

credit quality of a student based on the features in the dataset. Consider the number of inputs before determining the number of layers that your model will contain or the number of neurons on each layer. Then, compile and fit your model. Finally, evaluate the model to calculate its loss and accuracy.

To do so, complete the following steps:

1. Create a deep neural network by assigning the number of input features, the number of layers, and the number of neurons on each layer using Tensorflow's Keras.



HIDE HINT

You can start with a two-layer deep neural network model that uses the `relu` activation function for both layers.

2. Compile and fit the model using the `mse` loss function, the `adam` optimizer, and the `mse` evaluation metric.



HIDE HINT

When fitting the model, start with a small number of epochs, such as 50 or 100.

3. Evaluate the model using the test data to determine the model's loss and accuracy.
4. Save and export your model to an HDF5 file, and name the file `student_loans.h5`.

Predict Loan Repayment Success by Using your Neural Network Model

Use the model you saved in the previous section to make predictions on your reserved testing data.

To do so, complete the following steps:

1. Reload your saved model.
2. Make predictions on the testing data.
3. Create a DataFrame that includes both the predictions and the actual values.
4. Display a sample of the DataFrame you created in Step 3. Compare the prediction and actual values in this sample and describe what you notice.

Requirements

To receive all points, your Jupyter notebook file must have all of the following:

Prepare the Data for Use on a Neural Network Model (20 points)

- The `student_loans.csv` file was read into a Pandas DataFrame and a sample of the dataset was shown. (5 points)
- Two datasets were created: a target (y) dataset, which includes the "credit_ranking" column, and a features (x) dataset, which includes the other columns. (5 points)
- The features and target sets have been split into training and testing datasets. (5 points)
- Scikit-learn's `StandardScaler` was used to scale the features data. (5 points)

Compile and Evaluate a Model Using a Neural Network (40 points)

- A deep neural network was created with appropriate parameters. (10 points)
- The model was compiled and fit using the `mse` loss function, the `adam` optimizer, the `mse` evaluation metric, and a small number of epochs, such as 50 or 100. (10 points)
- The model was evaluated using the test data to determine its loss and accuracy. (10 points)
- The model was saved and exported to an HDF5 file named `student_loans.h5`. (10 points)

Predict Loan Repayment Success by Using your Neural Network Model (20 points)

- The saved model was reloaded. (5 points)
- The reloaded model was used to make predictions on the testing data. (5 points)
- There is a DataFrame that includes both the predictions and the actual values. (5 points)
- A sample of the DataFrame created in Step 3 is displayed, and there is a comparison of the values shown in the sample. (5 points)

Grading

This assignment will be evaluated against the requirements and assigned a grade according to the following table:

Grade	Points
A (+/-)	90+
B (+/-)	80–89
C (+/-)	70–79
D (+/-)	60–69
F (+/-)	< 60

Submission

Make sure to submit your work by the assignment due date. To do so, click Submit, then upload your project files. If you have any problems uploading your files, you may also provide a link to a folder within Google Drive, Dropbox, or a similar service. Get the details

problems uploading your files, you may also provide a link to folder within Google Drive, Dropbox, or a similar service. Set the sharing permissions so that anyone with the link can view your files.

Comments are disabled for graded submissions in Bootcamp Spot. If you have questions about your feedback, please notify your instructional staff or your Student Success Advisor.



[◀ Previous](#)[Next ▶](#)