TECH-GB.3332 Fall 2024 Prof. Alex Tuzhilin

# Term Project

**Due date:** December 3 (model submission) & December 6 (report submission)

# Introduction

In this project, you will build a model predicting the review ratings of flight experiences based on the flight review data for a group of airlines collected between 2016 and 2024 and stored in [dataset](https://drive.google.com/drive/folders/1BaotmEFxlrNjLnRpg-6uk0bVV78naWdz?usp=drive_link). Unlike Assignments 2 and 3, the final evaluation of the performance of your model will be done on a *hidden* test set to which you will only have access to the attributes but *not* to the ground truth labels (i.e., the actual values of the labels will be hidden from you). Moreover, we will post the performance scores of your model (as tested on this hidden dataset) on the Leaderboard, and all the Project Groups will *compete* against each other, as it is done in Kaggle and other types of data science competitions, such as Netflix Prize competition (e.g., see [www.kaggle.com](http://www.kaggle.com)). As in all of these competitions, you can have multiple submissions, thus constantly revising and improving your prediction model, and the best of your performance results will be shown on the Leaderboard. Finally the dataset on which you will train, finetune and test your model is publicly available to the whole class at [dataset](https://drive.google.com/drive/folders/1BaotmEFxlrNjLnRpg-6uk0bVV78naWdz?usp=drive_link).

We will describe the data, the prediction task, the project submission process, its deliverables and the logistics of the competition in the rest of this document.

# Dataset Overview

The [dataset](https://drive.google.com/drive/folders/1BaotmEFxlrNjLnRpg-6uk0bVV78naWdz?usp=drive_link) contains flight reviews for a group of airlines done between 2016 and 2024. Each record in this dataset has the format consisting of 10 attributes and 8 potential target labels. The descriptions of the review attributes are provided below:

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| Title | The title of the review. |
| Review Date | The date when the review was posted. (YYYY-MM-DD) |
| Airline | The airlines being reviewed include Qatar Airways, Emirates, Turkish Airlines |
| Name | The name of the reviewer. |
| Verified | Indicates whether the review is verified or not (Boolean variable: has values of True and False). |
| Review | The main text of the review. |
| Type of Traveler | The type of the traveler (e.g., Solo Leisure, Family Leisure, Business). |
| Month Flown | The month when the flight was taken. |
| Route | The route of the flight. |
| Class | The class of service (e.g., Economy Class, Business Class). |

The following are the performance ratings (targets or labels) measuring various aspects of the customer flight experiences:

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| Seat Comfort | Rating for seat comfort: ranges from 1 to 5 |
| Staff Service | Rating for staff service: ranges from 1 to 5 |
| Food & Beverages | Rating for food and beverage quality: range from 1 to 5 |
| Inflight Entertainment | Rating for inflight entertainment: ranges from 1 to 5 |
| Value For Money | Rating for value for money: ranges from 1 to 5 |
| Overall Rating | The overall rating given by the reviewer: is from 1 to 10 |
| Recommended | Indicates whether the reviewer would recommend the airline (Yes or No) |
| **Rating Class** | low (Overall Rating 1 to 3), mid (Overall Rating 4 to 7), high (Overall Rating 8-10) |

**[Training Dataset]** The training dataset contains 2000 records (samples) with all 18 features (10 review attributes and 8 performance ratings). You will use this training dataset to build *and* validate your model. The name of this file is: *airlines\_train\_2000.xlsx*

**[Test Dataset]** In addition to the training dataset, the test set of 500 records with *only* 10 attributes (without any performance ratings) will be used to evaluate your model performance. Your job is to improve the model performance using various types of techniques learned in the class.The *name of this file* is:*airline\_test\_2000\_student.xlsx*

# Task

The task of this project is to correctly predict the ***Rating Class*** target/label (i.e., whether the rating of a flight experience is low, mid or high) on the *test* dataset. The performance of each submission will be evaluated by calculating the *accuracy metric* between your model’s predicted values and the hidden ground truth values on the test dataset.

# Leaderboard

There will be the Leaderboard maintained in the [Gradescope](https://www.gradescope.com/courses/846424) system throughout the competition. For your information, Gradescope is an online tool (similar to Brightspace) that can automatically grade assignments using customized script(s). It is widely used in the programming courses. In case of our term project, the Gradescope system contains the ground truth labels of the test dataset at its backend (we entered this data there for the project). Once you submit to Gradescope your predicted results on the test dataset, the Gradescope Leaderboard *automatically* calculates your accuracy performance result and publishes it on the Leaderboard.

Your submission format to the Leaderboard should be in ***.xlsx***. There should be only **one column** in your submission file containing your *predicted* results on the *test* dataset, with the **header** called **“Rating Class”**. Your **submission name** should be your team’s Group Number in the format of “**Group X**”, where X is the group number,e.g., “Group 1”. Note that invalid format would lead to 0 or no score for the submission. *Only one* member of your team should submit the project, unless your teammates are added as group members, as was described in *Gradescope Submission Guidelines* below.

You will have an *unlimited* number of attempts to improve your model, submit its improved performance results and, hopefully, enhance your position on the Leaderboard. Note that you should restore your best performing result *manually* on the Leaderboard in case your newly submitted result is worse than the previous attempt by following the [submission guidelines](#clml3yl3nlca).

# Project Timeline

* November 9: At least one of your teammates should enroll in Gradescope by noon.
* November 18: Submit your first attempt on Gradescope by 11:59 PM.
* December 3: The leaderboard on Gradescope will be frozen by 5:00PM. No late submissions will be accepted after that.
* December 4: The top-3 groups will be invited to give a 5 min talk each in class on their model design.
* December 6: The final Term Project Report due on Brightspace by 11:59PM.

# Project Grading Policy and Expectations

Your final project will be graded based on the combination of the submitted report, the Leaderboard ranking, and the best model performance results.

* 5% for submitting the first attempt by November 18 (and 0 if nothing is submitted)
  + Note that the submission on November 18 is mandatory and is worth 5% of the total score. This submission aims to help you to learn the submission procedure and to make sure that you start working on the project early enough.
* 30% for the *absolute* performance result
  + Absolute performance refers to your team’s final accuracy score, as shown on the Leaderboard. Note that the accuracy rate of about 80% would be achieved by a simplistic (plain “vanilla”) model, of about 81% would be achieved by a better fine-tuned model, about 82% would constitute a good/“decent” performance, and about 83% would constitute a truly excellent (aspirational) performance.
* 15% for the Leaderboard ranking, i.e., the first place would get 15%, the second place – 14%, third – 13%, fourth – 12%, etc.
* 50% for the Project Report
  + The report should include the stuff described in the “Deliverables” section below.

**Note:** This project is limited only to the AI Studio (Rapid Miner). Performance achieved on other AI platforms will not be accepted. If you want, however, you can code your model in Python and use one of the AI frameworks, and then convert it into an AI Studio model.

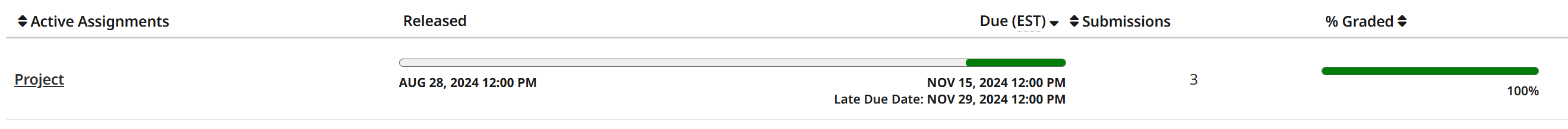
# Deliverables

* The report describing your best-performing model in detail, e.g., what you have done and why you designed your model this particular way, what your model performance results are and how you would interpret them.
* Describe briefly what you have tried before your final/best submission and what you have learned from the prior experiences and past failures.
* A zip file containing:
  + Your best model’s process with the name “*best\_model.rmp*”.
  + Your best model’s prediction in Gradescope submission format with the name “submission.xlsx”

# Gradescope Submission Guidelines

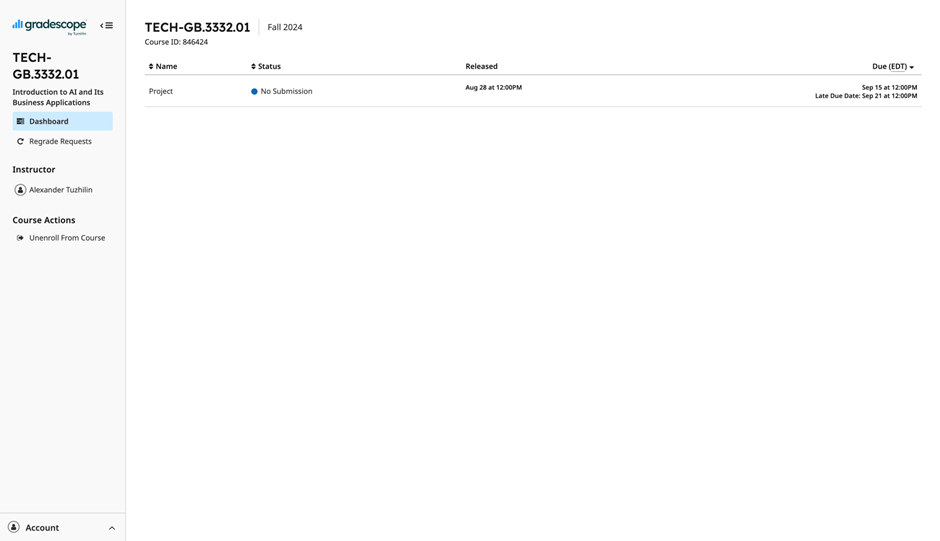
### How to Access Gradescope:

1. **Login to Gradescope and Enroll in the Course:**
   * Visit Gradescope.com
   * Register an account and enroll in the course with the Entry Code: **NYRE26**. If there is no place to enroll with an Entry Code during the registration process, then login to your account, click on “Add a course” under Student Courses section or “Enroll in Course” at the bottom of the page, and enter the Entry Code.
2. **Navigate to Your Course:**
   * After logging in, select the course from your Gradescope dashboard.
   * On the dashboard, you will see “Project” (see below). After clicking on the Project, you can see the submission option.

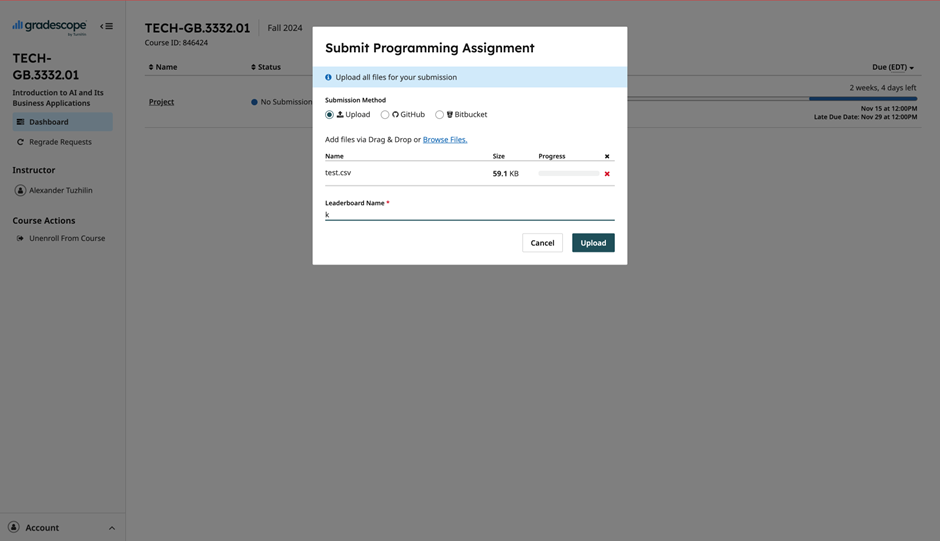


### How to Submit an Assignment

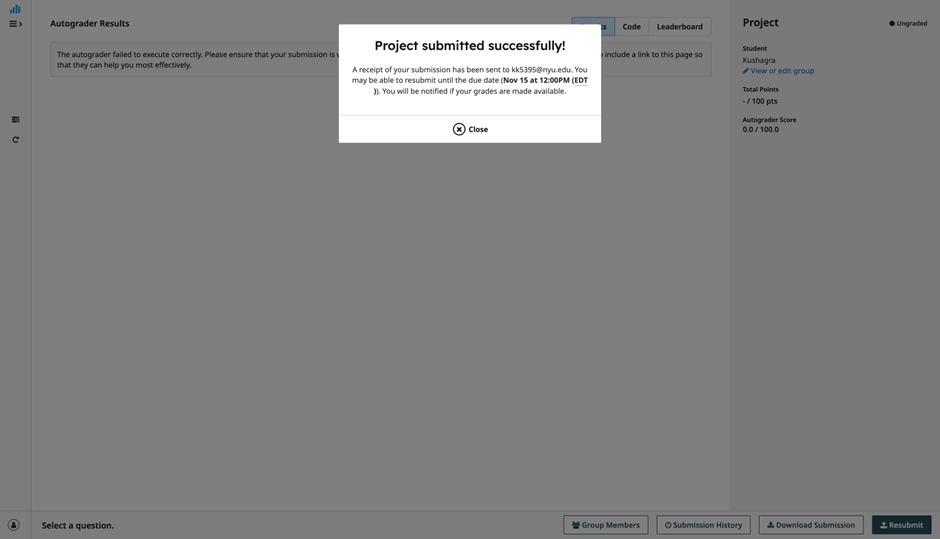
1. **Open the Assignment:** Click on the assignment title to open the submission page.



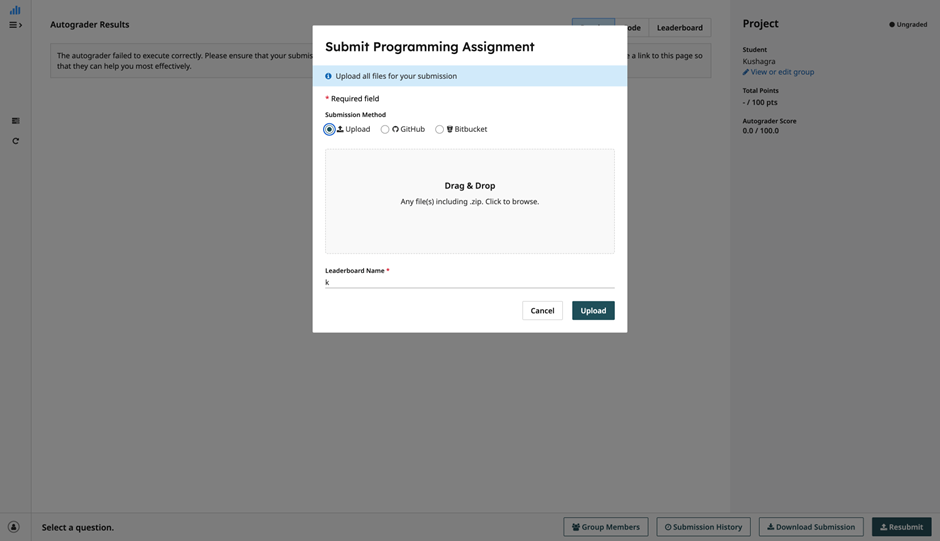
1. **Select File to Upload:**
   * Click the "Submit" button.
   * Upload the required files (only .xlsx files with header “Rating Class”). Follow the file format instructions provided in the assignment details.



1. **Confirm Submission:**
   * After uploading the file(s), make sure your Submission Name is your *Team name.* Then click on “Upload”
   * You should receive “Project submitted successfully!” confirmation (see below).



1. **Re-submissions:**
   * You can resubmit your assignment multiple times before the deadline by clicking the “Resubmit” on the bottom right. Each new submission will *replace* the previous one.



### 

### How to Group with your Teammates on Gradescope

1. Click on the Group Members option located on project submission page at the bottom next to the submission history.

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

1. Add your teammates account(s). After adding your teammates to the group, everyone in your team can submit.

### How to View Leaderboard Details

For the assignments having the leaderboard feature, the following instructions apply:

1. **Leaderboard Overview:**
   * The Leaderboard displays the top-performing submissions ranked on the grading criteria (i.e., highest accuracy scores in our case).
   * The rankings on the Leaderboard are updated every time you submit a new attempt. Note that if your latest submission has lower performance, your rank will be lower. You can always *restore* your previous submission to maintain your status by following the *Restore Past Submissions* instructions below.
2. **Viewing the Leaderboard:**
   * To view the leaderboard, navigate to the assignment after submission. You will see thr ranking section with your performance relative to others. An example of the Leaderboard is presented below:

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

### 

### How to Restore Past Submissions

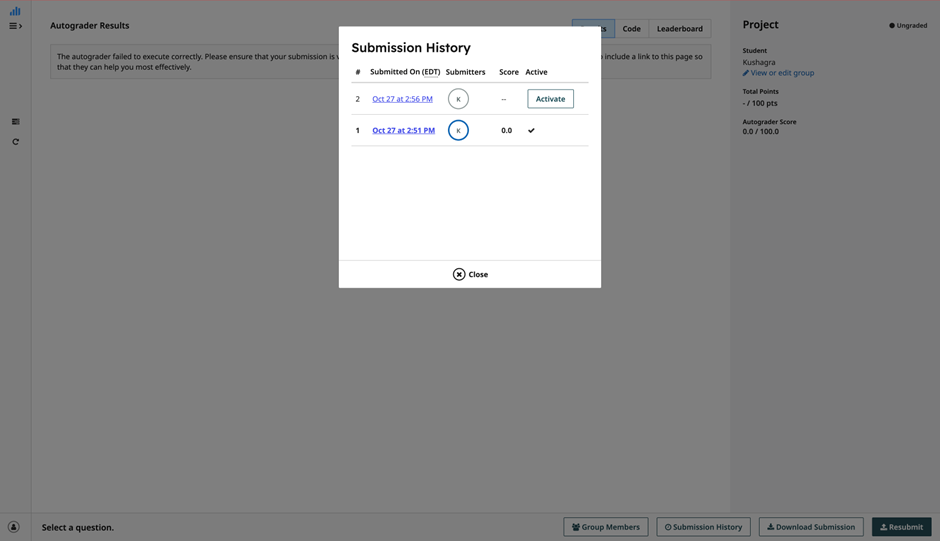
If your latest submission does not perform well (does not outperform your prior best submission), you may want to restore the prior attempt as your best-performing score. You can do this by following these instructions:

1. **Viewing Previous Submissions:**
   * You can view all your previous submissions by going to the Assignment page and selecting the “Submission History” option at the bottom.
   * Each submission is timestamped and shows the grades or feedback received for that attempt (see below).

图形用户界面, 应用程序

描述已自动生成

1. **Selecting a Previous Submission:**
   * You can choose one of your previous submissions as the best one for grading by
     + Going to the “Submission History” of the Assignment.
     + Clicking on the “Activate” button for the submission you want to restore.



### Common Issues and Troubleshooting

1. **File Upload Errors:** Make sure that your file type matches the assignment's allowed format (.xlsx) with the header named “Rating Class”
2. **Leaderboard Discrepancies:** If your latest submission does not appear on the Leaderboard, make sure that it was successfully graded and the submission file is formatted as .xlsx with the header named “Rating Class”.

# Additional Notes on the Project

## Starting the Project by Building a Simple DL Model

Although you are free to experiment in any way you want, we suggest that you start the project by *quickly* building a *simple* DL model that provides you some *initial* performance baseline (most likely, this first model will achieve accuracy rates at around 80% or less). Moreover, we recommend you to start with the models similar to the ones that we tried in the Lab and in Assignments 2 and 3. This will be your initial reference point that you can subsequently try to improve in different ways. In particular, you can modify the NN architecture and tune the hyperparameters of the model, such as the learning rate, the number of neurons in each layer, etc.

Furthermore, your choice of models and their operators is not limited to those we have studied. You are welcome to explore other functions in the AI Studio and also use the Hugging Face models as discussed below. As we have seen in Assignments 2 and 3, there are many useful operators in the AI Studio, and there are quite a few operators there that we have not explored. Navigating through the AI Studio and trying/playing with different operators could help. Here is a useful link to explore: <https://docs.rapidminer.com/latest/studio/getting-started/index.html>.

You can also right-click on specific operators in the AI Studio and click on the “Show Operator Info” option, which provides the description of the operator and a useful tutorial on how to use it.

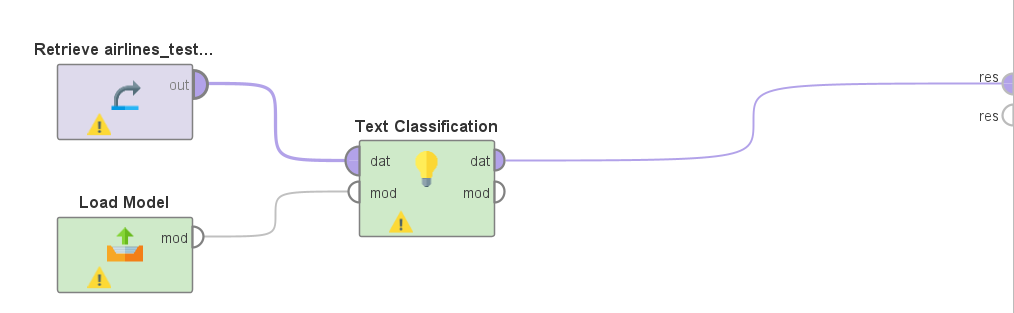
## Using Pretrained Models from Hugging Face

One possible way to improve performance of your model is to use pre-trained models from Hugging Face with the subsequent fine-tuning of these models using the dataset provided for this Term Project – similar to the way it was done in Part 1 of Assignment 3. We do not insist on your doing it; but you may want to try and see if anything good comes out of this idea.

## How to Generate Predictions for Your Model on the Test Dataset

* 1. **Fine-tuning the Models**

If you decide to use pre-trained models from Hugging Face, as described above, you can generate predictions for the fine-tuned model as in Assignment 3 and as shown below.

****

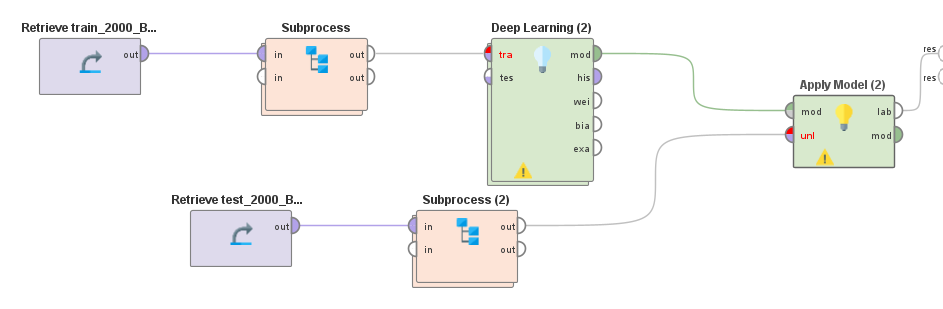
***Load Model*** [*Path: Generative Models/Models/Huggingface*]: modify the “project folder” parameter to the location of your fine-tuned model that you want to use for testing the dataset.

**Text Classification** [*Path: Generative Models/Models/Huggingface/Tasks*]: Note that you used the Text Classification operator in both of your homework assignments already, and you should proceed similarly here. As in these homeworks, remember to check the “use local model” box and use the correct prompt.

After the above process is run (by pressing the Run button), its performance results are reported in the Results panel. To export these results to Excel, use the “Write Excel” operator as it was done in Assignment 2 (the Embedding section). When submitting your results, make sure that the format follows the previously specified requirements: .xlsx file with header named “Rating Class”.

* 1. **The Deep Learning Models**

After you find the best hyperparameters setting (such as the learning rate, the number of neurons in each layer, the number of epochs, etc.) and determine the proper NN architecture for your model (such as the number of layers in the network and its other characteristics), you will have to *rerun* the Deep Learning model training process to generate predictions on the *test* dataset. Remember to process the test dataset the same way you process your training data to avoid formatting errors (see below).

****

As before, after you run the process above, its performance results can be observed in the Results panel (the dataset and the predicted values of the “Rating Class” target variable). To export the results in Excel, use the “Write Excel” operator mentioned in Assignment 2 (the Embedding section). When submitting your result, make sure that the format follows the requirements described in the [*Submission Format for Leaderboard*](#bi4eb7abgwla)above*.*

## Save Time by Properly Using the Download and the Load Model Operators

***Download Model*** [*Path: Generative Models/Models/Huggingface*]: Use the Hugging Face URL to download the pre-trained model from Hugging Face, as described in the homework assignments and in the Lab.

***Load Model*** [*Path: Generative Models/Models/Huggingface*]: Load the *saved* model by selecting the correct path for the project folder parameter. For example, once you downloaded a Hugging Face pre-trained model, you can change the “Download Model” operator to the “Load Model” operator in the process workflow to save time by avoiding repeated downloads. Furthermore, the “Load Model” operator can be used to avoid *repeated* fine-tuning processes when making predictions on the test dataset.