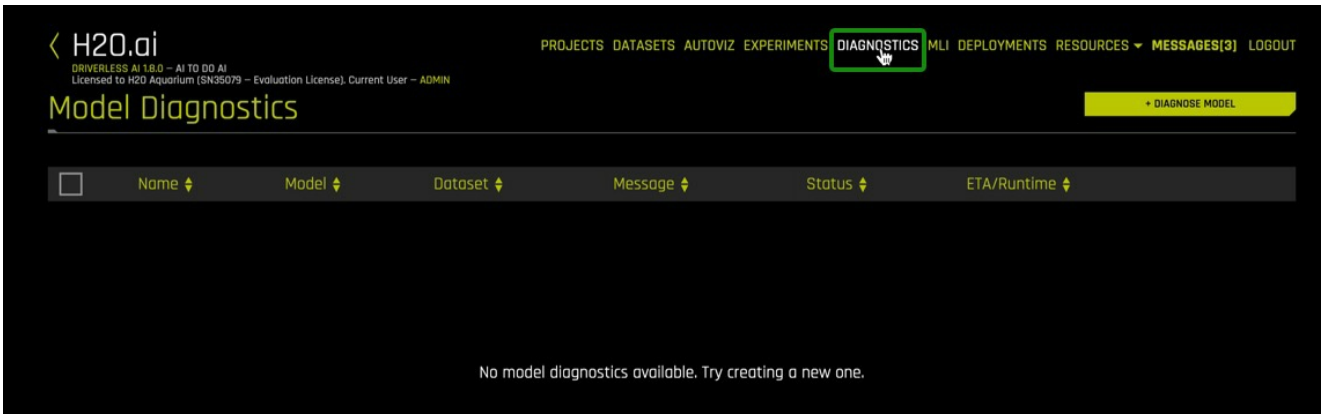
机器学习实验评分和分析教程-财务重点

h2oai.github.io/tutorials/machine-learning-experiment-scoring-and-analysis-tutorial-financial-focus

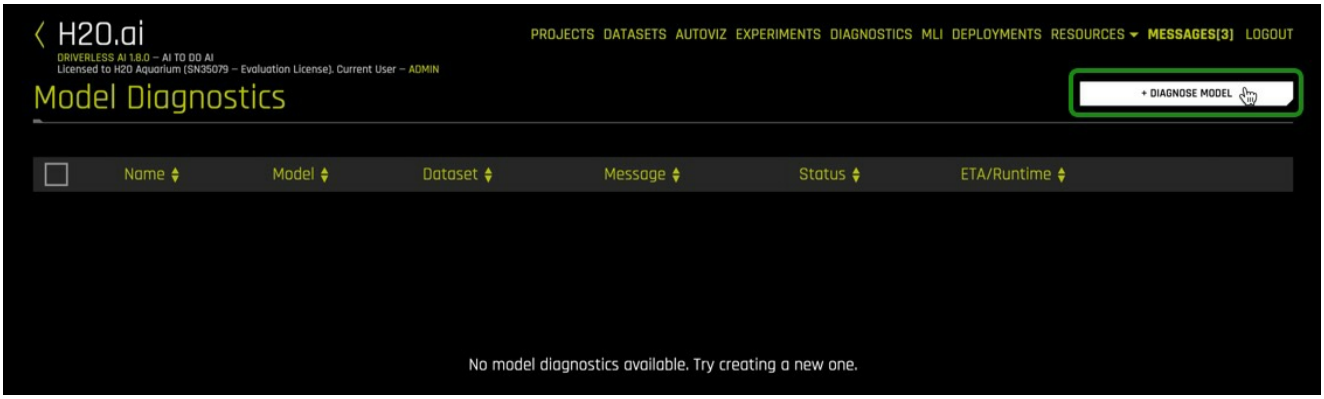
7.任务5：诊断评分和混淆矩阵

现在，我们将在freddie_mac_500_test集上运行模型诊断。诊断模型允许您通过Python API基于现有模型和数据集查看多个评分器的模型性能。

1.选择**诊断**

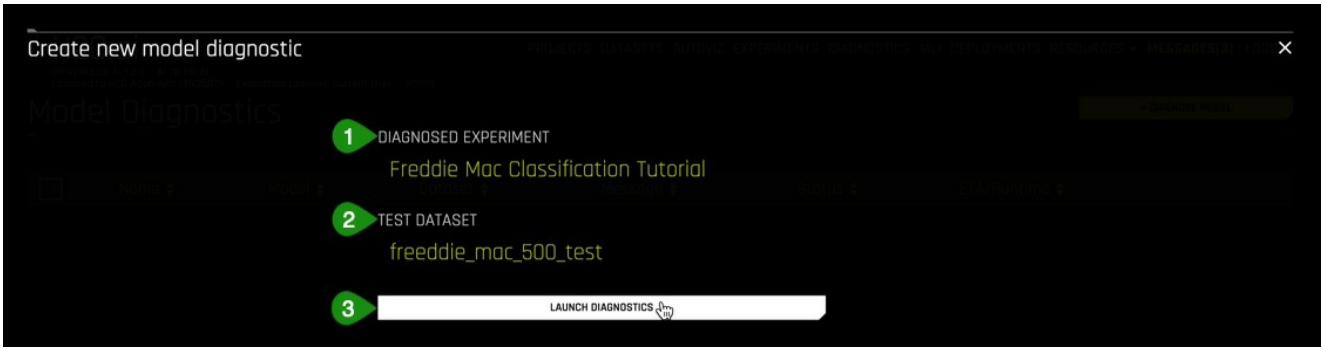


2.进入“ **诊断”**页面后，选择**+诊断模型**

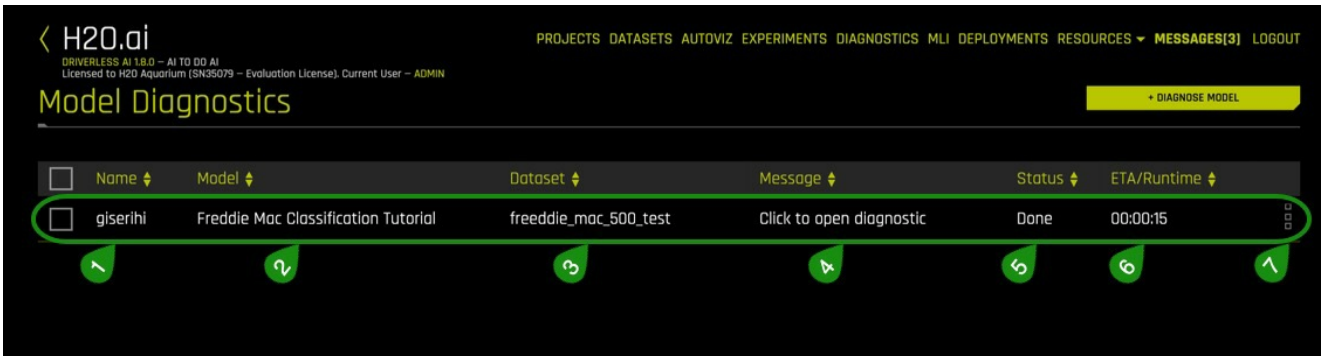


3.在**创建新模型诊断中**：

1. 单击“诊断实验”，然后选择您在任务4：**Freddie Mac分类教程中**完成的实验
2. 单击数据集，然后选择freddie_mac_500_test数据集
3. 通过单击**启动诊断程序**来启动诊断模型



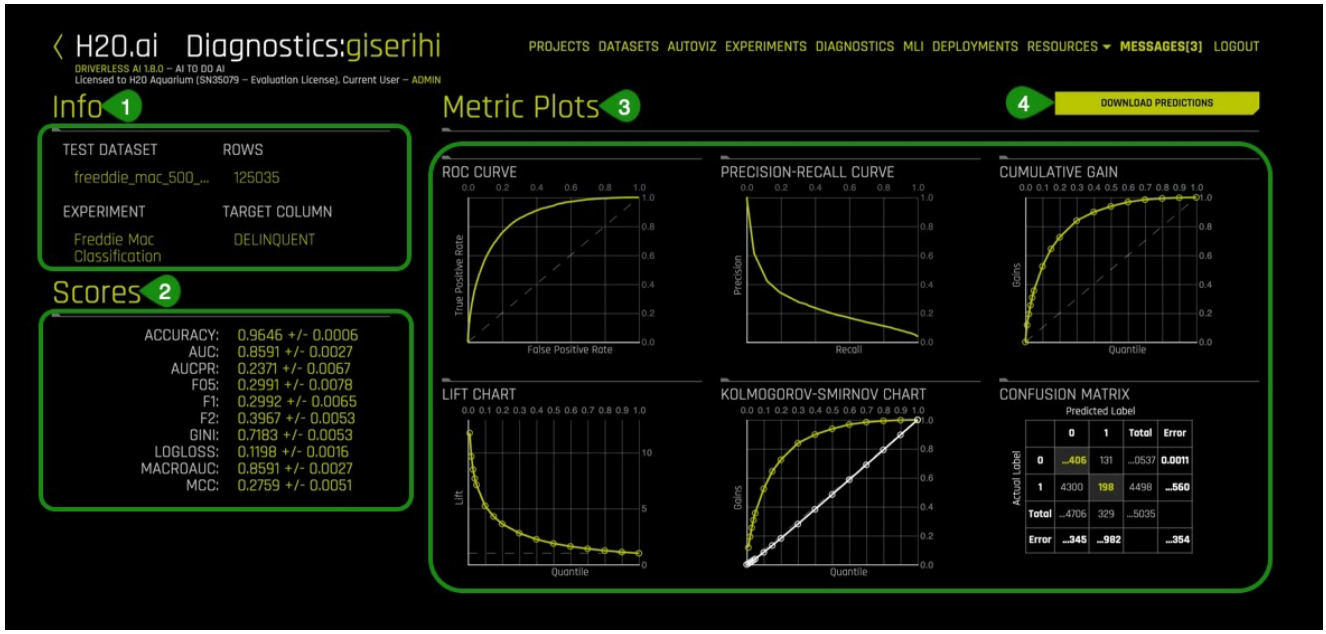
4.After the model diagnostics is done running, a model similar to the one below will appear:



Things to Note:

1. Name of new diagnostics model
2. **Model**: Name of ML model used for diagnostics
3. **Dataset**: name of the dataset used for diagnostic
4. **Message** : Message regarding new diagnostics model
5. **Status** : Status of new diagnostics model
6. **Time** : Time it took for the new diagnostics model to run
7. Options for this model

5. Click on the new diagnostics model and a page similar to the one below will appear:



Things to Note:

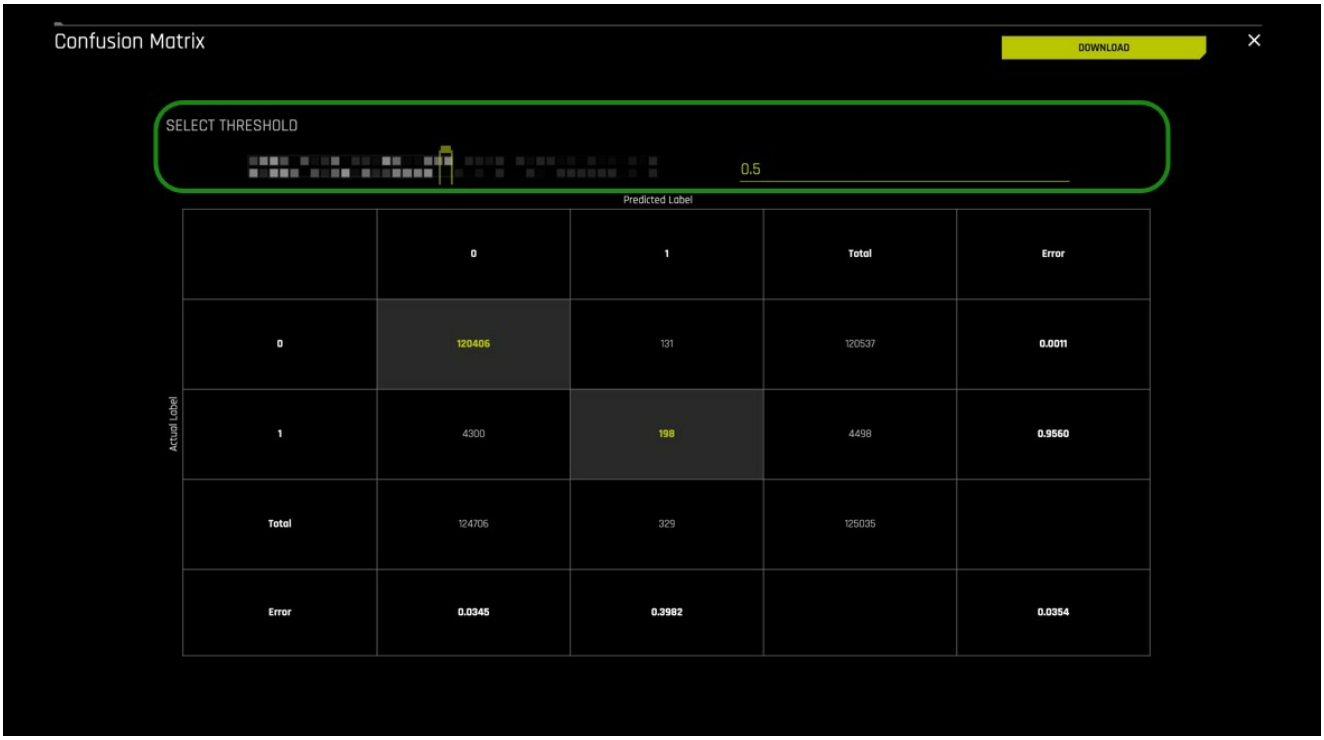
1. **Info**: Information about the diagnostics model including the name of the test dataset, name of the experiment used and the target column used for the experiment
2. **Scores**: Summary for the values for GINI, MCC, F05, F1, F2, Accuracy, Log loss, AUC and AUCPR in relation to how well the experiment model scored against a "new" dataset**Note:** The new dataset must be the same format and with the same number of columns as the training dataset
3. **Metric Plots**: Metrics used to score the experiment model including ROC Curve, Pre-Recall Curve, Cumulative Gains, Lift Chart, Kolmogorov-Smirnov Chart, and Confusion Matrix
4. **Download Predictions**: Download the diagnostics predictions

Note: The scores will be different for the train dataset and the validation dataset used during the training of the model.

Confusion Matrix

As mentioned in the concepts section, the confusion matrix is the root from where most metrics used to test the performance of a model originate. The confusion matrix provides an overview performance of a supervised model's ability to classify.

Click on the confusion matrix located on the **Metrics Plot** section of the Diagnostics page, bottom-right corner. An image similar to the one below will come up:



The confusion matrix lets you choose a desired threshold for your predictions. In this case, we will take a closer look at the confusion matrix generated by the Driverless AI model with the default threshold, which is 0.5.

The first part of the confusion matrix we are going to look at is the **Predicted labels** and **Actual labels**. As shown on the image below the **Predicted label** values for **Predicted Condition Negative** or **0** and **Predicted Condition Positive** or **1** run vertically while the **Actual label** values for **Actual Condition Negative** or **0** and **Actual Condition Positive** or **1** run horizontally on the matrix.

Using this layout, we will be able to determine how well the model predicted the people that defaulted and those that did not from our Freddie Mac test dataset. Additionally, we will be able to compare it to the actual labels from the test dataset.

Moving into the inner part of the matrix, we find the number of cases for True Negatives, False Positives, False Negatives and True Positive. The confusion matrix for this model generated tells us that :

- TP = 1 = 198 cases were predicted as **defaulting** and **defaulted** in actuality
- TN = 0 = 120,406 cases were predicted as **not defaulting** and **did not default**
- FP = 1 = 131 cases were predicted as **defaulting** when in actuality they **did not default**
- FN = 0 = 4,300 cases were predicted as **not defaulting** when in actuality they **defaulted**

The next layer we will look at is the **Total** sections for **Predicted label** and **Actual label**.

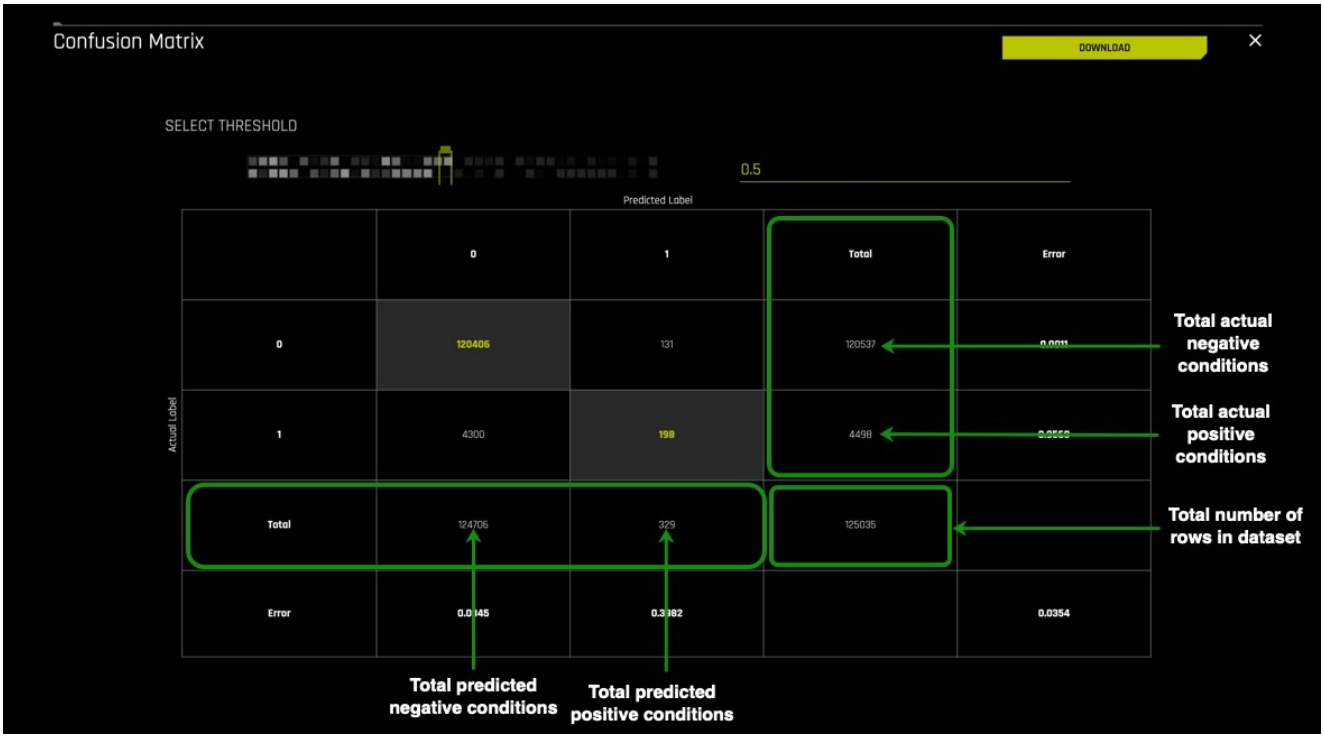
On the right side of the confusion matrix are the totals for the **Actual label** and at the base of the confusion matrix, the totals for the **Predicted label**.

Actual label

- 120,537 : the number of actual cases that did not default on the test dataset
- 4,498 : the number of actual cases that defaulted on the test

Predicted label

- 124,706 : the number of cases that were predicted to not default on the test dataset
- 329 : the number of cases that were predicted to default on the test dataset



The final layer of the confusion matrix we will explore are the errors. The errors section is one of the first places where we can check how well the model performed. The better the model does at classifying labels on the test dataset the lower the error rate will be. The **error rate** is also known as the **misclassification rate** which answers the question of how often is the model wrong?

For this particular model these are the errors:

- 131/120537 = 0.0011 or 0.11% times the model classified actual cases that did not default as defaulting out of the actual non-defaulting group
- 4300/4498 = 0.956 or 95.6% times the model classified actual cases that did default as not defaulting out of the actual defaulting group
- 4300/124706 = 0.0345 or 3.45% times the model classified predicted cases that did default as not defaulting out of the total predicted not defaulting group
- 198/329 = 0.602 or 60.2% times the model classified predicted cases that defaulted as defaulting out of the total predicted defaulting group
- (4300 + 131) / 125035 = **0.0354** This means that this model incorrectly classifies .0354 or 3.54% of the time.

What does the misclassification error of .0354 mean?
One of the best ways to understand the impact of this misclassification error is to look at the financial implications of the False Positives and False Negatives. As mentioned previously, the False Positives represent the loans predicted not to default and in reality did default.

Additionally, we can look at the mortgages that Freddie Mac missed out on by not granting loans because the model predicted that they would default when in reality they did not default.

One way to look at the financial implications for Freddie Mac is to look at the total paid interest rate per loan. The mortgages on this dataset are traditional home equity loans which means that the loans are:

- A fixed borrowed amount
- Fixed interest rate
- Loan term and monthly payments are both fixed

For this tutorial, we will assume a 6% Annual Percent Rate(APR) over 30 years. APR is the amount one pays to borrow the funds. Additionally, we are going to assume an average home loan of $167,473(this average was calculated by taking the sum of all the loans on the freddie_mac_500.csv dataset and dividing it by 30,001 which is the total number of mortgages on this dataset). For a mortgage of $167,473 the total interest paid after 30 years would be $143,739.01[1].

When looking at the False Positives, we can think about 131 cases of people which the model predicted should be not be granted a home loan because they were predicted to default on their mortgage. These 131 loans translate to over 18 million dollars in loss of potential income (131 * $143,739.01) in interest.

Now, looking at the True Positives, we do the same and take the 4,300 cases that were granted a loan because the model predicted that they would not default on their home loan. These 4,300 cases translate to about over 618 million dollars in interest losses since the 4,300 cases defaulted.

The misclassification rate provides a summary of the sum of the False Positives and False Negatives divided by the total cases in the test dataset. The misclassification rate for this model was .0354. If this model were used to determine home loan approvals, the mortgage institutions would need to consider approximately 618 million dollars in losses for misclassified loans that got approved and shouldn't have and 18 million dollars on loans that were not approved since they were classified as defaulting.

One way to look at these results is to ask the question: is missing out on approximately 18 million dollars from loans that were not approved better than losing about 618 million dollars from loans that were approved and then defaulted? There is no definite answer to this question, and the answer depends on the mortgage institution.

Scores

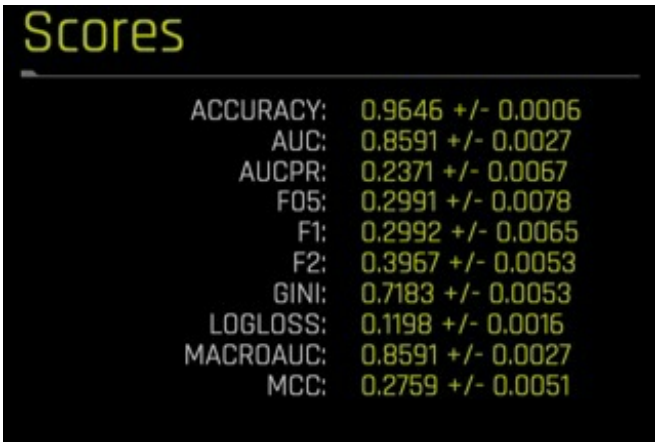
Driverless AI conveniently provides a summary of the scores for the performance of the model given the test dataset.

The scores section provides a summary of the Best Scores found in the metrics plots:

- **GINI**
- **MCC**
- **F1**
- **F2**
- **Accuracy**
- **Logloss**
- **AUC**
- **AUCPR**

The image below represents the scores for the **Freddie Mac Classification Tutorial** model using the freddie_mac_500_test dataset:

When the experiment was run for this classification model, Driverless AI determined that the best scorer for it was the Logarithmic Loss or **LOGLOSS** due to the imbalanced nature of the dataset. **LOGLOSS** focuses on getting the probabilities right (strongly penalizes wrong probabilities). The selection of Logarithmic Loss makes sense since we want a model that can correctly classify those who are most likely to default while ensuring that those that qualify for a loan get can get one.



Recall that Log loss is the logarithmic loss metric that can be used to evaluate the performance of a binomial or multinomial classifier, where a model with a Log loss of 0 would be the perfect classifier. Our model scored a LOGLOSS value = .1198+/- .0016 after testing it with test dataset. From the confusion matrix, we saw that the model had issues classifying perfectly; however, it was able to classify with an ACCURACY of .9646 +/- .0006. The financial implications of the misclassifications have been covered in the confusion matrix section above.

Driverless AI has the option to change the type of scorer used for the experiment. Recall that for this dataset the scorer was selected to be **logloss**. An experiment can be re-run with another scorer. For general imbalanced classification problems, AUCPR and MCC scorers are good choices, while F05, F1, and F2 are designed to balance recall against precision. The AUC is designed for ranking problems. Gini is similar to the AUC but measures the quality of ranking (inequality) for regression problems.

In the next few tasks we will explore the scorer further and the **Scores** values in relation to the residual plots.

References

[1] Amortization Schedule Calculator

Deeper Dive and Resources