ST2MLE Machine learning for IT Engineers
PROJECT
Binary Classification Identifying Age-Related Conditions with weighted multi-class logarithmic loss evaluation metric

Note that this project is derived from the ongoing competition initiated by InVitro Cell Research, LLC on Kaggle.
While we highly encourage your participation in the competition, it is not mandatory; it is entirely your choice.
Your decision regarding participation will not impact your project grade. Your project grade will be based solely on the deliverables you submit on Moodle and your final presentation.

Context – Use Case

They say age is just a number but a **whole host of health issues come with aging**. From heart disease and dementia to hearing loss and arthritis, aging is a risk factor for numerous diseases and complications. The growing **field of bioinformatics** includes research into **interventions that can help slow and reverse biological aging and prevent major age-related ailments**. Data science could have a role to play in developing new methods to solve problems with diverse data, even if the number of samples is small.

Currently, models like XGBoost and random forest are used to predict medical conditions yet the models' performance is not good enough. **Dealing with critical problems where lives are on the line, models need to make correct predictions reliably and consistently between different cases.**

In this project, you will work with measurements of health characteristic data to solve critical problems in bioinformatics.

Based on minimal training, you will create a model to predict if a person has any of three medical conditions (**B, D, G**), with an aim to improve on existing methods. The condition **A** corresponds to the No age-related condition.
!! *Note that due to confidentiality issues, InVitro Cell Research did not provide the original features and more background information about the data. The features are annotated with letters as listed below.*

You could help advance the growing field of bioinformatics and explore new methods to solve complex problems with diverse data.

This is a **Binary classification problem**. Remember that binary classification is the task of classifying the elements of a set into two groups (each called class) on the basis of a classification rule.

Dataset Description

The data comprises over fifty anonymized health characteristics linked to the three age-related conditions (B, D, G).

The objective is to predict whether a person has been diagnosed with any of these conditions or not, which can be framed as a binary classification task.

Thus your goal is to predict whether a subject has or has not been diagnosed with one of these conditions.

In other words, it a binary classification problem where the positive class (1) represents a person who has been diagnosed with at least one of these conditions, and the negative class (0) represents the person who has not been diagnosed with none of these conditions.

Remember that a probabilistic classifier is a classifier that is able to predict, given an observation of an input, a probability distribution over a set of classes (in your cases two), rather than only outputting the most likely class that the observation should belong to.

Note that for the competition, they ask you to make **probabilistic** predictions about whether a subject has been diagnosed with a specific condition or not.

!!Note: As part of this project, you have the freedom to utilize classification models that directly predict the target class. Additionally, you need to use other models that provide the probability of the subject belonging to a certain class.

!!Advice: Note that if you choose to participate in the competition and submit your solutions, you are required to submit only the implementation of the probabilistic classifier models.

Files and Fields Description:

Note that due to confidentiality issues, InVitro Cell Research did not provide the original features and more background information about the data. The features are annotated with letters as listed below.

The main dataset that you need to use is train.csv:

train.csv - The training set.

- **Id:** Unique identifier for each observation.
- **AB…GL:** Fifty-six anonymized health characteristics. All are numeric except for EJ, which is categorical.
- **Class A binary target: 1** indicates the subject has been diagnosed with one of the three conditions, **0** indicates none of the three conditions is detected.

Additional datasets:

greeks.csv - Supplemental metadata, only available for the training set.

- **Alpha:** Identifies the type of age-related condition, if present.
  - **A**: No age-related condition. Corresponds to class 0.
  - **B, D, G**: The three age-related conditions. Correspond to class 1.
- **Beta, Gamma, Delta**: Three experimental characteristics.
- **Epsilon**: The date the data for this subject was collected. Note that all of the data in the test set was collected after the training set was collected.

test.csv - The test set. Your goal is to predict the probability that a subject in this set belongs to each of the two classes. Attention: this dataset does not include the target column. Only features data are present.

sample_submission.csv - A sample submission file in the correct format. If you decide to participate to the competition, you need to take this submission sample into consideration. For more info, see the Competition Evaluation page for more details.

Project Instructions and Recommendations

To solve this project, you need to follow the following steps, after you understand the context and objectives of the project :

1. **Import the needed libraries**

2. **Load the Dataset**

3. **Start with some EDA (exploratory data analysis)**

4. **Based on what you observe in the EDA, decide how would you need to pre-process your data: for example:**

   - **Deal with missing data is exist**
   - **Deal with outliers if exist**
   - **Decide whether you need to scale and/or normalize your data (this is based on the models that you are expecting to fit on your data)**
   - **Feature extraction and engineering**

5. Noting that your dataset includes fifty-six anonymized health characteristics, do you need to consider all of them in your analysis ? Do not forget that you have to **untangle highly correlated explanatory variables**, as seen in the lecture. For this reason, **you need to conduct a correlation analysis between the explanatory variables to decide how to apply your features selection, extraction, dimensionality reduction…**

6. **Split your data (train.csv) into train set and test set**.

7. **Create your classification model and fit it on your train dataset. In this project, you are going to test multiple classification models (at least 4), compare their performance and decide which is seen the best in the context of this case study.**

Remember that in case you decide to participate to the ongoing Kaggle competition, you are required to only use probabilistic classifier models as explained previously.
!!Note: As part of this project, you have the freedom to utilize classification models that directly predict the target class. Additionally, you need to use other models that provide the probability of the subject belonging to a certain class.

Thus, you have two options:

- **Whether to apply 4 probabilistic classifier models.**
- **Or to apply at least 2 probabilistic classifier models out of the 4 that you will select.**

Here is a list of the most commonly used probabilistic classifier models:
There are various algorithms that can be used for class predictor probability in machine learning. The specific algorithm used depends on the nature of the problem, the available data, and the preferences of the researcher or practitioner. Here are a few commonly used algorithms for class predictor probability:

Logistic Regression: Logistic regression is a statistical model used for binary classification problems. It estimates the probability of a binary outcome based on input features by fitting a logistic function to the data.

Support Vector Machines (SVM): SVM is a popular algorithm for both binary and multi-class classification. It constructs a hyperplane that maximally separates different classes in a high-dimensional feature space.

Random Forest: Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. It can be used for both classification and regression tasks. In classification, Random Forest computes class probabilities based on the proportion of trees that vote for a particular class.

Gradient Boosting Models (e.g., XGBoost, LightGBM): Gradient Boosting is another ensemble learning technique that sequentially builds a series of weak learners, typically decision trees, to make predictions. It can be used for classification and regression tasks and provides probabilities based on the cumulative predictions of the individual weak learners.

Neural Networks: Deep learning models, specifically neural networks, can also be used for class predictor probability. Neural networks consist of multiple layers of interconnected nodes (neurons) and are capable of learning complex relationships in the data. The output layer of a neural network can provide class probabilities using appropriate activation functions (e.g., softmax).

Note that there are many other algorithms available for class predictor probability. The choice of algorithm depends on factors such as the dataset size, dimensionality, linearity of the problem, interpretability requirements, and computational resources

available. It's often recommended to experiment with different algorithms and select the one that yields the best performance for the specific task at hand.

8. Do not forget that it is highly recommended to **apply validation** as well **using cross validation techniques or other techniques to better configure your models with the optimal hyperparameters**…

9. **Make predictions and evaluate your model**.

   As seen in the lectures and previous lab activities, there are many evaluation metrics for classification models that can be used to evaluate the classification models. The choice of metrics depends on the specific requirements of the problem and the balance needed between precision and recall. Some of the commonly used evaluation metrics for classification models that we recommend you to use are:

   - **Accuracy**: Measures the overall correctness of the model's predictions by calculating the ratio of correctly classified instances to the total number of instances.
   - **Precision**: Also known as positive predictive value, it represents the proportion of correctly predicted positive instances out of all instances predicted as positive. It focuses on the accuracy of positive predictions.
   - **Recall** (Sensitivity or True Positive Rate): Measures the proportion of correctly predicted positive instances out of all actual positive instances. It focuses on the model's ability to identify positive instances.
   - **F1 Score**: A metric that combines precision and recall into a single value. It is the harmonic mean of precision and recall and provides a balanced assessment of a model's performance.
   - Specificity (True Negative Rate): Measures the proportion of correctly predicted negative instances out of all actual negative instances. It focuses on the model's ability to identify negative instances.
   - **Area Under the ROC Curve (AUC-ROC)**: Evaluates the performance of a classification model by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various classification thresholds. The AUC-ROC represents the probability that a randomly chosen positive instance will be ranked higher than a randomly chosen negative instance.
   - **Area Under the Precision-Recall Curve (AUC-PR)**: Similar to AUC-ROC, but plots precision against recall at various classification thresholds. AUC-PR provides a summary of the model's performance across different trade-offs between precision and recall.
   - **Weighted Multi Class Log Loss:**

     Other that the above listed evaluation metrics, you need to use the **Weighted Multi Class Log Loss**. Note that this is the metric that should be considered if you decide to participate to the competition.

     Weighted Multi Class Log Loss is an evaluation metric used to assess the performance of a multi-class classification model. It calculates the logarithmic loss for each predicted class and assigns a weight to each class based on its

importance. The weighted aspect of the metric accounts for class imbalance by assigning higher weights to underrepresented classes and lower weights to overrepresented classes. This helps prevent the model from favoring the majority class and encourages balanced predictions across all classes. The logarithmic nature of the loss function penalizes incorrect predictions more severely, providing a more nuanced evaluation of the model's performance.

To better understand the Weighted Multi Class Log Loss, you may check the following links:

https://www.kaggle.com/code/alejopaullier/weighted-multi-class-log-loss-explained

https://www.kaggle.com/competitions/icr-identify-age-related-conditions/overview/evaluation

NB: Do not forget to add internal comments for interpretation and explanation in each step. This is mandatory and will be graded.

As a recommendation, we suggest that you explore the ongoing discussions among data scientists regarding the competition topic. This will provide you with inspiration and insights into how others have tackled similar challenges related to this project. It can be beneficial for gaining awareness and finding potential solutions to issues you may encounter:https://www.kaggle.com/competitions/icr-identify-age-related-conditions/discussion

Project Deliverables and Presentation

The deliverables should include : cleaned dataset + Python Notebook (including your use case description, source code and internal comments for explanation and interpretation of the results) + PowerPoint presentation

The deadline for submitting your deliverables is set for 8 AM on the day of your final session.

The presentation of your work is to be done in class during the final session. The schedule of your project presentations will be shared with you by your instructor.
Note: All the group members should participate.
Every group has 10 minutes of presentation including demo and additional 5 minutes for Q/A.
Your presentation should include the following:

- quick explanation of the context of the project and objectives, and the use case (very brief).
- explanation to support your choice of the classifier models that you selected.
- demonstration of the code and the implementation.
- interpretation of the results (model performance, …).

Good Luck !!