89      0.88       625

    accuracy                           0.99     19384
   macro avg       0.94      0.94      0.94     19384
weighted avg       0.99      0.99      0.99     19384
```



Step 4: Answer the following question.

Question: How well does the logistic regression model predict both the `0` (healthy loan) and `1` (high-risk loan) labels?

Answer: The logistic regression model was 95% accurate at predicting the healthy vs high-risk loan labels

Step 2: Use the `LogisticRegression` classifier and the resampled data to fit the model and make predictions.

```python
# Instantiate the Logistic Regression model
# Assign a random_state parameter of 1 to the model
classifier = LogisticRegression(solver='lbfgs', random_state=1)

# Fit the model using the resampled training data
classifier.fit(X_ros_model, y_ros_model)
# Make a prediction using the testing data
predictions = classifier.predict(X_ros_model)
pd.DataFrame({'Predictions': predictions, 'Actual': y_ros_model})
```

28]:

	Predictions	Actual
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
...	...	...
150067	1	1
150068	1	1
150069	1	1
150070	1	1
150071	1	1

150072 rows × 2 columns

Step 3: Evaluate the model’s performance by doing the following:

- Calculate the accuracy score of the model.
- Generate a confusion matrix.
- Print the classification report.

```python
# Print the balanced_accuracy score of the model
print(f"The balanced accuracy score of the model is: {balanced_accuracy_score(y_ros_model, predictions)}")
```

```
The balanced accuracy score of the model is: 0.9945026387334079
```

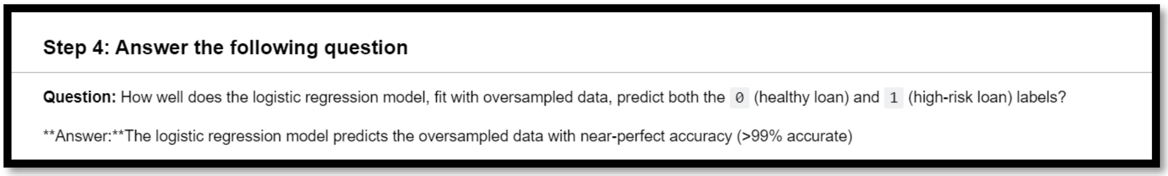
```python
# Generate a confusion matrix for the model
cf_matrix = confusion_matrix(y_ros_model, predictions)
cf_matrix
```

30]:

```
array([[74614,   422],
       [  403, 74633]], dtype=int64)
```

```python
# Print the classification report for the model
report = classification_report(y_ros_model, predictions)
print(report)
```

```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99     75036
           1       0.99      0.99      0.99     75036

    accuracy                           0.99    150072
   macro avg       0.99      0.99      0.99    150072
weighted avg       0.99      0.99      0.99    150072
```



Step 4: Answer the following question

Question: How well does the logistic regression model, fit with oversampled data, predict both the `0` (healthy loan) and `1` (high-risk loan) labels?

Answer: The logistic regression model predicts the oversampled data with near-perfect accuracy (>99% accurate)

Write a Credit Risk Analysis Report

1. **An overview of the analysis:** Explain the purpose of this analysis.

- The purpose of this analysis is to create and evaluate the accuracy of a data model that predicts the credity worthiness of potential borrowers from peer-to-peer lending services

2. **The results:** Using a bulleted list, describe the accuracy score, the precision score, and recall score of the machine learning model.

- Balanced Accuracy Score: 95.20% --> this means that when taking into account the sensitivity (recall and/or true positive rate) and specificity (true negative rate) of the model, the balanced prediction accuracy was 95.2%
- - Precision Score: 92% --> This means 92% of predicted positives were correct
- - Recall Score: 95% --> this means that the model was 95% precise in measuring true positive values our of all positive predictions made

3. **A summary:** Summarize the results from the machine learning model. Include your justification for recommending the model for use by the company. If you don't recommend the model, justify your reasoning.

- I would recommend using this model to predict the creditworthiness of borrowers, because it has over 95% accuracy in predicting the outcome of the repayment of the initial loan. That accuracy range could be easily molded into a business risk profile to ensure sufficient capital flow for the lenders to remain in business/make a profit.