****

**MACHINE LEARNING FOUNDATION 417**

**REPORT**

On

**AUTOMATING INDOOR WEATHER ADJUSTMENT**

Submitted by

**Aman Kumar Soni**

**Registration No: 11715307**

**Roll No: 62**

**Programme Name: Computer Science (P132)**

**Under the Guidance of**

Sanjay Kumar Singh

**School of Computer Science & Engineering**

**Lovely Professional University, Phagwara**

**(Feb-April, 2020)**

**Introduction:**

The Automating indoor weather adjustment was chosen for this project as it is an ideal candidate for classification tools like Linear Regression and Decision tree. The data set is an experimental data which describes binary classification (room classification) from Temperature, Humidity, Light and CO2. Ground-truth occupancy was obtained from time stamped pictures that were taken every minute. The project would be about measuring the performance of each classification tool, and compare the tools with respect to the data set for accuracy.

**Libraries Used:**

**Pandas:**

Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.

**Numpy:**

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data.  
Arbitrary data-types can be defined using Numpy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases

**Matplotlib:**

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.

One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

**Seaborn:**

Seaborn is a library for making statistical graphics in Python. It is built on top of matplotlib and closely integrated with pandas data structures. Here is some of the functionality that seaborn offers: A dataset-oriented API for examining relationships between multiple variables.

**Machine Learning Libraries that are used in Project:**

**Scikit-learn:** [scikit-learn](http://scikit-learn.org/stable/) is an open source Python library that implements a range of machine learning, pre-processing, cross-validation and visualization algorithms using a unified interface.

**Important features of scikit-learn:**

* Simple and efficient tools for data mining and data analysis. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means, etc.
* Accessible to everybody and reusable in various contexts.
* Built on the top of NumPy, SciPy, and matplotlib.
* Open source, commercially usable – BSD license.

1. **from sklearn import metrics:** In multilabel classification, this function computes subset accuracy: the set of labels predicted for a sample must exactly match the corresponding set of labels in y\_true.
2. **from sklearn.metrics import roc\_curve, auc:** Compute Receiver operating characteristic (ROC)
3. **from sklearn.metrics import accuracy\_score, confusion\_matrix,classification\_report:** Build a text report showing the main classification metrics
4. **from sklearn.preprocessing import StandardScaler:** Standardization of a dataset is a common requirement for many machine learning estimators: they might behave badly if the individual features do not more or less look like standard normally distributed data
5. **from matplotlib.colors import ListedColormap: ListedColormap**. Colormap object generated from a list of **colors**. This may be most useful when indexing directly into a colormap
6. **from matplotlib import style:**  the different available style sheets on a common set of example plots: scatter plot, image, bar graph, patches, line plot and histogram,
7. **#from sklearn.cross\_validation import train\_test\_split:** Split arrays or matrices into random train and test subsets
8. **from sklearn.tree import export\_graphviz:** This function generates a GraphViz representation of the decision tree, which is then written into out\_file. Once exported, graphical renderings can be generated using, for example:
9. **from sklearn.tree import DecisionTreeClassifier:** **Decision Trees (DTs)** are a non-parametric supervised learning method used for [classification](https://scikit-learn.org/stable/modules/tree.html#tree-classification) and [regression](https://scikit-learn.org/stable/modules/tree.html#tree-regression). The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.
10. **from sklearn.ensemble import RandomForestClassifier:** A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.
11. **from sklearn.svm import SVC:** **Support vector machines (SVMs)** are a set of supervised learning methods used for [classification](https://scikit-learn.org/stable/modules/svm.html#svm-classification), [regression](https://scikit-learn.org/stable/modules/svm.html#svm-regression) and [outliers detection](https://scikit-learn.org/stable/modules/svm.html#svm-outlier-detection).
12. **from sklearn.neural\_network import MLPClassifier:** Multi-layer Perceptron classifier. This model optimizes the log-loss function using LBFGS or stochastic gradient descent.

**Dataset:**

[https://archive.ics.uci.edu/ml/datasets/Occupancy+Detection+#](https://archive.ics.uci.edu/ml/datasets/Occupancy+Detection+%23)

https://ianlondon.github.io/blog/encoding-cyclical-features-24hour-time/

|  |  |
| --- | --- |
| **Abstract**: Experimental data used for binary classification (room occupancy) from Temperature,Humidity,Light and CO2. Ground-truth occupancy was obtained from time stamped pictures that were taken every minute. |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data Set Characteristics:** | Multivariate, Time-Series | **Number of Instances:** | 20560 | **Area:** | Computer |
| **Attribute Characteristics:** | Real | **Number of Attributes:** | 7 | **Date Donated** | 2016-02-29 |
| **Associated Tasks:** | Classification | **Missing Values?** | N/A | **Number of Web Hits:** | 77369 |

**Source:**

Luis Candanedo, luismiguel.candanedoibarra **'@'** umons.ac.be, UMONS.

**Data Set Information:**

Three data sets are submitted, for training and testing. Ground-truth occupancy was obtained from time stamped pictures that were taken every minute.   
For the journal publication, the processing R scripts can be found in:   
[[Web Link]](https://github.com/LuisM78/Occupancy-detection-data)

**Attribute Information:**

date time year-month-day hour:minute:second   
Temperature, in Celsius   
Relative Humidity, %   
Light, in Lux   
CO2, in ppm   
Humidity Ratio, Derived quantity from temperature and relative humidity, in kgwater-vapor/kg-air   
Occupancy, 0 or 1, 0 for not occupied, 1 for occupied status

**Relevant Papers:**

Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. Luis M. Candanedo, VÃ©ronique Feldheim. Energy and Buildings. Volume 112, 15 January 2016, Pages 28-39.

**Citation Request:**

Please cite the following publication:   
Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. Luis M. Candanedo, VÃ©ronique Feldheim. Energy and Buildings. Volume 112, 15 January 2016, Pages 28-39.

**Code Implementation file name:**

“Project.ipynb”

**Github link:**

<https://github.com/computerwala/Automating-indoor-weather-adjustments>