Northeastern University, College of Computer and Information Science

CS 6220 Data Mining — Assignment 6

Due: by end of day 18th Feb 2018 (100 points)

Regression

This assignment will require you to implement and interpret some of the regression concepts that were introduced in this module. Keep in mind that the main objective of this assignment is to highlight the insights that we can derive from applying these techniquesthe coding aspect is secondary. Accordingly, you are welcome to consult any online documentation and/or code that has been posted to the course website, so long as all references and sources are properly cited. You are also encouraged to use code libraries, so long as you acknowledge any source code that was not written by you by mentioning the original author(s) directly in your source code (comment or header).
Using the well-known Iris Flower dataset, you will be asked to construct a linear regression experiment that reconstructs the values for a given feature based on the values for the remaining ones. You will then be asked to evaluate the performance of each different combination and to provide visual outputs.

Objectives:

1. Apply linear regression to a dataset containing numerical features and evaluate it using R-squared metrics

2. Apply logistic regression to a dataset of mixed data types to generate predictions and evaluate it using classification accuracy

Submission: Submit your ipynb using the Blackboard submission portal as before.
Grading Criteria:
Follow the instructions in the pdf, and complete each task. You will be graded on the application of the modules topics, the completeness of your answers to the questions in the assignment notebook, and the clarity of your writing and code.

Regression

The Data

For this portion of the assignment you will be using the Iris Flower dataset, available here. The dataset consists of 50 samples from each of three species of Iris (*Iris setosa*, *Iris virginica* and *Iris versicolor*). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters.

The Idea: Using Linear Regression on the Iris Dataset

Often, we may want to predict one feature based upon other features. Your objective here will be to generate a linear model of one of the features (a continuous variable) in the Iris dataset using one or more of the remaining features and/or class values. In our case, were interested in finding the best linear model among those that can be generated from this dataset.

What to Do

First, load the Iris dataset. This can be done using the following snippet of code:

```
import pandas as pd
fileURL = 'http://archive.ics.uci.edu/ml/
          machine-learning-databases/iris/iris.data'
iris = pd.read_csv(fileURL, names=['Sepal Length', 'Sepal Width',
                                   'Petal Length', 'Petal Width',
                                   'Species'], header=None)
iris = iris.dropna()
```

Next, you can visualize the correlation between different features using the following snippet of code, which executes the provided pairs function:

```
pairs(iris)
```

Some pairs of features tend to be more correlated than others. Try to uncover related features by using linear regression to model the relationship between pairs of features. In other words, use one feature as a target (dependent) variable and another feature as a predictor (independent) variable. To generate a linear regression model, you may use the *linear_model.LinearRegression*() function available via the scikit-learn library. To run the model on the Iris data, first divide the dataset into training and testing sets, then fit the model on the training set and predict with the fitted model on the testing set. scikit-learn provides several functions for dividing datasets in this manner, including *cross_validation.KFold* and *cross_validation.train_test_split*.
Several statistics can be generated from a linear model. Given a fitted linear model, the following code outputs the model coefficients (the parameter values for the fitted model), the residual sum of squares (the model error), and the explained variance (the degree to which the model explains the variation present in the data):

```
# The coefficients
print('Coefficients:', regr.coef_)
# The mean squared error
print("Mean squared error: %.2f" %
    np.mean((coly_pred - coly_test) ** 2))
# Explained variance score: 1 is perfect prediction
print('Variance score: %.2f' % regr.score(colx_test, coly_test))
```

You can use these scores to measure the efficacy of a particular linear model.

What to Provide

Your output should contain the following:

- A scatterplot matrix of scatterplots, with one scatterplot for each pairwise combination of features.
- A plot of the linear regression models generated on each pairwise combination of features, with corresponding model statistics.
- A plot of the best overall linear regression model you were able to generate using any combination of features (including the use of multiple features used in combination to predict a single feature), with corresponding model statistics.

Given this output, respond to the following questions:

1. Based upon the linear models you generated, which pair of features appear to be most predictive for one another? Note that you can answer this question based upon the output provided for the linear models.
2. Suppose you tried to generate a classification model on this dataset, but only after removing the feature that you were best able to predict based upon other features. How would removing this feature affect the classification performance?