**IN3062**

Introduction to AI

Contents

1. Describe a machine learning problem and apply artificial intelligence techniques to that problem.
2. Describe the systematic application of your chosen artificial intelligence methodology to the chosen problem (for example, data preparation, parameter tuning).
3. Apply, compare, contrast and critically evaluate at least two ways of analysing your problem data.

To be answered:

What is your dataset, problem domain?  
Is your model classification or regression?   
Did you have any missing, corrupt or misleading data? If so, how did you cope it?   
Have you omitted some data? If so, why?   
Did you apply techniques to understand your dataset?   
What models did you use?   
How did you encode the input variables?   
What are the criteria for selecting model performance evaluation tools?   
What were your outputs?   
Did you have any problems or difficulties working with the dataset?

Steps to process data:

1. Import data
2. Clean Data
3. Split Data into training and testing data
4. Create model
5. Train model
6. Make predictions
7. Evaluate and improve

Numpy

Pandas – dataFrame

MatPlotLib – graphs

Scikit-Learn

8/12 : learning ML with <https://www.youtube.com/watch?v=7eh4d6sabA0> , <https://www.youtube.com/watch?v=VwVg9jCtqaU>

Looking for deepforest dataset <https://deepforest.readthedocs.io/en/latest/getting_started.html>

Sat4\_dataset: <https://www.kaggle.com/crawford/deepsat-sat4?select=sat4annotations.csv>

To be studied: <https://www.youtube.com/results?search_query=python+keras>

23/12: SAT-4 dataset seems to be too hard to grasp.

What is your dataset, problem domain?  
<https://www.kaggle.com/uciml/forest-cover-type-dataset> - this dataset seems fun.

Question ask is. What type of tree cover will be on certain combination of wildlife and soil?

Is your model classification or regression?  
This will be regression problem.

Did you have any missing, corrupt or misleading data? If so, how did you cope it?  
I am going to use some analytic’s methotds to find out more about this dataset.

IN3062 Coursework : Tree coverage prediction  
Jakub Zavacky  
Department of computer Science, City, University of London

This document is intended to demonstrate understanding of artificial intelligence methods and data preparation. This coursework is based on learnings from Introduction to artificial intelligence module. Dataset was chosen for it’s complexity and personal preference.

**I. Introduction**This paper contains my general understanding of artificial intelligence methods. The dataset chosen is Tree coverage dataset to be found on Kaggle website on the following link <https://www.kaggle.com/uciml/forest-cover-type-dataset>   
The main focus in this work is to predict what type of tree is present based on elevation, soil type and many more aspects. This work is written in python and will use following artificial intelligence and data manipulation libraries: TensorFlow, Numpy, Pandas, sklearn

This work is based on learning gained from lectures in module IN6062, and following sources. The youtube channel “Programming with Mosh”, especially his video <https://www.youtube.com/watch?v=7eh4d6sabA0> , the deeplizard website and their tensorFlow tutorials, particularly <https://deeplizard.com/learn/playlist/PLZbbT5o_s2xrwRnXk_yCPtnqqo4_u2YGL>

**II. Description of Dataset**The study area includes four wilderness areas located in the Roosevelt National Forest of northern Colorado. Each observation is a 30m x 30m patch. You are asked to predict an integer classification for the forest cover type. The seven types are:

1 - Spruce/Fir  
2 - Lodgepole Pine  
3 - Ponderosa Pine  
4 - Cottonwood/Willow  
5 – Aspen  
6 - Douglas-fir  
7 - Krummhol

The dataset set (cca 500 000 observations) contains both features and the Cover\_Type.   
Data Fields:  
Elevation - Elevation in meters  
Aspect - Aspect in degrees azimuth  
Slope - Slope in degrees  
Horizontal\_Distance\_To\_Hydrology - Horz Dist to nearest surface water features  
Vertical\_Distance\_To\_Hydrology - Vert Dist to nearest surface water features  
Horizontal\_Distance\_To\_Roadways - Horz Dist to nearest roadway  
Hillshade\_9am (0 to 255 index) - Hillshade index at 9am, summer solstice  
Hillshade\_Noon (0 to 255 index) - Hillshade index at noon, summer solstice  
Hillshade\_3pm (0 to 255 index) - Hillshade index at 3pm, summer solstice  
Horizontal\_Distance\_To\_Fire\_Points - Horz Dist to nearest wildfire ignition points  
Wilderness\_Area (4 binary columns, 0 = absence or 1 = presence) - Wilderness area designation  
Soil\_Type (40 binary columns, 0 = absence or 1 = presence) - Soil Type designation  
Cover\_Type (7 types, integers 1 to 7) - Forest Cover Type designation  
Wilderness\_areas and Soil\_type descriptions are found in Appendix A.

**III. Method  
Approach to this challenge is going to be by supervised learning. The output is regression as it predicts values in between**   
***Import data***  
Data has been imported by Pandas’ read\_csv function. In case we would have really big dataset we can use chunksize parameter to load part of csv file. Some datasets I was training on also came with separated csv files for training and test data  
***Cleaning the Data***  
Data has been checked if they have any empty fields with dataset.isnull().any(), and as it returned False for all columns, there is no need to substitute any values or remove any rows in the dataset. If that would be case, I would have to remove the rows with NaN values in Soild or wilderness columns, if the NaN values would be in the other columns I could substitute them with mean value in regards to wilderness area the data are missing at.  
***Split Data into training and testing data***  
I have separated Covet\_Type column from dataset to create features dataframe and label dataframe. Those dataframes I fed to sklearn library that has convenient function train\_test\_split, that splits data in training samples and testing samples. I am using random\_state=0 parameter to reproduce the same data samples so I can reasonably compare my models with other methods in the future.  
***Create model***   
I am using decision tree model and linear regression model from sklearn library. Creating model is fairly simple with sklearn library – you are just creating object.  
***Train model***  
Training model is again fairly simple as we just call .fit method on our model object and provide corresponding training data and training labels.  
***Make predictions***  
Predictions are based on testing data. This data has no input value and are new to our model. Model makes prediction based on what we showed him during training and classify the test data.  
***Evaluate and improve***  
Decision tree is evaluated as a accuracy score based on how many test data he recognised correctly. For linear regression I use mean error and square root error to classify model correctness. We can observe and try to improve result by adjusting ratio between training and testing data. We can also compare different models and decide which gives best results.

**IV. Experiment  
I have applied**

APENDIX A  
The wilderness areas are:  
1 - Rawah Wilderness Area  
2 - Neota Wilderness Area  
3 - Comanche Peak Wilderness Area  
4 - Cache la Poudre Wilderness Area  
The soil types are:  
1 Cathedral family - Rock outcrop complex, extremely stony.  
2 Vanet - Ratake families complex, very stony.  
3 Haploborolis - Rock outcrop complex, rubbly.  
4 Ratake family - Rock outcrop complex, rubbly.  
5 Vanet family - Rock outcrop complex complex, rubbly.  
6 Vanet - Wetmore families - Rock outcrop complex, stony.  
7 Gothic family.  
8 Supervisor - Limber families complex.  
9 Troutville family, very stony.  
10 Bullwark - Catamount families - Rock outcrop complex, rubbly.  
11 Bullwark - Catamount families - Rock land complex, rubbly.  
12 Legault family - Rock land complex, stony.  
13 Catamount family - Rock land - Bullwark family complex, rubbly.  
14 Pachic Argiborolis - Aquolis complex.  
15 unspecified in the USFS Soil and ELU Survey.  
16 Cryaquolis - Cryoborolis complex.  
17 Gateview family - Cryaquolis complex.  
18 Rogert family, very stony.  
19 Typic Cryaquolis - Borohemists complex.  
20 Typic Cryaquepts - Typic Cryaquolls complex.  
21 Typic Cryaquolls - Leighcan family, till substratum complex.  
22 Leighcan family, till substratum, extremely bouldery.  
23 Leighcan family, till substratum - Typic Cryaquolls complex.  
24 Leighcan family, extremely stony.  
25 Leighcan family, warm, extremely stony.  
26 Granile - Catamount families complex, very stony.  
27 Leighcan family, warm - Rock outcrop complex, extremely stony.  
28 Leighcan family - Rock outcrop complex, extremely stony.  
29 Como - Legault families complex, extremely stony.  
30 Como family - Rock land - Legault family complex, extremely stony.  
31 Leighcan - Catamount families complex, extremely stony.  
32 Catamount family - Rock outcrop - Leighcan family complex, extremely stony.  
33 Leighcan - Catamount families - Rock outcrop complex, extremely stony.  
34 Cryorthents - Rock land complex, extremely stony.  
35 Cryumbrepts - Rock outcrop - Cryaquepts complex.  
36 Bross family - Rock land - Cryumbrepts complex, extremely stony.  
37 Rock outcrop - Cryumbrepts - Cryorthents complex, extremely stony.  
38 Leighcan - Moran families - Cryaquolls complex, extremely stony.  
39 Moran family - Cryorthents - Leighcan family complex, extremely stony.  
40 Moran family - Cryorthents - Rock land complex, extremely stony.