BIG DATA PROJECT 3

Project outline(translated from Greek)

Author:
Benekos Konstantinos, 1067754, up1067754@ac.upatras.gr

March 23, 2024

Subject 1

Download from the website 'UCI Machine Learning Repository,' and specifically from the webpage (link) the dataset containing observations regarding crime rates per 100,000 inhabitants in areas of the USA (variable ViolentCrimesPerPop), along with socio-economic characteristics for each area (Communities and Crime Data Set). The dataset is located in the file communities.data, which you will find if you follow the link 'Data Folder' on the aforementioned webpage. The website also provides information about the variables in the dataset. You can find the same information in the file 'Data Set Description' from the above website to see the order and meaning of each variable in the dataset. After familiarizing yourself with the dataset, its attributes, and their significance, estimate the coefficients of the following regression model.

$$ViolentCrimesPerPop = \beta_1 medIncome + \beta_2 whitePerCap +$$

$\beta_3 blackPerCap + \beta_4 hispPerCap + \beta_5 NumUnderPov +$
$\beta_6 PctUnemployed + \beta_7 HousVacant + \beta_8 MedRent +$
$\beta_9 NumStreet + \beta_0$ (1)

In regards to the following:

- I) Write a program in R and Python which estimates the coefficients of the above linear regression model using the Ordinary Least Squares (OLS) method, using the Communities and Crime dataset you have downloaded. Your programs should display on the screen the values of the estimated coefficients. Your programs should also remove any observations that have at least one missing value during the preprocessing phase. Follow such preprocessing of the data for all topics of this assignment.

- II) Write a program only in Python that estimates the coefficients of the above linear regression model using the Batch Gradient Descent, using the Communities and Crime Data Set. To do this, in your Python program, create a function named batchGradientDescent that accepts the following parameters and calculates the coefficients of a multiple linear regression model using the Batch Gradient Descent method:

  $$\text{batchGradientDescent}(\text{independentVars}, \text{dependentVar}, \text{thetas}, \alpha = 0.01, \text{numIters} = 100) \quad (2)$$

  Where independentVars is the matrix of values of the independent variables, dependentVar is the matrix of values of the dependent variable, thetas is a vector with the initial values of the coefficients ϑ, alpha is the value of the learning parameter α, and numIters is the number of iterations that the method should perform. The implementation of the Batch Gradient Descent method should be done from scratch using the update formula to estimate the coefficients of linear regression models. The function should not use any existing Python library that provides the Batch Gradient Descent method. The termination criterion of the Batch Gradient Descent method is the number of iterations.

  The process of estimating the coefficients should terminate after the specified number of iterations determined by the numIters argument. The batchGradientDescent function you will create should return both a vector of the estimated coefficients and the cost function values at each iteration of the Batch Gradient Descent method.

After implementing the batchGradientDescent function, use it to estimate the coefficients of the linear regression model mentioned above for the same dataset (Communities and Crime). Set appropriate values for the learning parameter α and the number of iterations numIters for estimating the coefficients. Your program should, after estimating the coefficients, display: **a)** The values of the estimated coefficients, **b)** the values of the cost function as a function of the number of iterations, graphically plotted, demonstrating that you have chosen the correct values for the learning parameter α and the number of iterations.

Furthermore, compare the coefficients estimated using the Batch Gradient Descent method with those estimated by the Ordinary Least Squares (OLS) method in question i). What observations/comments can you make?

Subject 2

Write a program in R and Python that estimates the coefficients of a linear regression model using the Stochastic Gradient Descent method. Use the programs in R and Python that you have written, setting appropriate values for the learning rate parameter α and the number of iterations, and estimate the coefficients of the linear regression model mentioned in question 1 of the current assignment for the same dataset (Communities and Crime Data Set). Your programs in R and Python should display on the screen:

- a) The estimated coefficients
- b) The graphical representation of the cost function values as they have emerged from the execution of the Stochastic Gradient Descent method you have implemented.

What observations can you make both about the coefficients as well as about the graphical representation of the cost function values obtained using the Stochastic Gradient Descent method when compared to the coefficients and cost function values obtained in subquestion II) of task 1 of the current assignment?

Subject 3

"From the file named 'Chapter-06-Exercises.pdf' which you can find in the section 'Lecture 3: Regression analysis' on the course website(eclass, answer the questions of Exercise 19.

Subject 4

From the file named 'Chapter-06-Exercises.pdf' which you can find in the section 'Lecture 3: Regression analysis' on the course website (eclass), answer the questions of Exercise 58.

Subject 5

"Download from the website(link) a dataset for forest fires in areas of Portugal. The data include geographical and meteorological information when fires occurred as well as the area burned, measured in hectares. With the aim of predicting the burned area based on prevailing meteorological conditions, write code in R and Python to estimate the coefficients of the following regression model and make an assessment of the predictive accuracy of the model.

$$area = \beta_1 temp + \beta_2 wind + \beta_3 rain + \beta_0 \tag{3}$$

More specifically:

- I) Using all observations in the file you downloaded, use 10-fold cross-validation during which the coefficients of the above regression model will be estimated using the method of Ordinary Least Squares (OLS) and additionally calculate the mean squared error (RMSE) of the prediction. Your program should display the mean squared error (RMSE).
- II) Estimate again the coefficients of the same regression model using the method of Ordinary Least Squares and 10-fold cross-validation, but this time do not use the entire dataset for training and evaluation, only those observations where the value of the dependent variable (variable 'area') is less than 3.2 hectares (i.e., area ¡ 3.2), indicating small fires. Display the mean squared error (RMSE) of the prediction.

What conclusion can you draw regarding the accuracy of the predictions if you compare the mean squared errors estimated in cases I) and II) above?

Subject 6

Present the machine learning method known as Gradient Boosted Regression Trees. The presentation of the method should come in the form of text with at least 1000 words. Specifically, the presentation you will make should address the following questions:

- 1) What kind of data can be used in this method ?
- 2) What issue does this approach seek to solve?
- 3) Describe the method in a short text.
- 4) Mention the cases where the method is used. To do this, search the internet and provide at least 3 publications from the field of economics that use this method, and briefly comment on them.

Subject 7

The aim of this subject is to familiarize oneself with clustering issues. Specifically, you will use the K-means algorithm (and a version tailored for categorical data) and Hierarchical clustering to address the following:

- I) In this question, you are asked to create a system belonging to the category of movie recommender systems using the R and Python languages.Recommender systems (or recommendation systems - link) are systems that suggest new products to users based either on their preferences or on products they have purchased in the past and liked. The goal of such recommendation systems is to provide personalized services to users. All modern e-commerce systems such as Amazon and eBay provide such recommendation systems.

  You are asked to write a program in R and Python that recommends new movies to a user based on the movie preferences of that particular user. For this purpose, you are provided with two data files along with the assignment: movies.csv and ratings.csv. The movies.csv file is a comma-separated values (csv) file that contains the titles of 9125 movies along with the categories to which each movie belongs (e.g., action, drama, horror, film-noir, etc.). Each line of the movies.csv file is in the format:

“movieId , “title , Action, Adventure, Animation, Children, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, IMAX, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western, (no genres listed)

which is interpreted as follows: the movie with ID(“movieId)” has the title (“title)” and belongs to one or more of the following categories: Action, Adventure, Animation, Children, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, IMAX, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western, (no genres listed). The variables indicating categories are called category variables and are binary variables (dummy variables) taking only the values 0 or 1, specifying the categories to which the movie belongs. The presence of a one (1) in a category variable means that the movie belongs to that category. A value of 0 means that the movie does not belong to that category. A movie may belong to more than one category. For example, a line in the movies.csv file may look like this:

2,Jumanji (1995),0,1,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0

This line of code is interpretted as ”The movie with ID 2 is titled ”Jumanji (1995)” and belongs to the categories Adventure, Children, and Fantasy, as the corresponding variables have a value of 1”. For example, it does not belong to the Western category because the corresponding variable has a value of 0. If there is a value of 1 in the ”(no genres listed)” variable, it means that the category of the respective movie is either unknown or cannot be determined from the other category variables. A movie can belong to multiple categories, as shown by the example of the movie ”Jumanji (1995)” above. Each movie has a unique ID (movieId) listed in the file.

The ratings.csv file contains movie ratings from 671 distinct users. Overall, it contains 100,004 ratings from these 671 users. Each line of the file has the format:

userId, movieId, rating, timestamp

This is interpreted as follows: the user with ID (userId) rated the movie with ID (movieId) with a score of (rating), and the rating was made at (timestamp). The value (timestamp) specifies the time at which the rating was made and refers to the date and time. Specifically, the (timestamp) value is an integer representing the number of seconds elapsed since midnight on January 1st, 1970 (also known as ”the Epoch”) until the moment the rating was made. For example, the following line from the ratings.csv file:

3, 296, 4.5, 1298862418

The above statement can be interpreted as follows: User with ID 3 rated the movie with ID 296 with a score of 4.5, and this rating was made at timestamp 1298862418 (which translates to 1298862418 seconds after midnight on January 1, 1970). The movie IDs in the ratings.csv file correspond to the movies listed in the movies.csv file. Therefore, we can see that the movie with ID 296 in the above line of the ratings.csv file refers to the movie ”Pulp Fiction (1994)” as indicated in the movies.csv file. Thus, user 3 rated ”Pulp Fiction” with a score of 4.5.

Ratings are given on a scale from 1 to 5, with 1 being the lowest rating and 5 being the highest. Half-ratings such as 2.5, 3.5, etc., can also be provided. A user can rate more than one movie, but a specific user can rate a particular movie only once. For the purpose of this task, you can ignore the timestamp (timestamp) in the ratings.csv file.

Your code, to be written in both R and Python, should appropriately process the movies.csv and ratings.csv files. Given a specific user ID (which you will provide in your program), it should display movies that match the preferences of that user and that the user has not yet seen.

While there are various approaches to building such a recommendation system for the movie problem, the movie recommendation system you will create will be based on the following approach: It will recommend to user X to watch those movies T that they have not seen and which are quite similar to movies they have seen and liked a lot.

The recommendation system you will implement in R and Python will make use of the K-means clustering algorithm.

Below is a description of such an algorithm in words, which you should implement in both R and Python programming languages.

- 1) Study the movies file (movies.csv).
- 2)Cluster the movies read from the movies.csv file based on their categories using the K-means algorithm. The goal of clustering with the K-means algorithm is to group movies that belong to the same categories (and therefore are similar to each other) into the same cluster. You will perform clustering using the K-means algorithm, taking into account only the variables indicating the movie categories from the data in the movies.csv file (i.e., the dummy variables such as Action, Adventure, Animation, Children, etc.). However, because the movie category variables do not take continuous values (they take binary values of 0 and 1, hence are called dummy variables), you cannot use distance-based clustering functions relying on Euclidean distance. Therefore, in both the R and Python environments, you should find and install appropriate libraries that allow you to run the K-means algorithm with a distance calculation method suitable for the data in the ratings.csv file. We refer you to the 'amap' library in R and the 'Kmodes' library in Python, which you should install on your own computer and provide you with the appropriate versions of the K-means algorithm for categorical data. Pay special attention to the usage manuals of these libraries to determine which function to use and with which arguments to execute it.To determine the exact number of clusters K in both the R and Python programs, use the Elbow method. Specifically, run the K-means clustering algorithm with all values of K (centers) from 2 to 100. For each value of K that you run the K-means algorithm (K=2, 3, 4, 5, . . . ,100), keep track of the value of the objective function that tells you how good the clustering was for that specific value of K (such as Sum of Squared Error, Average dispersion, etc. – you can choose the objective function). Graphically represent the values of K along with the value

of the chosen objective function and select the value of K that shows the greatest decrease in the objective function and continues with non-significant changes. Apply the Elbow method in this graph to determine the value of K (centers)."

- 3) Once you have determined the appropriate value of K (centers) with which you obtain the best value of the objective function, rerun the K-means algorithm with the selected value of K to obtain the final clusters of the data.
- 4) Read the ratings.csv file and find, for each movie in the movies.csv file, the average rating given to it by users. Consider that the movie ratings are quantitative variables and that the arithmetic mean can be calculated. If you prefer not to use the mean, you can calculate the mode for each movie instead.
- 5) Select a specific user, for example, the user with ID 198 (or any other). Find out which movies they have rated, and for each movie rated by user 198, determine the cluster to which it belongs, as obtained from step 3 above. Consequently, from the data in the ratings.csv file, isolate the ratings of user 198 and introduce a new variable named clusterId. For each movie rated by user 198, assign the value of the clusterId variable to the cluster to which that movie belongs, as determined in step 3.
- 6) Find the average rating (or the mode if you prefer) given by user 198 to the clusters to which the movies they have rated belong. Specifically, group the movies rated by user 198 based on the cluster to which each movie belongs, and for each group, find the average rating of the movies rated by user 198 that belong to that group. This way, you can gain an understanding of the user's opinion for each cluster.
- 7) Next, remove those clusters of user 198 that have a low average movie rating. Since 'low' is a relative concept, define a low rating as a rating value less than 3.5.
- 8) If there are no clusters for the user with an average movie rating equal to or greater than 3.5 (for example, if all cluster averages are less than 3.5), then no movie recommendations can be made for the user, and the message 'Sorry, no recommendations for you!' should be displayed.
- 9) If there are clusters with an average rating greater than or equal to 3.5 for user 198, then for each such cluster, find the top 2 movies with the highest ratings within that cluster, which user 198 has not seen. Display the titles of these movies as recommendations for user 198, based on their preferences. Specifically, display the message 'You may also like the following movies,' followed by the titles of the recommended movies.

Summarizing the requirements of this subject, you should deliver the following:

- a) Code written in R, which implements the above movie recommendation algorithm for a user. The R code should display the graph mentioned in point 2) and provide movie recommendations for a specific user, which should be inputtable by the user of the program.
- b) Code written in Python, which implements the above movie recommendation algorithm for a user. The Python code should display the graph mentioned in

point 2) and provide movie recommendations for a specific user, which should be inputtable by the user of the program.

- II) Write code in R and Python, which reads the data from the europe.csv file, provided with the assignment, and performs hierarchical clustering on the dataset of the europe.csv file. In the R environment, use the functions dist() and hclust() of R, while in Python, use the appropriate functions from the scikit-learn library, which you can find here:scikit-lear.org and you must have installed in your working environment.Both the program in R and the program in Python you create should display as output the dendrogram of clusters with labels being the name of each country.

- III) Study the publication found in the file paper-8iii.pdf and provide a summary. The summary must definitely mention the data analysis techniques presented as well as the reason why these techniques have been used. You may mention any other relevant aspect as you see fit.

Subject 8

The aim of this project is to familiarize with topics of association analysis using the arules package of the R tool. Because the arules package is not pre-installed in the R environment, it needs to be installed and used on your system. The user manual of the arules package can be found here: arules.pdf For the purposes of this project, download the 'Fertility Data Set' from the UCI Machine Learning Repository, following this link: Fertility-dataset . The page contains information regarding the interpretation of the values in the dataset. Additional details about the interpretation of the dataset can be found in the following publication: SemenFertilityPrediction.pdf The 'Fertility Data Set' includes data for assessing male fertility along with elements of their lifestyle. The questions this project aims to study are to discover factors of lifestyle that co-occur with changes in male fertility. Specifically, the following are requested:

- I) Remove from the 'Fertility Data Set' the columns related to the variables 'Age at the time of analysis' and 'Number of hours spent sitting per day ene-16'. The modified dataset without these two attributes will be referred to as the 'Modified Fertility Data Set'. Then, write code in R that applies the Apriori algorithm to all remaining attributes of the modified dataset with default values. What is the number of rules returned and what do the rules returned by this algorithm signify?

- II) For the modified 'Fertility Data Set', write code in R that applies the Apriori algorithm with a minimum support threshold of 0.02, confidence of 1, and a constraint that limits the right-hand side of the rules to only include 'Diagnosis=altered'. How many and which rules are returned?

- III) A rule Y is considered redundant when there exists a rule X with a greater or equal lift than Y, and additionally, Y is a super-rule of X. A rule has the general form $LHS => RHS$. Rule Y is a super-rule of X if $LHS(Y)LHS(X)$ and $RHS(Y) == RHS(X)$. For example, $A, B => C$ is a super-rule of $A => C$. The lift for each rule is provided by the Apriori algorithm. Sort the rules obtained from question II) by lift and then remove redundant rules. Provide the set of rules remaining after removing redundant rules.