MIE 1624 Introduction to Data Science and Analytics – Fall 2024

Assignment 2
Due Date: 11:59pm, November 10, 2024
Submit via Quercus

Background:

For this assignment, your task is to analyze the provided dataset and answer questions listed below. You will then write a 3-page report to present results of your analysis. In your report, make use of visual aids to effectively convey your findings. The format of your visualizations (i.e., with tables/plots/etc.) is up to you, but ensure they clearly communicate the results of your analysis. In your report, explain how you arrive at the answers to the questions and justify why your answers are reasonable for the given data/question. You must interpret your final results in the context of the dataset for your problem.

Introduction:

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

Kaggle, a platform known for its data science competitions and datasets, annually conducts Kaggle Machine Learning & Data Science Survey, most comprehensive dataset available on the state of Machine Learning and Data Science. Based on the survey, Kaggle also launched a challenge/competition. 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 at https://www.kaggle.com/competitions/kaggle-survey-2022

The dataset (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 Jupyter 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 values of the 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 implement 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 discard some rows (data samples) for data cleaning purposes. 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.

You may get started with this assignment using **assignment2_template.ipynb**. The template contains some basic data analysis procedures that might be helpful for you, i.e., reading the dataset, and a skeleton for implementing ordinal logistic regression. Note that the filename has to be **renamed** properly before submission is made (see later sections for details).

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. **[2 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***. 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 those 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.

   In your PDF report, for the features you cleaned, provide some insight on ***why*** you think the values are missing and ***how*** your approach might impact the overall analysis. You can choose a *subset* of features to use in later questions if you think that is reasonable and can work on data cleaning for only those features – however, you should NOT discard many features without sufficient justification!

   Your submission must include the following:

   - Data cleaning code that handles missing values and categorical features (in .ipynb)
   - Explanation about each of your data cleaning steps and justification of your approach (in PDF report).
     - You don't need to explain the data cleaning steps for each *feature*. You can group the features with similar cleaning steps and explain the cleaning and encoding you applied to each group and why – that will be enough.

   Hint: Take a close look at the dataset before tackling this question. What is the meaning of each column? Be careful when you interpret missing values within the dataset.

2. **[3.5 pts] Exploratory data analysis and feature selection:**

   In this question, you explore how feature selection and feature engineering are useful tools in machine learning in the context of the tasks in this assignment. Work on **exploratory data analysis**

to see what features seem to be important. Then, 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).

For the exploratory data analysis, visualize the order of **feature importance**. Based on the feature importance plot, conclude which of the original attributes in the data are most related to a survey respondent's yearly compensation.

Not all features need to be used in the later analysis; 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 is a dimensionality reduction technique, but is not effective for categorical features – think about what kind of other techniques can be used). Use at least one feature selection technique – describe the technique and ***provide justification*** on why you selected that set of features.

Your submission must include the following:

- Exploratory data analysis code that visualizes the order of feature importance (in .ipynb) and insights on which original features are most important to predict a respondent's salary (in PDF report) (1pt)
- Feature engineering/generation code that creates new feature(s) from existing ones. You may use domain knowledge or external data, though it's optional. The features don't have to improve the model, but you should explain why you believed those might help with prediction. (0.5pt)
- Implementation of the feature selection technique of your choice (in .ipynb) (1pt)
- Explanation of the feature selection technique above and justification of your approach (in PDF report) (1pt)

3. [3.5 pts] Model implementation:

3.1. Implement **ordinal logistic regression** algorithm. (The skeleton is provided in the notebook.) (1pt)

3.2. Perform **10-fold cross-validation** on the *training* data. How does your model accuracy compare across the folds? Report the average and variance of accuracy across folds. You can use default hyperparameter values. (1pt)

3.3. Identify one hyperparameter that has a direct impact on the bias-variance trade-off. Tweaking this hyperparameter (add this hyperparameter to the `OrdinalLogisticRegression` class), provide an analysis of the model performance based on bias-variance trade-off. Conclude which value is the best in terms of bias-variance trade-off. You can leave other hyperparameters default. (1pt)

3.4. Is scaling/normalization of features needed for our task? Apply scaling/normalization if necessary, and justify the reason why scaling/normalization is (not) needed. If you are applying scaling/normalization on the data, make sure you apply the technique on the testing and training data separately. (0.5pt)

4. [3 pts] Model tuning:

4.1. Choosing a proper criterion to determine the "best-performing" model for a given task requires selecting performance measures, for example accuracy, precision, recall and/or F1-score to compare the model performance. You can find a description of these metrics at the end of this file. Explain **why accuracy cannot be a suitable performance metric** for this problem. (0.5pt)

4.2. Identify all the hyperparameters that the ordinal logistic regression model can have (check out the logistic regression model implementation in scikit-learn). Select **two hyperparameters** for model tuning and justify your selection. Improve the performance of the ordinal logistic regression model and select a final best-performing model, using **grid search** based on a metric (or metrics) chosen in Question 4.1. There is no requirement of the minimum model performance, as long as your model implementation and tuning are done correctly and well explained. (1.5pt)

4.3. **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. (1pt)

Hint: How do you extract feature importance from your ordinal logistic regression model? Think about a reasonable representation that highlights which features your model relied on for prediction.

5. [3 pts] Testing & Discussion:

5.1. Use your best-performing 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.) Report the performance on the test set vs. the training set. (0.5pt)

5.2. Looking at the overall fit of the model, how would you further improve the performance (test, training)? Is it overfitting or underfitting? Why? (1pt)

5.3. Plot the distribution of true target variable values and their predictions on both the training set and test set. (1pt)

5.4. What insights have you gained from the dataset and your trained classification model? (0.5pt)

Insufficient discussion will lead to the deduction of marks.

Recommended steps to get started:

1) Download `assignment2_template.ipynb` from Quercus
2) Go to Google Colab (https://colab.google/), click on 'Open Colab', and upload `assignment2_template.ipynb`
3) Start working on `#TODO` in the template (note that `#TODO` does not cover every step from Questions 1-5. Read each question carefully and make sure you answer all the questions.)

Submission:

1) Produce an IPython Notebook (.ipynb file) containing your implementation and analyses you performed to answer the questions for the given data set. Make sure you have brief comments to every step of your analysis, so that we know what analysis you did. **Your Jupyter notebook needs to run on Google Colab!** Add the following two lines at the **beginning** of your notebook so you can work with the provided data on Colab:

```
from google.colab import files
uploaded = files.upload()
```

When you check your code before submission, select the 'Kernel' tab and then 'Restart & Run All' to make sure that all the codes run without any errors. **If your code does not run properly on Google Colab, substantial marks will be deducted.**

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 each step, what you are doing, why it is important, and the pros and cons of that approach. You can have an appendix for additional figures and their captions and cite them in the report if you need more space. Figures and tables should be properly formatted. While there is no specific grading criterion for writing quality, unclear/inaccurate statements and figures that fail to convey the justification of your answers, may lead to a partial deduction of points.

What to submit:

1. Submit via Quercus a Jupyter (IPython) notebook with the following naming convention:
   lastname_studentnumber_assignment2.ipynb
   Make sure that you **comment** on 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. **If a program cannot be evaluated because of errors, or because it varies from specification, you will receive zero marks.**
2. Submit a report in PDF (up to 3 pages + appendix) including the findings from your analysis. Use the following naming conventions **lastname_studentnumber_assignment2.pdf**.

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()
```

- **Data file:**
    - **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 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)**

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. You may not want to 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.

Appendix

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.