Big Data and Machine Learning with Python - Exercises

Michael E. Rose, PhD
Course at LMU, March 2019

All question can and should be answered in teams of up to two people (that is, you answer as a team).

Create one GitHub repository for your team with each team member being a collaborator. For each question below, create one script that contains both code and answers (as comments at the end of the script). All scripts need to adhere to PEP8 and must be readable to someone that knows Python.

Save the script in the main folder, properly named in correspondence to the name of the exercise. Apart from the script, there should be one folder named "output" to store output such as figures and tables.

1 Exercises for Pandas, APIs and Plotting

Sketch for today: French Taunting

1. Coding workflow
   - Read "Code and Data" by M. Gentzkow and J. Shapiro, chapters 2, 3 and 4.
   - What brought Gentzkow and Shapiro to the conclusion, that version control is a necessity?

2. Alcohol
   - Load the data from `./data/drinks.csv` into a DataFrame. The data is on country's alcoholic consumption.
   - Which continent drinks most beer and wine on average?
   - Create a Boxenplot of "beer_servings" by continent.
   - Reshape the data and create a 3x1 figure for "beer_servings", "wine_servings" and "spirit_servings" by continent (i.e. three boxenplots that share their y-axis and a their color coding).
   - Save the figure as `./out/alcohol.pdf`

3. Tips
   - Load seaborn's tips dataset using `seaborn.load_dataset("tips")`.
   - Create a lineplot of "tips" on "total_bill" with markers, line styles and color by "day", and facets by "sex".
   - Label the axis so that the unit becomes apparent.
   - Add a title to the legend and use the full day of the week-name (i.e. "Thursday" instead of "Thu").
   - Save the figure as `./out/tips.pdf`

4. Occupations
   - Import the pipe-separated dataset from `https://raw.githubusercontent.com/justmarkham/DAT8/master/data/u.user` into a DataFrame. The data is on occupations and demographic information.
   - Set "user_id" as index and name the index "User".
   - Print the last 10 entries and the first 25 entries.
   - What is the type of each column?
   - How many different occupations are there in the dataset? What is the most frequent occupation?
   - What are the age values with the least occurrence?
   - Create a histogram for occupations, sorted alphabetically.
   - Save the figure as `./out/occupations.pdf`

5. Euro 2012 I
   - Read the data from `./data/Euro_2012.csv` into a DataFrame with column "Teams" as index. The data is on the UEFA Champtionship 2012 (Euro 2012).

- How many teams played in the Euro 2012?
- Which team has the highest shooting accuracy?
- Plot shooting accuracy versus passing accuracy.
- Which team has the second-most shots on target?
- Eliminate Italy from the dataset. Which team has the second-most shots on target now?
- How many penalty goals did England score?
- Present only the Shooting Accuracy from England, Italy and Russia.
- Create a new DataFrame called discipline using the columns "Yellow Cards" and "Red Cards" (and the index).
- Sort discipline primarily by red cars and secondarily by yellow cards.
- Output the data as tab-separated textfile `./out/discipline.tsv`.

6. Iris

- Read the Iris dataset from `https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data` directly from the Internet.
- Name the columns in the following way: "sepal_length (in cm)", "sepal_width (in cm)", "petal_length (in cm)", "petal_width (in cm)" and "class".
- Set values of the rows 10 to 29 of the column 'petal_length (in cm)' to missing.
- Replace missing values with 1.0.
- Save the comma-separated file as `./out/iris.csv` without index.
- Visualize the distribution of all of the continuous variables by "class" with a catplot of your choice.
- Save the figure as `./out/iris.pdf`.

7. Memory

- Load the comma-separated data from `https://query.data.world/s/wsjbxdqhw6z6izgdxijv5p2lfqh7gx` into a DataFrame.
- Inspect the DataFrame using `.info()` and with `.info(memory\_usage="deep")`. What is the difference between the two calls? How much space does the DataFrame require in memory?
- Create a copy of the object with only columns of type object by using `.select\_dtypes(include=['object'])`.
- Look at the summary of this object new (using `.describe()`). Which columns have very few unique values compared to the number of observations?
- Does it make sense to convert a column of type object to type category if more than 50% of the observations contain unique values? Why/Why not?
- Convert all columns of type object to type category where you deem this appropriate.
- What is the final size in memory?
- Could above routine have speeded up somewhere? Hint: Look at the documentation for .read_csv().

8. Euro 2012 II

- Load the data from `./data/Euro_2012.csv` into a DataFrame.
- Add a column "Wikipedia" displaying the Wikipedia page ID of the country (use `.apply()` on column "Team" to apply the corresponding function which queries Wikipedia individually).
- Output the data as semicolon-separated `./out/wikipedia.ssv`.

2 Exercises for Supervised Machine Learning

Sketch for today: Machine that goes ping

1. Feature Engineering
   - Load the Boston House Price dataset using `sklearn.datasets.load_boston()`
   - Extract polynomial features (without bias) and interactions up to a degree of 2.
   - How many features do you end up with?
   - Create a pandas DataFrame using the polynomials.
   - Use the originally provided feature names to generate names for the polynomials (Hint: `.get_feature_names()` accepts a parameter) and use them as column names.
   - Add the dependent variable to the DataFrame and name the column "y".
   - Save the DataFrame as comma-separated textfile named `./out/polynomials.csv`.
2. Regularization
   - Read the data from exercise "Feature Engineering" (`./out/polynomials.csv`) into a DataFrame.
   - Use column "y" as target variable and all other columns as predicting variables and split them as usual.
   - Learn a Ridge model and a Lasso model using the provided data with default parameters. Why do you get a ConvergenceWarning? What are the accuracy scores?
   - Create a DataFrame containing the learned coefficients of both models and the feature names as index. In how many rows are the Lasso coefficients equal to 0 while the Ridge coefficients are not?
   - Using `matplotlib.pyplot`, create a horizontal bar plot of dimension 10x30 showing the coefficient sizes.
   - Save the figure as `./out/polynomials.pdf`.
3. Random Forest Classification
   - Load the breast cancer dataset using `sklearn.datasets.load_breast_cancer()`.
   - Create a histogram for the target variable.
   - Split the data as usual into a test and training set.
   - Learn a Random Forest model for the following parameters using 5-fold Cross-Validation (i.e. `GridSearchCV()`: 10 estimators, 50 estimators, 100 estimators (a pipeline is not necessary). What is the best parameter?
   - Print the confusion matrix for the best model to screen. Does it look good?
   - For the best model, print accuracy score, precision, recall and f1-score.
   - Plot the feature importances as a simple barplot (with axis names) and save it as `./out/forest_breast_importances`
4. Neural Networks Classification
   - Load the breast cancer dataset using `sklearn.datasets.load_breast_cancer()`. As usual, split the data into test and training set.
   - Read up about the Area Under the Curve of the Reciever Operating Characteristic Curve, the so-called ROC-AUC-metric, i.e. at towardsdatascience.com.
   - Learn a a Neural Network after MinMax-Scaling with in total four parameter combinations (and 1000 iterations) of your choice using 5-fold Cross-Validation. Use the ROC-AuC-score metric to pick the best model.
   - Plot a heatmap for the mean test scores of your pipeline and label the axes ticks using your used values.
   - Plot a heatmap of the confusion matrix for the best model (Hint: the best estimator has a property `_final_estimator`).
5. Neural Network Regression
   - Load the diabetes dataset using `sklearn.datasets.load_diabetes()`. The data is on health and diabetes of 442 patients. Split the data as usual.
   - Learn a a Neural Network with 1000 iterations, lbfgs-solver and tanh-activation after Standard-Scaling with in total nine parameter combinations of your choice using 4-fold Cross-Validation.
   - What are your best parameters? How well does it perform?
   - Plot a heatmap showing the coefficients with their variable names and save it as `./out/nn_diabetes_importances.p` (Hint: the best estimator has a property `_final_estimator`).

3 Exercises for Unsupervised Machine Learning

Sketch for today: Village Witch

- Principal Component Analysis
  - Read the textfile `./data/olympics.csv` into a DataFrame using the first column as index. The data lists the individual performances of 33 male athlets during various competitions of the 1988 Olympic summer games.
  - Print summary statistics for each of the variables.
  - Scale the data such that all variables have unit variance.
  - Fit a plain vanilla PCA model. Store the components in a DataFrame to display the loadings of each variable. Which variables load most prominently on the first component? Which ones on the second? Which ones on the third? How would you thus interpret those components?
  - How many components do you need to explain at least 90% of the data? (Hint: use `np.cumsum()` for this.)

- Clustering
  - Load the iris dataset using `sklearn.datasets.load_iris()`. The data is on classifying flowers.
  - Scale the data such that each variable has unit variance.
  - Assume there are three clusters. Fit a K-Means model, an Agglomerative Model and a DBSCN model (with min_sample equal to 2 and eps equal to 1). Store the cluster assignments in a DataFrame.
  - Read up about the Silhouette Score. What does it do? Compute the silhouette scores using `sklearn.metrics.silhou` for each cluster type versus the scaled features. Why do you have to drop noise assignments from the DBSCAN? Which clustering method has the highest Silhouette score?
  - Add sepal width and petal length including the corresponding original names to the DataFrame. (Beware of the dimensionality.)
  - Plot a three-fold scatter plot using sepal width as x-variable and petal length as y-variable, with dots colored by the cluster assignment and facets by cluster algorithm. (Hint: Melt the DataFrame using columns "sepal length (cm)" and "sepal width (cm)" as id variables.)