DEPAUL UNIVERSITY College of Computing and Digital Media

DSC 478 - Programming Machine Learning Applications

Fall 2025

Assignment 1

Due Date: Sunday, September 28

For this assignment you will experiment with Python. NumPy, and Pandas in order to perform some basic data preprocessing and exploratory analysis tasks. You will use a modified version of the Adult Census Data Set. In the subset provided here, some of the attributes have been removed and some initial preprocessing has been performed. You must only use **Python**, **NumPy**, **Pandas** to perform the preprocessing and analysis tasks for this assignment. For visualization you should use **Matplotlib** (or you may use the **Seaborn** visualization library if you are familiar with it).

1. **[5 pts]** Download the data set **adult-modified-09-13-2025.csv** and load it into an appropriate data structure such as a Pandas dataframe. Explore the general characteristics of the data as a whole: examine the means, standard deviations, and other statistics associated with the numeric attributes and frequencies associated with categorical attributes.

2. **[10 pts]** Using Pandas, compute the number of missing values for each attribute in the data. Fill the missing values for all numeric attributes using the mean value for the attribute. After filling in the missing numeric values, drop all rows where a categorical attribute contains a missing value. Next, remove columns **education** and **native-country**. Show that the final resulting dataframe does not contain missing values and display the new mean and standard deviations for the numeric attributes. For the remaining parts of this assignment, you should use this new dataframe without missing values and the aforementioned columns.

3. **[10 pts]** For the three numeric attributes (**age**, **hours-per-week**, **education-num**), display box plots that show the overall dispersion and skew in these variables. Next, create histograms for these three variables showing the overall data distribution in each. Finally, display a scatter plot of **age** (x-axis) vs. **education-num** (y-axis)**.**

4. **[10 pts]** Create bar charts for all the categorical attributes in the data that show the distribution of category frequencies (e.g., Male vs. Female for the **sex** attribute, Private vs. Public vs. Self-emp, etc. for the **workclass** attribute, and so on).

5. **[10 pts]** Perform a cross-tabulations of each of the **workclass** and **race** attributes with the **income** attribute. Show the resulting cross-tab tables as well as bar charts to visualize the relationships between these pairs of attributes. [Hint: you can use aggregation functions in Pandas such as **cross-tab()**, then either using Matplotlib directly or the **plot()** function in Pandas create the bar charts]. As an illustration, consider this example **graph depicting the cross-tabulation of sex with income** (Note: this example is based on a different data set, so yours will be different). In the case of **race** vs. **income** cross-tab, create another chart comparing the percentages of each race category that fall in the low-income group.
6. **[10 pts]** Characterize the population group who work in the private sector and who have an education level of less than a bachelor's degree (i.e., **education-num** values of less than 13). You may consider first creating a separate dataframe consiting of this subset of the data. Then provide an analysis of the key chracteristics of this population group based on the statistics and data distributions of other attributes.

7. **[15 pts]** Compare and contrast the characteristics of the low-income and high-income categories across the different attributes. As in the previous problem you may consider first creating separate subsets of the data based on these income categories and then characterizing each subset by observing summary statistics for each group across different variables. Discuss your observations focusing specifically on unique characteristics that seem to distinguish between the two groups. You may use charts or plots for visualizing the differences in your analysis to support your observations. [**Note:** the discussion of your observations about the key characteristics will be a significant part of the score for this problem.]

8. **[5 pts]** Convert the data into the **standard spreadsheet format**. Note that this requires converting each categorical attribute into multiple binary ("dummy") attributes (one for each values of the categorical attribute) and assigning binary values corresponding to the presence or not presence of the attribute value in the original record). The numeric attributes should remain unchanged. Save this data in a new dataframe and show the top 10 rows in the new dataframe. Also save this new table into a local file called **adult_numeric.csv**.

9. **[10 pts]** Using the numeric data set with the dummy variables (of the previous part), perform basic correlation analysis among the attributes. You need to construct a complete Correlation Matrix (with rows and columns

corresponding to each variable). [**Hint:** you can create the correlation matrix by using the **corr()** function in Pandas or **corrcoef** function in NumPy]. Next, using your correlation matrix, display in decreasing order of correlations, all attributes and their correlations to **education-num**. Repeat this step to display correlations with the attribute **income_<=50K**. Briefly discuss your general observations about this sample of adult population based on this correlation analysis.

10. **[5 pts]** Discretize the age attribute into 3 categories (corresponding to "young", "mid-age", and "old"). Do not change the original age attribute or add the discretized age to the table. Create a new dataframe with the numeric and the discretized age attributes as two columns and display the top 10 rows of the new dataframe.

11. **[10 pts]** Use **Min-Max Normalization** to transform the values of the attribute **education-num** the range 0.0-1.0 (without changing the original data). Next, perform **zscore normalization** (on the original data) to standardize the values of all numeric attributes (**age**, **hours-per-week**, **education-num**). The latter step should be performed on all three attributes at the same time instead of one-by-one (you may wish to first create a separate dataframe with only these attributes and perform the operation on the whole dataframe. Note: for this problem, you should write your own code to perform the normalization; do not use pre-existing functions such as scikit-learn's MinMaxScaler(). Finally, show the top 10 rows of the three versions of the **hours-per-week** attribute (original, normalized, and standardized) side-by-side in a new dataframe.

Notes on Submission: You must submit your Jupyter Notebook (similar to examples in class) which includes your documented code, results of your interactions, and any discussions or explanations of the results. Please organize your notebook and label sections so that it's clear what parts of the notebook correspond to which problems in the assignment (**submissions that are not well-organized, not well-documented, or are difficult to read will be penalized**). Please submit the notebook in both IPYNB and HTML formats (along with any auxiliary files). **Do not compress or Zip your submission files**; each file should be submitted independently. Your assignment should be submitted via D2L.
