LASSONDE SCHOOL OF ENGINEERING | YORK U

Assignment 2
Introduction to Artificial Intelligence and Logic Programming
Prepared by: Dr. Ruba Alomari

Contents

Instructions 1

Part 1 – Predict Adult Income. 1

Dataset 1

Tasks 1

Submission: 4

Part 2 – Write Prolog Queries and Rules. 4

Database 4

Tasks 5

Submission: 5

Instructions

- This is an individual assignment.
- Do not share this assignment document or upload it to online sharing websites. Doing so violates the academic integrity policy.
- Copying another person's work is a breach of the academic integrity policy. Credit the source and author of any code you reuse.

Part 1 – Predict Adult Income.

Dataset

Use the Adult training dataset available at https://archive.ics.uci.edu/dataset/2/adult to predict whether an adult's income exceeds $50K/yr based on census data.

Before proceeding with this assignment, read about the adult dataset and get yourself familiar with the dataset information from the above link.

Tasks

Task 1 (5 Marks): Load the dataset from the following url:
https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data

Note that at no point in this assignment, should you save the dataset locally to your machine. Doing so will result in zero marks for this assignment.

Ensure to add column headers if they are not present in the dataset.

Check https://www.askpython.com/python/examples/read-data-files-in-python if you need a refresher on how to use Pandas to read .data files.

Task 2 (5 Marks): Take a quick look at the data structure (i.e., X) using `.head()`, `.info()`, `.describe()`, and `.shape`.

Task 2.1: Plot a histogram of the data.

Task 3 (10 Marks): There are missing values in this dataset that are entered as a ? , check for and display the number of these missing values.

Tip: The number of missing values should be similar to the ones shown below.

```
age                0
workclass       1836
fnlwgt             0
education          0
education-num      0
marital-status     0
occupation      1843
relationship       0
race               0
sex                0
capital-gain       0
capital-loss       0
hours-per-week     0
native-country   583
dtype: int64
```

Task 3.1: Replace the ? character missing values you found in the previous step with null (nan), and run a .isna().sum() on your dataframe.

Tip: Your output should look similar to the one below at this step:

```
[20]  adult_df.isna().sum()
      ✓ 0.0s

...   age                0
      workclass       1836
      fnlwgt             0
      education          0
      education-num      0
      marital-status     0
      occupation      1843
      relationship       0
      race               0
      sex                0
      capital-gain       0
      capital-loss       0
      hours-per-week     0
      native-country   583
      income             0
      dtype: int64
```

Task 4 (5 Marks): Check for duplicate rows, and remove them.

Task 5 (10 Marks): Create and apply a preprocessing pipeline to:

1. Fill in the missing numerical values with the mean using a `SimpleImputer`.
2. Scale the numerical columns using `StandardScaler`. Do not scale the target.
3. Fill in the missing categorical values with the `most_frequent` value using `SimpleImputer`.
4. Encode the categorical columns using `OneHotEncoder`. Do not encode the target.

- Display your pipeline.
- Print your dataframe `.shape`.

Tip: Your dataframe.shape at this point should be `(32537, 106)`.

Task 6 (5 Marks): Print out the target `value_counts()`. Check if the target needs any kind of data cleaning. For example, suppose your target has labels that look like <=50K**.** and <=50K, in which case you will need to remove the **.** from the first value, and replace all the instances that are <=50K**.** with <=50K .

Task 7 (10 Marks): Split the data into 80% training set and 20% testing set, print the shape of `X_train, X_test, y_train, y_test` in one command.

Tip: Shapes should be (26029, 105) (26029,) (6508, 105) (6508,).

Task 8 (25 Marks): Train a `svm` classifier (`svc`) to predict if the income of the adult exceeds 50K on the training set using: `kernel = rbf, gamma = 1,` and `C =0.1`.

Note due to the large training dataset size, this will take some time.

Task 8.1: Test your model on the X_Test, and report the `classification_report` on the `y_test` and `y_predict`.

Task 8.2: Plot the confusion matrix of your test results using `ConfusionMatrixDisplay.from_predictions(y_test, y_predict)`

Task 8.3: Repeat Tasks 8.1 and 8.2 with C=1. Repeat both tasks again with C=10.

Compare and discuss the results. What was the effect of changing the value of C? Which C value will you choose for the final model and why? Answer this question in a markdown cell in your notebook.

Submission:

Submit a url to your notebook, as well as your notebook using the following instructions:

1- Run your notebook, and save it with the results.
2- Upload your notebook to your github. Submit the url to your github notebook in eclass. In the submission dropbox on eclass there is a space for you to paste your url.
   - Failure to submit a working github link -50 Marks.
3- Submit to eclass your source code notebook **.ipynb** with the code already run and the output results included and saved in your notebook. Note that the dataset in your notebook should be loaded from the uci url. Do not save the dataset to or load it from your local drive.
   - Failure to submit a working notebook -50 Marks
4- Name your submitted notebook as: A2-FirstName-LastName.ipynb.
5- In your notebook make sure to list:
   - The question/task you are answering in a markdown cell before each code section (similar to the end-to-end example posted to eclass)
   - Comment your code.
   - Submit a clean notebook free of errors and do not include any extra commands that are not needed to perform the tasks:
     i. Extra commands that are not necessary to perform the task will result in 5 marks deduction per extra command.
     ii. Errors in the output cells will result in 5 marks deduction per error.
     iii. Write efficient code.

Code Hints: There are many ways you can write code, below are some hints to be used :

- Use (adult_df == 'character').sum() or adult_df.isin(['character']).sum(axis=0) to find the number of a specific character per column (i.e., if the character is ? for example)
- Use df.replace("character", np.nan, inplace=True) to replace a character value.

Part 2 – Write Prolog Queries and Rules.

Database

Download the movies.pl database attached to this assignment. The database has the following predicates:

```
movie(M, Y) <- movie M came out in year Y
director(M, D) <- movie M was directed by director D
actor(M, A, R) <- actor A played role R in movie M
actress(M, A, R) <- actress A played role R in movie M
```

Tasks

Task 1 (18 Marks): Write queries to answer the following questions.

1. In which year was the movie American Beauty released?
2. Find the movies released in the year 2000.
3. Find the movies released before 2000.
4. Find the movies released after 1990.
5. Find an actor who has appeared in more than one movie.
6. Find a director of a movie in which Scarlett Johansson appeared.

Task 2 (7 Marks): Task 2. Add a rule to the database to define a released_after predicate such that:
released_after(M, Y) <- the movie was released after the given year.

Submission:

- Submit a .pdf file with your answers, name the file: A2-FirstName-LastName.pdf.