CIS 6930 - Applied Machine Learning Using Python (Fall 2019)
Project 2 - Due Date: Dec. 8, 2019 Sunday 11:59 pm

This project is a two-fold programming assignment, where you will implement a machine learning system to solve a classification/regression problem and a clustering problem.

For the first part, you will use the NBA Rookie Stats dataset to predict if a player will last over 5 years or not. For the second task, you are given the Online Shoppers Intention dataset and you are to cluster these data to provide helpful insights. The details of these two datasets and the questions you will try to answer are as follows.

1 NBA Player Longevity Prediction

1.1 The Data

For this first part, you are given the NBA Rookie Stats dataset (provided at data.world). This dataset is accessible on our Canvas course site.

This dataset totals 21 columns and 1340 rows. The 21 features are play name (Name), games played (GP), minutes played (MIN), points per game (PPG), field goals made (FGM), field goal attempts (FGA), field goal percent (FG%), three points made (3PM), three point attempts (3PA), three point percent (3P%), free throws made (FTM), free throw attempts (FTA), free throw percent (FT%), offensive rebounds (OREB), defensive rebounds (DREB), rebounds (REB), assists (AST), steals (STL), blocks (BLK), turnovers (TOV), and target (TAR).

Each row in the table represents a player's *rookie statistics*, stats of that player's first season.

Out of these 21 attributes, the last attribute is the class attribute for which your system will predict about. It is a Boolean attribute, where “o” means the career length of the player is less than 5 years, and “1” greater than or equal to 5 years. The other 20 attributes are the features your models may consider. Out of these 20, there is 1 text attribute and 19 numerical attributes.

1.2 The Task

You are to explore the following classification/regression models to predict the target value and report the comparison of their performances based F_1 scores.

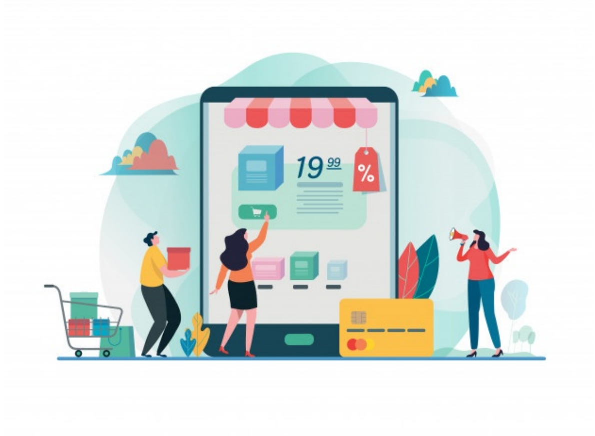
1. K-nearest neighbors [1]
2. Random forests [2]
3. Logistic regression [3]
4. Artificial neural networks [4]

1.3 The Questions

Here are the questions you need to address eventually in the project report.

1. When you prepare the data for training the models, did you discover any attribute to remove or any new attribute to add? If you did, discuss the choices.
2. Normalizing (a.k.a., scaling) features is desirable for distance-based models, e.g., k-nearest neighbors. Did you try feature normalization for some of the models? If so, talk about if any improvement.
3. Regularization is a common practice to battle overfitting. How is varying the penalty parameter in logistic regression affect the performance F_1 score on testing? (The logistic regression penalty parameter may be 'none', 'l1', 'l2' or 'elasticnet'.)
4. These models have hyperparameters. When training, experiment using GridSearch to select hyperparameters for your models. What are the best hyperparameters among those you tried?
5. Which model you experimented with gives the best F_1 score on testing?

2 Online Shopper Clustering



[1] sklearn.neighbors.KNeighborsClassifier

[2] sklearn.ensemble.RandomForestClassifier

[3] sklearn.linear_model.LogisticRegression

[4] sklearn.neural_network.MLPClassifier

2.1 The Data

For this second part, you are going to consider the Online Shoppers Intention dataset (provided at UCI Machine Learning Repository). This dataset too is accessible on our Canvas course site.

This dataset has 18 columns and 12,330 rows. The 18 attributes include 8 categorical and 10 numerical attributes. The last attribute "Revenue" is the class label: "0" means not ending up shopping, and "1" means ending up shopping. The meaning of the other attributes are the following.

"Administrative", "Administrative Duration", "Informational", "Informational Duration", "Product Related" and "Product Related Duration" represent the number of different types of pages visited by the visitor in that session and total time spent in each of these page categories. The values of these features are derived from the URL information of the pages visited by the user and updated in real time when a user takes an action, e.g. moving from one page to another.

The "Bounce Rate", "Exit Rate" and "Page Value" features represent the metrics measured by "Google Analytics" for each page in the e-commerce site. The value of "Bounce Rate" feature for a web page refers to the percentage of visitors who enter the site from that page and then leave ("bounce") without triggering any other requests to the analytics server during that session.

The value of "Exit Rate" feature for a specific web page is calculated as for all page views to the page, the percentage that were the last in the session. The "Page Value" feature represents the average value for a web page that a user visited before completing an e-commerce transaction.

The "Special Day" feature indicates the closeness of the site visiting time to a specific special day (e.g. Mothers Day, Valentine's Day) in which the sessions are more likely to be finalized with transaction. The value of this attribute is determined by considering the dynamics of e-commerce such as the duration between the order date and delivery date. For example, for Valentins day, this value takes a nonzero value between February 2 and February 12, zero before and after this date unless it is close to another special day, and its maximum value of 1 on February 8. The dataset also includes operating system, browser, region, traffic type, visitor type as returning or new visitor, a Boolean value indicating whether the date of the visit is weekend, and month of the year.

2.2 The Task

You are to explore the following clustering models with $k = 4$ to provide insight of the dataset and report the comparison of their performances.

1. K-means [5]
2. Complete-Linkage Agglomerative nesting [6]

When clustering, do NOT consider the last attribute "Revenue."

[5] sklearn.cluster.KMeans

[6] sklearn.cluster.AgglomerativeClustering

2.3 Performance Measures

Let us take the last attribute "Revenue" as the label and let us denote by $\mathcal{C} = \{C_1, C_2\}$ the two clusters it gives. Similarly, we denote by $\mathcal{C}^* = \{C_1^*, \ldots, C_4^*\}$ the clusters the model generates. Let us define $\lambda_i \in \{1, 2\}$ to be the cluster label of example $\boldsymbol{x}_i$ in clustering $\mathcal{C}$, and $\lambda_i^* \in \{1, 2, 3, 4\}$ cluster label of $\boldsymbol{x}_i$ in clustering $\mathcal{C}^*$.

We now define two sets S and D:

- $S = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) | i < j, \lambda_i = \lambda_j, \lambda_i^* = \lambda_j^*\}$
- $D = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) | i < j, \lambda_i \neq \lambda_j, \lambda_i^* \neq \lambda_j^*\}$

Intuitively, S is the set of all example pairs that are labeled the same in $\mathcal{C}$ and that are put in the same cluster in $\mathcal{C}^*$, and D is the set of all example pairs that are labeled differently in $\mathcal{C}$ and that are put in different clusters in $\mathcal{C}^*$.

The **Rand Index** (RI) takes S and D and computes $RI = \frac{2(|S|+|D|)}{m(m-1)}$, where m is the number of examples. RI is a value in the unit interval $[0, 1]$, the bigger the better the model. Thereafter, you will use RI to compare k-means and agglomerative nesting.

If we do not consider "Revenue" at all, we may use Davies-Bouldin Index (DBI) or Dunn Index (DI) from our lecture. In fact, you will just k-means and agglomerative nesting using DBI too.

2.4 The Questions

Here are the questions you need to address eventually in the project report.

1. When you prepare the data for training the models, how did you deal with the categorical attributes, for the clustering models we examine here only takes numerical features?
2. Which model is better considering their RI scores?
3. Which model is better considering their DBI scores?

3 Requirements

1. Your project should be in Python 3 and you are free to use any Python package to help you develop your programs.
2. For the first part, when separate the dataset for training and testing, use 80% randomly selected for training and the rest for testing. During training, use 10-fold cross validation to pick the best model learned. Finally, it will be tested on the testing set.
3. For the second part, there will not be training or testing, and use the whole dataset for clustering.

4 Deliverables

Zip the following to [your-last-name]_Project2.zip and submit to Canvas.

1. A directory that contains all your Python programs that are may be .py or .ipynb.
2. A README file that contains instructions to run your Python programs.
3. A PDF report that describes your results on these two parts. In particular, it must address the aforementioned questions.