CSE 537 Assignment 5 – Report

Decision tree classifier

Atanu Ghosh(110280569)

Bhargava Mourya(110308227)

1. Decision Tree

How to run:

```
python decision_tree.py <path_to_input_file>
```

Given data is a mix of discrete and continuous attributes. Idea is to pick an attribute which gives more information gain among all the attributes and make it as a root of the tree. Definition of children of this root depends on whether the root attribute is discrete or continuous. In case of discrete, root contains as many children as the size of its domain. In case of continuous attribute, we sort the range of data, pick a value, split the data into two halves and find informational gain achieved with this split. All possible consecutive splits are taken into consideration.

After the root node is calculated, the function is recursively called to calculate the rest of the nodes and eventually the tree. The tree is stored as a dict of dicts in python. Currently all the values of training set is taken into consideration. The data set was split in 80:20 ratio of training to test set. To avoid overfitting at some point we consider to strop expanding the tree, where the ratio of positive to negative or vice versa results is greater than a certain threshold like 90%

A sample output tree obtained is shown below:

```
{
    8: {
        9: {
            5: {
                0: {
                    '+': '+'
                },
                3: {
                    0: {
                        '+': '+'
                    },
                    11: {
                        0: {
                            '+': '+'
                        }
                    },
                    6: {
                        0: {
                            1: {
                                2: {
                                    '+': '+'
                                }
                            },
                            '-': '-'
                }
        }
},
11: {
        0: {
                '+': '+'
        },
        3: {
                0: {
                        '+': '+'
                },
                12: {
                        0: {
                                1: {
                                        2: {
                                                '-': '-'
                                        }
                                },
                                '-': '-'
                        }
                }
        },
        6: {
                0: {
                        '-': '-'
                }
        }
},
6: {
        0: {
                1: {
                        '-': '-'
                },
                '+': '+',
                11: {
                        1: {
                                '-': '-'
                        }
                }
        },
        3: {
                0: {
                        '-': '-'
                },
                11: {
                    0: {
                        '-': '-'
                    }
                }
            }
        }
    }
},
5: {
    0: {
        1: {
            '-': '-'
        },
        '-': '-'
    },
    3: {
        0: {
            '-': '-'
        }
    },
    6: {
        0: {
            '+': '+'
        },
        11: {
            0: {
                '-': '-'
            },
            3: {
                0: {
                    '-': '-'
                },
                12: {
                    0: {
                        1: {
                            2: {
                                '+': '+'
                            }
                        },
                        '-': '-'
                    }
                }
            }
        }
    },
9: {
    0: {
        '-': '-'
    },
    11: {
        0: {
            '-': '-'
        },
        3: {
            0: {
                1: {
                    2: {
                        '+': '+'
                    }
                },
                '-': '-'
            }
        }
    },
    12: {
        0: {
            '-': '-'
        },
        6: {
            0: {
                1: {
                    '-': '-'
                },
                3: {
                    11: {
                        1: {
                            2: {
                                '-': '-'
                            },
                            '-': '-'
                        }
                    }
                },
                '-': '-'
            }
        }
    }
},
12: {
    0: {
```

```
                    '-': '-'
                }
            }
        }
    }
}
```

Avoiding Overfitting, Tree Pruning:

To avoid overfitting, we terminate the tree when the ratio of positive to negative values is greater than 0.8. This is one method which we employed and got the following tree as a result. A much shorter tree.

```
{
    8: {
        9: {
            5: {
                0: {
                    '+': '+'
                },
                3: {
                    0: {
                        '+': '+'
                    }
                },
                6: {
                    0: {
                        '-': '-'
                    },
                    11: {
                        0: {
                            '-': '-'
                        }
                    },
                    3: {
                        0: {
                            '+': '+'
                        },
                        11: {
                            0: {
                                '+': '+'
                            }
                        }
                    }
                },
                '+': '+',
11: {
        0: {
                '+': '+'
        },
        3: {
                0: {
                        '+': '+'
                },
                12: {
                        0: {
                                '-': '-'
                        },
                        6: {
                                0: {
                                        1: {
                                                2: {
                                                        '+': '+'
                                                }
                                        },
                                        '+': '+'
                                }
                        }
                }
        },
        6: {
                0: {
                        '-': '-'
                },
                3: {
                        0: {
                                '+': '+'
                        }
                },
                12: {
                        0: {
                                1: {
                                        '-': '-'
                                },
                                '+': '+'
                        }
                }
        }
},
12: {
        0: {
```

```
                        '+': '+'
                    }
                }
            }
        }, 5: {
            '-': '-'
        }
    }
}
```

The accuracy after splitting into 80:20 ratio of training to test in both the above cases ranges from **76% to 85%**

Handling ‘?’:

In testing the data with the already built decision tree classifier, we might encounter a ‘?’ for some attribute which is to be tested. In this case, we consider all possible values the attribute can take and iteratively test the data for each possible value. In each case, we keep track if it is classified as ‘+’ or ‘-‘ and their respective counts. If count of ‘+’ is greater than or equal to count of ‘-‘, the given data is classified as ‘+’ else ‘-‘.

Results: (after running 10 iterations)

```
Accuracy:  78.587196468 %
Accuracy:  83.3759590793 %
Accuracy:  77.3684210526 %
Accuracy:  78.5876993166 %
Accuracy:  77.8290993072 %
Accuracy:  80.9523809524 %
Accuracy:  84.2857142857 %
Accuracy:  80.3867403315 %
Accuracy:  77.5641025641 %
Accuracy:  78.9173789174 %
```