MIE 1624 Introduction to Data Science and Analytics – Winter 2023

Assignment 2
Due Date: 11:59pm, March 12, 2023
Submit via Quercus

Background:

For this assignment, you are responsible for answering the questions below based on the dataset provided. You will then need to submit a 3-page report in which you present the results of your analysis. In your report, you should use visual forms to present your results. How you decide to present your results (i.e., with tables/plots/etc.) is up to you but your choice should make the results of your analysis clear and obvious. In your report, you will need to explain what you have used to arrive at the answer to the research question and why it was appropriate for the data/question. You must interpret your final results in the context of the dataset for your problem.

Background:

In this assignment, we will work on the "**2022 Kaggle Machine Learning & Data Science Survey**" dataset.

The purpose of this challenge was to "*tell a data story about a subset of the data science community represented in this survey, through a combination of both narrative text and data exploration.*" More information on this competition can be found on: https://www.kaggle.com/competitions/kaggle-survey-2022

The dataset provided (kaggle_survey_2022_responses.csv) contains the survey results provided by Kaggle. The survey results from 23997 participants are shown in 296 columns, representing survey questions. Not all questions are answered by each participant, and responses contain various data types. In the dataset, column 'Q29' *"What is your current yearly compensation (approximate $USD)?"* contains the **ordinal categorical** target variable. The original data (kaggle_survey_2022_responses.csv) has been transformed to **clean_kaggle_data_2022.csv** as per the code given in **KaggleSalary_DataSet.ipynb**. In the dataset to be used for Assignment 2 (**clean_kaggle_data_2022.csv** – file to be read in notebook for this Assignment, **You should work with the clean dataset for this assignment**), rows with the null values of salaries have been dropped. In addition, two columns ('Q29_Encoded' and 'Q29_buckets') have been added at the end. Column 'Q29_buckets' (Target Variable for Assignment 2) has been obtained by combining some salary buckets in the column 'Q29'. Column 'Q29_Encoded' has been obtained by label encoding the column 'Q29_buckets'.

The purpose of this assignment is to train, validate, and tune multi-class ordinal classification models that can predict a survey respondent's current yearly compensation bucket, based on a set of survey responses by a data scientist.

Classification is a supervised machine learning approach used to assign a discrete value of one variable when given the values of others. Many types of machine learning models can be used for training classification problems, such as logistic regression, decision trees, kNN, SVM, random forest, gradient-boosted decision trees, and neural networks. In this assignment, you are **required to use the ordinal logistic regression algorithm**, but feel free to experiment with other algorithms.

For the purposes of this assignment, **any subset of data can be used** for data exploration and for classification purposes. For example, you may focus only on one country, exclude features, or engineer new features. If a subset of data is chosen, it **must contain at least 5000 training examples**. You must **justify and explain** why you are selecting a subset of the data, and how it may affect the model.

Data is often split into training and testing data. The training data is typically further divided to create validation sets, either by just splitting, if enough data exists, or by using **cross-validation** within the training set. The model can be iteratively improved by tuning the hyperparameters or by feature selection.

Learning objectives:

1. Understand how to clean and prepare data for machine learning, including working with multiple data types, incomplete data, and categorical data. Perform data standardization/normalization, if necessary, prior to modeling.
2. Understand how to apply machine learning algorithms (ordinal logistic regression) to the task of classification.
3. Improve on skills and competencies required to compare performance of classification algorithms, including application of performance measurements, and visualization of comparisons.
4. Understand how to improve the performance of your model.
5. Improve on skill and competencies required to collate and present domain specific, evidence-based insights.

Questions:

The following sections should be included but the order does not need to be followed. The discussion for each section is included in that section's marks.

1. **[1 pt] Data cleaning:**

   While the data is made ready for analysis, several values are **missing**, and some features are **categorical**. Note that some values that appear "null" indicate that a survey respondent did not select that given option from a multiple-choice list. For example – "*Which of the following hosted notebook products do you use on a regular basis? (Select all that apply) - Selected Choice - Binder / JupyterHub*".

   For the data cleaning step, handle missing values however you see fit and ***justify your approach***. Additionally, provide some insight on why you think the values are missing and how your approach might impact the overall analysis. Suggestions include filling the missing values with a certain

value (e.g. mode for categorical data) and completely removing the features with missing values. Another method could be filling them with a separate value showing missingness, e.g., "unknown". Secondly, convert categorical data into numerical data by encoding and explain why you used this particular encoding method.

These tasks can be done interchangeably, e.g., encoding can be done first.

Your submission must include the following:

- Data cleaning code that handles missing values and categorical features (with proper justification, discarding part of the features is acceptable as well) (in .ipynb)
- Explanation about each of your data cleaning step and justification of your approach (PDF)

2. [3 pts] Exploratory data analysis and feature selection:

Explain how feature engineering is a useful tool in machine learning in the context of the tasks in this assignment. Apply feature engineering and then select the features to be used for analysis either manually or through some feature selection algorithm (e.g., regularized regression).

Not all features need to be used; features can be removed or added as desired. If the resulting number of features is very high, dimensionality reduction can also be used (e.g., PCA). Use at least one feature selection technique – describe the technique and ***provide justification*** on why you selected that set of features.

For the exploratory data analysis step, visualize the order of feature importance. Some possible methods include correlation plot, or a similar method. Given the data, which of the original attributes in the data are most related to a survey respondent's yearly compensation?

Your submission must include the following:

- Exploratory data analysis code that visualizes the order of feature importance (in .ipynb)
- Feature engineering/selection code (in .ipynb)
- Explanation about your feature engineering/selection technique above and justification of your approach (PDF)

3. [4 pts] Model implementation:

Implement **ordinal logistic regression** algorithm on the *training data* using **10-fold cross-validation**. How does your model accuracy compare across the folds? What is the average and variance of accuracy for folds? Treating each value of hyperparameter(s) as a new model, which model performed best? Give the reason based on bias-variance trade-off. An output of your algorithm should be a probability of belonging to each of the salary buckets. Apply scaling/normalization of features, if necessary, and justify the reason why scaling/normalization is (not) needed.
Hint: If you are applying scaling/normalization on the data, apply the technique on the testing and training data separately.

4. **[3 pts] Model tuning:**

   Identify all the hyperparameters in your model. Select two hyperparameters for model tuning and justify your selection. Improve the performance of the models from the previous step with hyperparameter tuning and select a final optimal model using grid search based on a metric (or metrics) that you choose. Choosing an optimal model for a given task (comparing multiple classifiers on a specific domain) requires selecting performance measures, for example accuracy, precision, recall and/or F1-score to compare the model performance. Find a description of these metrics at the end of this file. Explain why accuracy cannot be a suitable performance metric for this problem. There is no minimum model performance, as long as your methodology is reasonable and well explained.
   Create the feature importance graph of your model to see which features were the most determining in model predictions. Compare this graph with the feature importance graph obtained in Section 2.

5. **[4 pts] Testing & Discussion:**

   Use your optimal model to make classifications on the *test set*. (Note that the test set should not be used in any form during the training process, even as a validation set.) How does your model perform on the test set vs. the training set? The overall fit of the model, how to improve the performance (test, training)? Is it overfitting or underfitting? Why? Plot the distribution of true target variable values and their predictions on both the training set and test set. What insight have you gained from the dataset and your trained classification model?

   Insufficient discussion will lead to the deduction of marks.

Submission:

1) Produce an IPython Notebook (.ipynb file) detailing the analysis you performed to answer the questions for the given data set.

2) Produce a 3-page report explaining your response to each question for the given data set and detailing the analysis you performed. When writing the report, make sure to explain for each step, what you are doing, why it is important, and the pros and cons of that approach.

Tools:

- **Software:**
  - **Python Version 3.X** is required for this assignment. Make sure that your Jupyter notebook runs on Google Colab (https://colab.research.google.com) portal. All libraries are allowed but here is a list of the major libraries you might consider: Numpy, Scipy, Sklearn, Matplotlib, Pandas.
  - No other tool or software besides Python and its component libraries can be used to touch the data files. For instance, using Microsoft Excel to clean the data is **not allowed**.
  - Upload the required data file to your notebook on Google Colab – for example,
    ```
    from google.colab import files
    ```

```
uploaded = files.upload()
```

- **Required data files:**
  - **clean_kaggle_data_2022.csv**: file to be read in notebook for this Assignment
  - The data file cannot be altered by any means. The notebook will be run using the local version of this data file. Do not save anything to file within the notebook and read it back.

- Auxiliary files:
  - **kaggle_survey_2022_responses.csv**: original survey responses.
  - **kaggle_survey_2022_answer_choices.pdf**: the questions and answer choices in the survey.
  - **kaggle_survey_2022_methodology.pdf**: the methodology and survey flow logic of the survey.
  - **KaggleSalary_DataSet.ipynb**: the code used to transform the original survey responses (**kaggle_survey_2022_responses.csv**) to the clean dataset (**clean_kaggle_data_2022.csv)**

What to submit:

1. Submit via Quercus a Jupyter (IPython) notebook containing your implementation and motivation for all the steps of the analysis with the following naming convention:
   lastname_studentnumber_assignment2.ipynb
   Make sure that you **comment** your code appropriately and describe **each step** in sufficient detail. Respect the above convention when naming your file, making sure that all letters are lowercase and underscores are used as shown. **A program that cannot be evaluated because it varies from specifications will receive zero marks.**
2. Submit a report in PDF (up to 3 pages) including the findings from your analysis. Use the following naming conventions **lastname_studentnumber_assignment2.pdf**.

Late submissions will receive a standard penalty:

- up to one hour late - no penalty
- one day late - 15% penalty
- two days late - 30% penalty
- three days late - 45% penalty
- more than three days late - 0 mark

Other requirements and tips:

1. A large portion of marks are allocated to analysis and justification. Full marks will not be given for the code alone.
2. Output must be shown and readable in the notebook. The only files that can be read into the notebook are the files posted in the assignment **without** modification. All work must be done within the notebook.

3. Ensure the code runs in full before submitting. Open the code in Google Colab and navigate to Runtime -> Restart runtime and Run **all** Cells. Ensure that there are no errors.
4. Do not re-run cross-validation (it can run for a very long time). When cross-validation is finished, output (print) the results (optimal model parameters). Hard-code the results in the model parameters and comment out the cross-validation code used to generate the optimal parameters.
5. You have a lot of freedom with how you want to approach each step and with whatever library or function you want to use. As open-ended as the problem seems, the emphasis of the assignment is for you to be able to ***explain the reasoning behind every step***.
6. The output of the classifier when evaluated on the training set must be the same as the output of the classifier when evaluated on the testing set, but you may clean and prepare the data as you see fit for the training set and the testing set.
7. When evaluating the performance of your algorithm, keep in mind that there can be an inherent trade-off between the results on various performance measures.

A Brief Introduction into the Most Common Performance Metrics: (source: https://medium.com/@MohammedS/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b)

Accuracy: refers to the total number of correct predictions over the total number of predictions.
Precision: refers to the total number of true positive predictions over the total number of datapoints predicted as positive.
Recall or Sensitivity: refers to the total number true positive predictions over total the number of all datapoints with actual positive labels.
Specificity: refers to the total number of true negative predictions over the total number of all datapoints with actual negative labels.
F1-score: refers to the harmonic mean of precision and recall.