Task Description for Data Science

This is a task description for the Data Science topic. It should be read in conjunction with the overall **Test** [**Project and Report**](https://ncl.instructure.com/courses/39977/assignments/154862) specification.

**Scenario**

You have been hired to consult for a winery in Portugal. They produce varieties of their traditional "Vinho Verde" and would like to understand what really determines their quality, so they can optimise their production. There are three datasets:

[**winequality-red.csv**](https://ncl.instructure.com/courses/39977/files/5499015/download?download_frd=1)****

[**winequality-white.csv**](https://ncl.instructure.com/courses/39977/files/5499012/download?download_frd=1)

****[**winequality.names**](https://ncl.instructure.com/courses/39977/files/5499017/download?download_frd=1)

****

The two datatests include 4899 and 1600 data points, respectively. Each data point is a set of 12 variables; 11 of these are physicochemical measures, and the last is the perceived **quality** (sensory output).

A fraction of each dataset is reserved for independent validation and will be used to make predictions on submissions, in addition to to checking that learning is done correctly (i.e., by separating training and test sets etc.).

**Task**

Your task is to create one or more models to predict quality given all other variables. To achieve this, you will:

1. **Explore each of the two datasets**

[Do this by performing aggregations, computing summary statistics (using **pandas**](https://pandas.pydata.org/docs/user_guide/index.html)

[**(https://pandas.pydata.org/docs/user\_guide/index.html)** ), and plotting (with e.g. **seaborn**](https://seaborn.pydata.org/api.html)[**(https://seaborn.pydata.org/api.html)** or **matplotlib Pyplot**](https://matplotlib.org/stable/tutorials/introductory/pyplot.html)[**(https://matplotlib.org/stable/tutorials/introductory/pyplot.html)**](https://matplotlib.org/stable/tutorials/introductory/pyplot.html) ) the data. Specifically, you should be able to:

A. Describe the distribution of wine quality across all samples, separately for red and white, and compare the quality distributions between reds and whites. Create suitable plots to illustrate.

B. Discretise the alcohol content variables (separately for whites and reds) into low, mid, high based on its distribution. Create a 3-valued "alcohol\_cat" variable to represent this.

1/4

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | Task Description for Data Science: Introducing Contemporary Topics in Computing |
|  |  |  | low < (average - [**stddev (https://numpy.org/doc/stable/reference/generated/numpy.std.html)**](https://numpy.org/doc/stable/reference/generated/numpy.std.html) |
|  |  |  |
|  |  |  |
|  |  |  | ) |
|  |  |  | (average - stddev) < mid < (average + stddev) |
|  |  |  |
|  |  |  |
|  |  |  | high > (average + stddev). |
|  |  |  |
|  |  |  |

C. Describe the distribution of wine quality as in (1.A), but separately for low-, mid-, and high-alcohol content. Create suitable plots to illustrate. Can you draw any conclusions on the relationship between alcohol content and quality?

D. Plot the residual sugar variable and identify a suitable threshold to separate "sweet" from "dry" wines\*. Create a new "isSweet" binary variable to represent these two classes. The distributions of residual sugar are skewed for both reds and whites (in fact most wines in this [dataset are dry according to the official definition, e.g., **https://winefolly.com/deep-dive/sugar-in-wine-chart/) (https://winefolly.com/deep-dive/sugar-in-wine-chart/))** .](https://winefolly.com/deep-dive/sugar-in-wine-chart/)) Apractical approach in this case is to pick a threshold that splits the dataset (almost) evenly, as that will give you two balanced classes for your classifier. So your task is to find a threshold such that each class has approximately the same number of records.

E. Using the threshold from (1.D), repeat the distribution analysis of quality vs isSweet. Are sweet wines perceived as lower or higher quality than dry wines?

1. **Try and determine which subset of your variables can be most useful for learning** This is in preparation to applying machine learning to create your model**.** To do this, analyse correlations between:

 Each pair of variables

 Each variable and the outcome (**quality**)

 [Produce a visual representation of the Correlation Matrix, using either **seaborn**](https://seaborn.pydata.org/generated/seaborn.heatmap.html)

[**(https://seaborn.pydata.org/generated/seaborn.heatmap.html)** or **matplotlib**](https://matplotlib.org/stable/gallery/images_contours_and_fields/image_annotated_heatmap.html)[**(https://matplotlib.org/stable/gallery/images\_contours\_and\_fields/image\_annotated\_heatmap.ht**](https://matplotlib.org/stable/gallery/images_contours_and_fields/image_annotated_heatmap.html)

[**ml)** . Comment on how some of the variables may relate to others. For this task, you may use](https://matplotlib.org/stable/gallery/images_contours_and_fields/image_annotated_heatmap.html) Pandas methods to automatically create correlation matrices, for

[example **pandas.DataFrame.corr() (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html)** . Take care when](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html) **using** the correctmetrics, i.e. you must choose between {‘pearson’, ‘kendall’, ‘spearman’}. Consult the documentation to learn about these metrics.

1. **Experiment with one or more machine learning approaches** This creates a predictive model for quality. Some options are:

 Consider this as a **classification** problem. You can do this by considering the quality labels, ie. '5', '6', ... '9' and reducing the number of possible labels to two (binary classification). You should experiment with different thresholds and compare results, for instance define "low" to be quality <6, "high" >=6, then change the threshold to 5 or 7, etc.

 Consider this a **regression** problem, where quality is now a continuous variable.

1. [**Evaluate each model using k-fold cross validation (https://scikit-**](https://scikit-learn.org/0.16/modules/generated/sklearn.cross_validation.cross_val_score.html)

[**learn.org/0.16/modules/generated/sklearn.cross\_validation.cross\_val\_score.html)**](https://scikit-learn.org/0.16/modules/generated/sklearn.cross_validation.cross_val_score.html)

2/4

Task Description for Data Science: Introducing Contemporary Topics in Computing

Report the model's performance on both the training set and test set, using appropriate metrics [for the kind of model chosen. For example, a binary classifier can be evaluated using **f1-score**](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html)[**(https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1\_score.html)** , **ROC curves**](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html)

[**(https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc\_curve.html)** , **AUC**](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.auc.html)[**(https://scikit-learn.org/stable/modules/generated/sklearn.metrics.auc.html)** . A regression](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.auc.html) **model** is

[usually evaluated using **MSE (https://scikit-**](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html)

[**learn.org/stable/modules/generated/sklearn.metrics.mean\_squared\_error.html)** , RMSE and **related**](https://scikit-learn.org/stable/modules/model_evaluation.html#regression-metrics)

[**"error" metrics (https://scikit-learn.org/stable/modules/model\_evaluation.html#regression-metrics)**](https://scikit-learn.org/stable/modules/model_evaluation.html#regression-metrics)

.

**Extensions**

There are several areas where you can go further:

Additional descriptive analysis:



1. Using both isSweet and alcohol\_cat, create suitable plots to characterise quality with respect to these two variables (hint: see notebook used in the practical, where seaborn functions were used for this).
2. Are these quality distributions different for reds and whites? What happens if you put them together and disregard the colour?

Use your correlation analysis above (Task 2) to decide which subset of variables you are going to use for learning. Consider that:



Using two variables that are highly correlated with each other is redundant. Consider training two models using only one of the two, compare performance, and choose which variable is more effective.



Using variables that are uncorrelated with the outcome is generally not useful Using variables that have very small variance across the training set is not useful



Additional Learning, e.g. what happens if you try and predict *all possible quality labels* as opposed to just the two labels "low" and "high"?



**What to Include in Your Report**

The What Was Done and How section of your report could include:

A critical analysis of the choices you made in Tasks 1, 2, 3 above, and how they affected your model.



[Comments on your results: which approach works best, and why? Is there any **overfitting**](https://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html)

****

[**(https://scikit-learn.org/stable/auto\_examples/model\_selection/plot\_underfitting\_overfitting.html)**](https://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html) ?

Can you think of ways to improve performance while still generalising well?

You should include a combination of code cells and narrative cells (like comments on code, using [**markdown (https://www.markdownguide.org/cheat-sheet/)**](https://www.markdownguide.org/cheat-sheet/) ) in your notebook. This allows you to construct a complete "Data Science story" of your Test project.



3/4

Task Description for Data Science: Introducing Contemporary Topics in Computing

It is your decision as to what to include to demonstrate the work you have done.