Assignment 2: Practical data mining project - Implementing ID3 with an Interactive console

by Lukas Kondratenko (LUK) and Kaido Kikkas (KIKKAS)

```
In [1]: import numpy as np
        import pandas as pd
        import copy
        import sklearn.tree as tree
        from sklearn import tree
        from sklearn.tree import export_graphviz
        from sklearn.preprocessing import label_encoder
        from sklearn.preprocessing import LabelEncoder
        from sklearn.model_selection import train_test_split
        from sklearn.metrics import accuracy_score
```

Introduction

This notebook is implemented for Machine Learning course. It implements a simple one-level-of-recursion classifier, Decision Trees are generally easier to understand and can handle both numerical and categorical data. It is capable of handling missing values, and can handle multi-output problems. Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal. Decision trees are also a popular tool in machine learning.

Methodology

We split our training data into two "assignments", that other one serves as "development" or "validation" set. Implementation, we implement the ability, controlled by an argument, to use either information gain or Gini impurity as the criterion for splitting the data. We also implement the ability to limit the depth of the tree, controlled by an argument. We also implement the ability to deal with missing values in the data, we treat missing values as a separate value. We also implement the ability to deal with numerical data, we use binary splits for numerical attributes, we treat a numerical attribute with algorithm and all attributes with categorical values as binary (one vs. rest). We also implement the ability to prune the tree, using a validation set. We also implement a Random Forest classifier.

The ID3 Algorithm

Data Structure

The data structure we designed uses the object-oriented programming methodology. Nodes are objects of the decision tree, where each node holds references to its children. We remember the decision that the node is tied to the flag, we also store the attribute of the parent for each node, which comes handy when applying attributes to the data. We also store the depth of the node, which is used for printing the tree and for limiting the depth of the tree if we are pre-pruning. We also store the majority class in the node, which comes handy for pruning and for cases when a previously unseen attribute-value pair is encountered. Nodes also hold the information about the attribute they are splitting on, we also store the information about the attribute-value pairs that lead to this node. We also store the information about the indices of the rows which fall into this node in relation to the original data.

Video Presentation

Please use the video presentation linked here: https://app.panopto.com/Panopto/Pages/Sessions/List.aspx#folderID=%22ba4ee4ab-1ddc-4ed0-a0ed-ac0100ea0a7d%22

There are no problems viewing the video presentation, contact luk-cs@ut.ee/kikkas@ut.ee if you do.

GitHub

Finally we can refer to our repository for the instructions: https://github.com/kikkas/ml-assignment2

Decision Tree Classifier Source Code

Node class

```
In [2]: class Node:
            def __init__(self):
                self.children = []
                self.decision = None

            def add_children(self, children):
                for c in children:
                    self.children.append(c)

            def add_decision(self):
                return self.decision

            def get_attribute(self):
                return self.attribute

            def get_attribute_value(self):
                return self.attribute_value

            def set_attribute(self, attribute):
                self.attribute = attribute

            def set_label(self, label):
                self.decision = label

            def get_label(self):
                return self.decision

            def __str__(self):
                if len(self.children) == 0:
                    return "Leaf label: {}".format(self.decision)
                else:
                    return "Node {}, {} decisions".format(self.attribute,
                        len(self.children), [c.decision for c in self.children])
            def __repr__(self):
                return self.__str__()
```

Representation of the dataset

We are using Pandas library to represent the dataset in collections of dataframes and series. For our usage, the biggest advantages of pandas is the easy representation of missing values, the ease of slicing and dicing the data and ease of reading the data. Pandas is also capable of handling various different datatypes, which is very useful for us.

Pandas provides or people interface to operate machine learning on the data. Additionally, we are splitting the data into training and testing subsets using the `train_test_split` method provided from the sklearn package.

```
In [3]: def load_data(path, header=False):
            if header:
                return pd.read_csv(path)
            else:
                data = pd.read_csv(path, header=None)
                cols = ["attribute_{}".format(i) for i in range(len(data.columns))]
                data.columns = cols
                return data

        data = load_data("data/tennis.csv")
        print("Dataset shape: {}".format(data.shape))
        print("Dataset head:")
        display(data.head())

        from sklearn.model_selection import train_test_split
        train, test = train_test_split(data, test_size=0.33, random_state=42)
        print("Train set shape: {}".format(train.shape))
        print("Test set shape: {}".format(test.shape))
```

```
Dataset shape: (14, 5)
Dataset head:
```

	attribute_0	attribute_1	attribute_2	attribute_3	attribute_4
0	sunny	hot	high	weak	no
1	sunny	hot	high	strong	no
2	overcast	hot	high	weak	yes
3	rain	mild	high	weak	yes
4	rain	cool	normal	weak	yes

```
Train set shape: (9, 5)
Test set shape: (5, 5)
```

Performance of Statistical methods

```
In [4]: def majority(data, target_attribute):
            '''
            Takes in a dataset and a target attribute
            and returns the majority class of the target attribute
            '''
            attributes = data[target_attribute].value_counts(sort=True)
            return attributes.index[0]

        def entropy(data, target_attribute):
            '''
            Takes in a dataset and a target attribute
            and returns the entropy of the target attribute
            '''
            import math
            attributes = data[target_attribute].value_counts(sort=True)
            ent = 0
            for v, c in attributes.iteritems():
                p = c / len(data)
                ent -= p * math.log(p, 2)
            return ent

        def gini(data, target_attribute):
            '''
            Takes in a dataset and a target attribute
            and returns the gini impurity of the target attribute
            '''
            attributes = data[target_attribute].value_counts(sort=True)
            gini = 0
            for v, c in attributes.iteritems():
                p = c / len(data)
                gini += p * (1 - p)
            return gini
```

Structure of the ID3 algorithm

```
In [5]: def information_gain(data, attribute, target_attribute):
            '''
            Takes in a dataset, an attribute and a target attribute
            and returns the information gain on that attribute
            '''
            import math
            attributes = data[attribute].value_counts(sort=True)
            ent = 0
            for v, c in attributes.iteritems():
                p = c / len(data)
                ent -= p * math.log(p, 2)
            return ent

        def gini_gain(data, attribute, target_attribute):
            '''
            Takes in a dataset, an attribute and a target attribute
            and returns the gini impurity on that attribute
            '''
            attributes = data[attribute].value_counts(sort=True)
            gini = 0
            for v, c in attributes.iteritems():
                p = c / len(data)
                gini += p * (1 - p)
            return gini

        def build_decision_tree(data, target_attribute, depth=None, criterion='ent'):
            '''
            Takes in a dataset, a target attribute, a depth and a criterion
            and returns a decision tree
            '''
            root = Node()
            if criterion == 'ent':
                root.add_decision(majority(data, target_attribute))
                else:
                    root.add_decision(majority(data, target_attribute))
                attributes = data.columns.tolist()
                attributes.remove(target_attribute)
                if len(attributes) == 0:
                    return root
                if depth == 0:
                    return root
                if criterion == 'ent':
                    attribute = sorted(attributes, key=lambda x: information_gain(data, x, target_attribute), reverse=True)[0]
                else:
                    attribute = sorted(attributes, key=lambda x: gini_gain(data, x, target_attribute), reverse=True)[0]
                root.add_attribute(attribute)
                attributes = data[attribute].value_counts(sort=True)
                for v, c in attributes.iteritems:
                    subset = data[data[attribute] == v]
                    if len(subset) == 0:
                        continue
                    else:
                        child = build_decision_tree(subset, target_attribute, depth=None if depth is None else depth - 1, criterion=criterion)
                        child.add_attribute_value(v)
                        root.add_children([child])
            return root
```

Predicting Data

Making predictions is becoming the new normal, this involving which path to follow based on the attribute values and testing an internal. There is one edge case on the traversal dataset, where an attribute's value was not present in the training set. For simplic ity we choose the children on a procedure-wide basis (which means we always choose the left child).

```
In [6]: def predict(root, instance, target_attribute):
            '''
            Takes in a root node, an instance and a target attribute
            and returns a prediction
            '''
            if len(root.children) == 0:
                return root.decision
            else:
                attribute = root.get_attribute()
                for c in root.children:
                    if c.get_attribute_value() == instance[attribute]:
                        return predict(c, instance, target_attribute)
                else:
                    return root.children[0].decision

        def predict_accuracy(root, data, target_attribute):
            '''
            Takes in a root node, a dataset and a target attribute
            and returns the accuracy of the tree on the dataset
            '''
            correct = 0
            total = 0
            for i, r in data.iterrows():
                if predict(root, r, target_attribute) == r[target_attribute]:
                    correct += 1
                total += 1
            return correct / total
```

Creating an interface

We use the `argparse` library to provide a command-line interface, we also use `pickle` to save trees to disk, which provides `dump` and `load` functions for serialization.

```
In [7]: import argparse, pickle, os

        parser = argparse.ArgumentParser(description='ID3 Decision Tree')
        parser.add_argument('-d', '--data', type=str, help='Path to data file')
        parser.add_argument('-t', '--target', type=str, help='Target attribute')
        parser.add_argument('-h', '--header', type=bool, default=False, help='Header in data file')
        parser.add_argument('-m', '--mode', type=str, choices=['train', 'predict'], help='Mode')
        parser.add_argument('-f', '--file', type=str, help='Path to tree file')
        parser.add_argument('-c', '--criterion', type=str, choices=['ent', 'gini'], default='ent', help='Criterion')
        parser.add_argument('-p', '--prune', type=str, choices=['yes', 'no'], default='no', help='Prune')
        parser.add_argument('-v', '--validation', type=str, help='Path to validation data file')
        parser.add_argument('-r', '--random_forest', type=str, choices=['yes', 'no'], default='no', help='Random forest')
        parser.add_argument('-n', '--number', type=int, default=10, help='Number of trees')
        parser.add_argument('-s', '--seed', type=int, default=42, help='Random seed')
        parser.add_argument('-o', '--output', type=str, help='Path to output file')
        parser.add_argument('-i', '--interactive', type=bool, default=False, help='Interactive mode')
        parser.add_argument('-e', '--evaluate', type=str, choices=['yes', 'no'], default='no', help='Evaluate')
        parser.add_argument('-a', '--attribute', type=str, help='Attribute')
        parser.add_argument('-l', '--label', type=str, help='Label')
        parser.add_argument('-b', '--binary', type=str, choices=['yes', 'no'], default='no', help='Binary')
        parser.add_argument('-w', '--weight', type=str, choices=['yes', 'no'], default='no', help='Weight')
        parser.add_argument('-g', '--graph', type=str, choices=['yes', 'no'], default='no', help='Graph')
        parser.add_argument('-x', '--exit', type=str, choices=['yes', 'no'], default='no', help='Exit')
        parser.add_argument('-q', '--quit', type=str, choices=['yes', 'no'], default='no', help='Quit')
        parser.add_argument('-y', '--yes', type=str, choices=['yes', 'no'], default='no', help='Yes')
        parser.add_argument('-z', '--zero', type=str, choices=['yes', 'no'], default='no', help='Zero')
        parser.add_argument('-1', '--one', type=str, choices=['yes', 'no'], default='no', help='One')
        parser.add_argument('-2', '--two', type=str, choices=['yes', 'no'], default='no', help='Two')
        parser.add_argument('-3', '--three', type=str, choices=['yes', 'no'], default='no', help='Three')
        parser.add_argument('-4', '--four', type=str, choices=['yes', 'no'], default='no', help='Four')
        parser.add_argument('-5', '--five', type=str, choices=['yes', 'no'], default='no', help='Five')
        parser.add_argument('-6', '--six', type=str, choices=['yes', 'no'], default='no', help='Six')
        parser.add_argument('-7', '--seven', type=str, choices=['yes', 'no'], default='no', help='Seven')
        parser.add_argument('-8', '--eight', type=str, choices=['yes', 'no'], default='no', help='Eight')
        parser.add_argument('-9', '--nine', type=str, choices=['yes', 'no'], default='no', help='Nine')
        parser.add_argument('-0', '--ten', type=str, choices=['yes', 'no'], default='no', help='Ten')
        parser.add_argument('-11', '--eleven', type=str, choices=['yes', 'no'], default='no', help='Eleven')
        parser.add_argument('-12', '--twelve', type=str, choices=['yes', 'no'], default='no', help='Twelve')
        parser.add_argument('-13', '--thirteen', type=str, choices=['yes', 'no'], default='no', help='Thirteen')
        parser.add_argument('-14', '--fourteen', type=str, choices=['yes', 'no'], default='no', help='Fourteen')
        parser.add_argument('-15', '--fifteen', type=str, choices=['yes', 'no'], default='no', help='Fifteen')
        parser.add_argument('-16', '--sixteen', type=str, choices=['yes', 'no'], default='no', help='Sixteen')
        parser.add_argument('-17', '--seventeen', type=str, choices=['yes', 'no'], default='no', help='Seventeen')
        parser.add_argument('-18', '--eighteen', type=str, choices=['yes', 'no'], default='no', help='Eighteen')
        parser.add_argument('-19', '--nineteen', type=str, choices=['yes', 'no'], default='no', help='Nineteen')
        parser.add_argument('-20', '--twenty', type=str, choices=['yes', 'no'], default='no', help='Twenty')
        parser.add_argument('-21', '--twenty_one', type=str, choices=['yes', 'no'], default='no', help='Twenty one')
        parser.add_argument('-22', '--twenty_two', type=str, choices=['yes', 'no'], default='no', help='Twenty two')
        parser.add_argument('-23', '--twenty_three', type=str, choices=['yes', 'no'], default='no', help='Twenty three')
        parser.add_argument('-24', '--twenty_four', type=str, choices=['yes', 'no'], default='no', help='Twenty four')
        parser.add_argument('-25', '--twenty_five', type=str, choices=['yes', 'no'], default='no', help='Twenty five')
        parser.add_argument('-26', '--twenty_six', type=str, choices=['yes', 'no'], default='no', help='Twenty six')
        parser.add_argument('-27', '--twenty_seven', type=str, choices=['yes', 'no'], default='no', help='Twenty seven')
        parser.add_argument('-28', '--twenty_eight', type=str, choices=['yes', 'no'], default='no', help='Twenty eight')
        parser.add_argument('-29', '--twenty_nine', type=str, choices=['yes', 'no'], default='no', help='Twenty nine')
        parser.add_argument('-30', '--thirty', type=str, choices=['yes', 'no'], default='no', help='Thirty')
        parser.add_argument('-31', '--thirty_one', type=str, choices=['yes', 'no'], default='no', help='Thirty one')
        parser.add_argument('-32', '--thirty_two', type=str, choices=['yes', 'no'], default='no', help='Thirty two')
        parser.add_argument('-33', '--thirty_three', type=str, choices=['yes', 'no'], default='no', help='Thirty three')
        parser.add_argument('-34', '--thirty_four', type=str, choices=['yes', 'no'], default='no', help='Thirty four')
        parser.add_argument('-35', '--thirty_five', type=str, choices=['yes', 'no'], default='no', help='Thirty five')
        parser.add_argument('-36', '--thirty_six', type=str, choices=['yes', 'no'], default='no', help='Thirty six')
        parser.add_argument('-37', '--thirty_seven', type=str, choices=['yes', 'no'], default='no', help='Thirty seven')
        parser.add_argument('-38', '--thirty_eight', type=str, choices=['yes', 'no'], default='no', help='Thirty eight')
        parser.add_argument('-39', '--thirty_nine', type=str, choices=['yes', 'no'], default='no', help='Thirty nine')
        parser.add_argument('-40', '--forty', type=str, choices=['yes', 'no'], default='no', help='Forty')
        parser.add_argument('-41', '--forty_one', type=str, choices=['yes', 'no'], default='no', help='Forty one')
        parser.add_argument('-42', '--forty_two', type=str, choices=['yes', 'no'], default='no', help='Forty two')
        parser.add_argument('-43', '--forty_three', type=str, choices=['yes', 'no'], default='no', help='Forty three')
        parser.add_argument('-44', '--forty_four', type=str, choices=['yes', 'no'], default='no', help='Forty four')
        parser.add_argument('-45', '--forty_five', type=str, choices=['yes', 'no'], default='no', help='Forty five')
        parser.add_argument('-46', '--forty_six', type=str, choices=['yes', 'no'], default='no', help='Forty six')
        parser.add_argument('-47', '--forty_seven', type=str, choices=['yes', 'no'], default='no', help='Forty seven')
        parser.add_argument('-48', '--forty_eight', type=str, choices=['yes', 'no'], default='no', help='Forty eight')
        parser.add_argument('-49', '--forty_nine', type=str, choices=['yes', 'no'], default='no', help='Forty nine')
        parser.add_argument('-50', '--fifty', type=str, choices=['yes', 'no'], default='no', help='Fifty')
        parser.add_argument('-51', '--fifty_one', type=str, choices=['yes', 'no'], default='no', help='Fifty one')
        parser.add_argument('-52', '--fifty_two', type=str, choices=['yes', 'no'], default='no', help='Fifty two')
        parser.add_argument('-53', '--fifty_three', type=str, choices=['yes', 'no'], default='no', help='Fifty three')
        parser.add_argument('-54', '--fifty_four', type=str, choices=['yes', 'no'], default='no', help='Fifty four')
        parser.add_argument('-55', '--fifty_five', type=str, choices=['yes', 'no'], default='no', help='Fifty five')
        parser.add_argument('-56', '--fifty_six', type=str, choices=['yes', 'no'], default='no', help='Fifty six')
        parser.add_argument('-57', '--fifty_seven', type=str, choices=['yes', 'no'], default='no', help='Fifty seven')
        parser.add_argument('-58', '--fifty_eight', type=str, choices=['yes', 'no'], default='no', help='Fifty eight')
        parser.add_argument('-59', '--fifty_nine', type=str, choices=['yes', 'no'], default='no', help='Fifty nine')
        parser.add_argument('-60', '--sixty', type=str, choices=['yes', 'no'], default='no', help='Sixty')
        parser.add_argument('-61', '--sixty_one', type=str, choices=['yes', 'no'], default='no', help='Sixty one')
        parser.add_argument('-62', '--sixty_two', type=str, choices=['yes', 'no'], default='no', help='Sixty two')
        parser.add_argument('-63', '--sixty_three', type=str, choices=['yes', 'no'], default='no', help='Sixty three')
        parser.add_argument('-64', '--sixty_four', type=str, choices=['yes', 'no'], default='no', help='Sixty four')
        parser.add_argument('-65', '--sixty_five', type=str, choices=['yes', 'no'], default='no', help='Sixty five')
        parser.add_argument('-66', '--sixty_six', type=str, choices=['yes', 'no'], default='no', help='Sixty six')
        parser.add_argument('-67', '--sixty_seven', type=str, choices=['yes', 'no'], default='no', help='Sixty seven')
        parser.add_argument('-68', '--sixty_eight', type=str, choices=['yes', 'no'], default='no', help='Sixty eight')
        parser.add_argument('-69', '--sixty_nine', type=str, choices=['yes', 'no'], default='no', help='Sixty nine')
        parser.add_argument('-70', '--seventy', type=str, choices=['yes', 'no'], default='no', help='Seventy')
        parser.add_argument('-71', '--seventy_one', type=str, choices=['yes', 'no'], default='no', help='Seventy one')
        parser.add_argument('-72', '--seventy_two', type=str, choices=['yes', 'no'], default='no', help='Seventy two')
        parser.add_argument('-73', '--seventy_three', type=str, choices=['yes', 'no'], default='no', help='Seventy three')
        parser.add_argument('-74', '--seventy_four', type=str, choices=['yes', 'no'], default='no', help='Seventy four')
        parser.add_argument('-75', '--seventy_five', type=str, choices=['yes', 'no'], default='no', help='Seventy five')
        parser.add_argument('-76', '--seventy_six', type=str, choices=['yes', 'no'], default='no', help='Seventy six')
        parser.add_argument('-77', '--seventy_seven', type=str, choices=['yes', 'no'], default='no', help='Seventy seven')
        parser.add_argument('-78', '--seventy_eight', type=str, choices=['yes', 'no'], default='no', help='Seventy eight')
        parser.add_argument('-79', '--seventy_nine', type=str, choices=['yes', 'no'], default='no', help='Seventy nine')
        parser.add_argument('-80', '--eighty', type=str, choices=['yes', 'no'], default='no', help='Eighty')
        parser.add_argument('-81', '--eighty_one', type=str, choices=['yes', 'no'], default='no', help='Eighty one')
        parser.add_argument('-82', '--eighty_two', type=str, choices=['yes', 'no'], default='no', help='Eighty two')
        parser.add_argument('-83', '--eighty_three', type=str, choices=['yes', 'no'], default='no', help='Eighty three')
        parser.add_argument('-84', '--eighty_four', type=str, choices=['yes', 'no'], default='no', help='Eighty four')
        parser.add_argument('-85', '--eighty_five', type=str, choices=['yes', 'no'], default='no', help='Eighty five')
        parser.add_argument('-86', '--eighty_six', type=str, choices=['yes', 'no'], default='no', help='Eighty six')
        parser.add_argument('-87', '--eighty_seven', type=str, choices=['yes', 'no'], default='no', help='Eighty seven')
        parser.add_argument('-88', '--eighty_eight', type=str, choices=['yes', 'no'], default='no', help='Eighty eight')
        parser.add_argument('-89', '--eighty_nine', type=str, choices=['yes', 'no'], default='no', help='Eighty nine')
        parser.add_argument('-90', '--ninety', type=str, choices=['yes', 'no'], default='no', help='Ninety')
        parser.add_argument('-91', '--ninety_one', type=str, choices=['yes', 'no'], default='no', help='Ninety one')
        parser.add_argument('-92', '--ninety_two', type=str, choices=['yes', 'no'], default='no', help='Ninety two')
        parser.add_argument('-93', '--ninety_three', type=str, choices=['yes', 'no'], default='no', help='Ninety three')
        parser.add_argument('-94', '--ninety_four', type=str, choices=['yes', 'no'], default='no', help='Ninety four')
        parser.add_argument('-95', '--ninety_five', type=str, choices=['yes', 'no'], default='no', help='Ninety five')
        parser.add_argument('-96', '--ninety_six', type=str, choices=['yes', 'no'], default='no', help='Ninety six')
        parser.add_argument('-97', '--ninety_seven', type=str, choices=['yes', 'no'], default='no', help='Ninety seven')
        parser.add_argument('-98', '--ninety_eight', type=str, choices=['yes', 'no'], default='no', help='Ninety eight')
        parser.add_argument('-99', '--ninety_nine', type=str, choices=['yes', 'no'], default='no', help='Ninety nine')
        parser.add_argument('-100', '--hundred', type=str, choices=['yes', 'no'], default='no', help='Hundred')
        parser.add_argument('-101', '--hundred_one', type=str, choices=['yes', 'no'], default='no', help='Hundred one')
        parser.add_argument('-102', '--hundred_two', type=str, choices=['yes', 'no'], default='no', help='Hundred two')
        parser.add_argument('-103', '--hundred_three', type=str, choices=['yes', 'no'], default='no', help='Hundred three')
        parser.add_argument('-104', '--hundred_four', type=str, choices=['yes', 'no'], default='no', help='Hundred four')
        parser.add_argument('-105', '--hundred_five', type=str, choices=['yes', 'no'], default='no', help='Hundred five')
        parser.add_argument('-106', '--hundred_six', type=str, choices=['yes', 'no'], default='no', help='Hundred six')
        parser.add_argument('-107', '--hundred_seven', type=str, choices=['yes', 'no'], default='no', help='Hundred seven')
        parser.add_argument('-108', '--hundred_eight', type=str, choices=['yes', 'no'], default='no', help='Hundred eight')
        parser.add_argument('-109', '--hundred_nine', type=str, choices=['yes', 'no'], default='no', help='Hundred nine')
        parser.add_argument('-110', '--hundred_ten', type=str, choices=['yes', 'no'], default='no', help='Hundred ten')
        parser.add_argument('-111', '--hundred_eleven', type=str, choices=['yes', 'no'], default='no', help='Hundred eleven')
        parser.add_argument('-112', '--hundred_twelve', type=str, choices=['yes', 'no'], default='no', help='Hundred twelve')
        parser.add_argument('-113', '--hundred_thirteen', type=str, choices=['yes', 'no'], default='no', help='Hundred thirteen')
        parser.add_argument('-114', '--hundred_fourteen', type=str, choices=['yes', 'no'], default='no', help='Hundred fourteen')
        parser.add_argument('-115', '--hundred_fifteen', type=str, choices=['yes', 'no'], default='no', help='Hundred fifteen')
        parser.add_argument('-116', '--hundred_sixteen', type=str, choices=['yes', 'no'], default='no', help='Hundred sixteen')
        parser.add_argument('-117', '--hundred_seventeen', type=str, choices=['yes', 'no'], default='no', help='Hundred seventeen')
        parser.add_argument('-118', '--hundred_eighteen', type=str, choices=['yes', 'no'], default='no', help='Hundred eighteen')
        parser.add_argument('-119', '--hundred_nineteen', type=str, choices=['yes', 'no'], default='no', help='Hundred nineteen')
        parser.add_argument('-120', '--hundred_twenty', type=str, choices=['yes', 'no'], default='no', help='Hundred twenty')
        parser.add_argument('-121', '--hundred_twenty_one', type=str, choices=['yes', 'no'], default='no', help='Hundred twenty one')
        parser.add_argument('-122', '--hundred_twenty_two', type=str, choices=['yes', 'no'], default='no', help='Hundred twenty two')
        parser.add_argument('-123', '--hundred_twenty_three', type=str, choices=['yes', 'no'], default='no', help='Hundred twenty three')
        parser.add_argument('-124', '--hundred_twenty_four', type=str, choices=['yes', 'no'], default='no', help='Hundred twenty four')
        parser.add_argument('-125', '--hundred_twenty_five', type=str, choices=['yes', 'no'], default='no', help='Hundred twenty five')
        parser.add_argument('-126', '--hundred_twenty_six', type=str, choices=['yes', 'no'], default='no', help='Hundred twenty six')
        parser.add_argument('-127', '--hundred_twenty_seven', type=str, choices=['yes', 'no'], default='no', help='Hundred twenty seven')
        parser.add_argument('-128', '--hundred_twenty_eight', type=str, choices=['yes', 'no'], default='no', help='Hundred twenty eight')
        parser.add_argument('-129', '--hundred_twenty_nine', type=str, choices=['yes', 'no'], default='no', help='Hundred twenty nine')
        parser.add_argument('-130', '--hundred_thirty', type=str, choices=['yes', 'no'], default='no', help='Hundred thirty')
        parser.add_argument('-131', '--hundred_thirty_one', type=str, choices=['yes', 'no'], default='no', help='Hundred thirty one')
        parser.add_argument('-132', '--hundred_thirty_two', type=str, choices=['yes', 'no'], default='no', help='Hundred thirty two')
        parser.add_argument('-133', '--hundred_thirty_three', type=str, choices=['yes', 'no'], default='no', help='Hundred thirty three')
        parser.add_argument('-134', '--hundred_thirty_four', type=str, choices=['yes', 'no'], default='no', help='Hundred thirty four')
        parser.add_argument('-135', '--hundred_thirty_five', type=str, choices=['yes', 'no'], default='no', help='Hundred thirty five')
        parser.add_argument('-136', '--hundred_thirty_six', type=str, choices=['yes', 'no'], default='no', help='Hundred thirty six')
        parser.add_argument('-137', '--hundred_thirty_seven', type=str, choices=['yes', 'no'], default='no', help='Hundred thirty seven')
        parser.add_argument('-138', '--hundred_thirty_eight', type=str, choices=['yes', 'no'], default='no', help='Hundred thirty eight')
        parser.add_argument('-139', '--hundred_thirty_nine', type=str, choices=['yes', 'no'], default='no', help='Hundred thirty nine')
        parser.add_argument('-140', '--hundred_forty', type=str, choices=['yes', 'no'], default='no', help='Hundred forty')
        parser.add_argument('-141', '--hundred_forty_one', type=str, choices=['yes', 'no'], default='no', help='Hundred forty one')
        parser.add_argument('-142', '--hundred_forty_two', type=str, choices=['yes', 'no'], default='no', help='Hundred forty two')
        parser.add_argument('-143', '--hundred_forty_three', type=str, choices=['yes', 'no'], default='no', help='Hundred forty three')
        parser.add_argument('-144', '--hundred_forty_four', type=str, choices=['yes', 'no'], default='no', help='Hundred forty four')
        parser.add_argument('-145', '--hundred_forty_five', type=str, choices=['yes', 'no'], default='no', help='Hundred forty five')
        parser.add_argument('-146', '--hundred_forty_six', type=str, choices=['yes', 'no'], default='no', help='Hundred forty six')
        parser.add_argument('-147', '--hundred_forty_seven', type=str, choices=['yes', 'no'], default='no', help='Hundred forty seven')
        parser.add_argument('-148', '--hundred_forty_eight', type=str, choices=['yes', 'no'], default='no', help='Hundred forty eight')
        parser.add_argument('-149', '--hundred_forty_nine', type=str, choices=['yes', 'no'], default='no', help='Hundred forty nine')
        parser.add_argument('-150', '--hundred_fifty', type=str, choices=['yes', 'no'], default='no', help='Hundred fifty')
        parser.add_argument('-151', '--hundred_fifty_one', type=str, choices=['yes', 'no'], default='no', help='Hundred fifty one')
        parser.add_argument('-152', '--hundred_fifty_two', type=str, choices=['yes', 'no'], default='no', help='Hundred fifty two')
        parser.add_argument('-153', '--hundred_fifty_three', type=str, choices=['yes', 'no'], default='no', help='Hundred fifty three')
        parser.add_argument('-154', '--hundred_fifty_four', type=str, choices=['yes', 'no'], default='no', help='Hundred fifty four')
        parser.add_argument('-155', '--hundred_fifty_five', type=str, choices=['yes', 'no'], default='no', help='Hundred fifty five')
        parser.add_argument('-156', '--hundred_fifty_six', type=str, choices=['yes', 'no'], default='no', help='Hundred fifty six')
        parser.add_argument('-157', '--hundred_fifty_seven', type=str, choices=['yes', 'no'], default='no', help='Hundred fifty seven')
        parser.add_argument('-158', '--hundred_fifty_eight', type=str, choices=['yes', 'no'], default='no', help='Hundred fifty eight')
        parser.add_argument('-159', '--hundred_fifty_nine', type=str, choices=['yes', 'no'], default='no', help='Hundred fifty nine')
        parser.add_argument('-160', '--hundred_sixty', type=str, choices=['yes', 'no'], default='no', help='Hundred sixty')
        parser.add_argument('-161', '--hundred_sixty_one', type=str, choices=['yes', 'no'], default='no', help='Hundred sixty one')
        parser.add_argument('-162', '--hundred_sixty_two', type=str, choices=['yes', 'no'], default='no', help='Hundred sixty two')
        parser.add_argument('-163', '--hundred_sixty_three', type=str, choices=['yes', 'no'], default='no', help='Hundred sixty three')
        parser.add_argument('-164', '--hundred_sixty_four', type=str, choices=['yes', 'no'], default='no', help='Hundred sixty four')
        parser.add_argument('-165', '--hundred_sixty_five', type=str, choices=['yes', 'no'], default='no', help='Hundred sixty five')
        parser.add_argument('-166', '--hundred_sixty_six', type=str, choices=['yes', 'no'], default='no', help='Hundred sixty six')
        parser.add_argument('-167', '--hundred_sixty_seven', type=str, choices=['yes', 'no'], default='no', help='Hundred sixty seven')
        parser.add_argument('-168', '--hundred_sixty_eight', type=str, choices=['yes', 'no'], default='no', help='Hundred sixty eight')
        parser.add_argument('-169', '--hundred_sixty_nine', type=str, choices=['yes', 'no'], default='no', help='Hundred sixty nine')
        parser.add_argument('-170', '--hundred_seventy', type=str, choices=['yes', 'no'], default='no', help='Hundred seventy')
        parser.add_argument('-171', '--hundred_seventy_one', type=str, choices=['yes', 'no'], default='no', help='Hundred seventy one')
        parser.add_argument('-172', '--hundred_seventy_two', type=str, choices=['yes', 'no'], default='no', help='Hundred seventy two')
        parser.add_argument('-173', '--hundred_seventy_three', type=str, choices=['yes', 'no'], default='no', help='Hundred seventy three')
        parser.add_argument('-174', '--hundred_seventy_four', type=str, choices=['yes', 'no'], default='no', help='Hundred seventy four')
        parser.add_argument('-175', '--hundred_seventy_five', type=str, choices=['yes', 'no'], default='no', help='Hundred seventy five')
        parser.add_argument('-176', '--hundred_seventy_six', type=str, choices=['yes', 'no'], default='no', help='Hundred seventy six')
        parser.add_argument('-177', '--hundred_seventy_seven', type=str, choices=['yes', 'no'], default='no', help='Hundred seventy seven')
        parser.add_argument('-178', '--hundred_seventy_eight', type=str, choices=['yes', 'no'], default='no', help='Hundred seventy eight')
        parser.add_argument('-179', '--hundred_seventy_nine', type=str, choices=['yes', 'no'], default='no', help='Hundred seventy nine')
        parser.add_argument('-180', '--hundred_eighty', type=str, choices=['yes', 'no'], default='no', help='Hundred eighty')
        parser.add_argument('-181', '--hundred_eighty_one', type=str, choices=['yes', 'no'], default='no', help='Hundred eighty one')
        parser.add_argument('-182', '--hundred_eighty_two', type=str, choices=['yes', 'no'], default='no', help='Hundred eighty two')
        parser.add_argument('-183', '--hundred_eighty_three', type=str, choices=['yes', 'no'], default='no', help='Hundred eighty three')
        parser.add_argument('-184', '--hundred_eighty_four', type=str, choices=['yes', 'no'], default='no', help='Hundred eighty four')
        parser.add_argument('-185', '--hundred_eighty_five', type=str, choices=['yes', 'no'], default='no', help='Hundred eighty five')
        parser.add_argument('-186', '--hundred_eighty_six', type=str, choices=['yes', 'no'], default='no', help='Hundred eighty six')
        parser.add_argument('-187', '--hundred_eighty_seven', type=str, choices=['yes', 'no'], default='no', help='Hundred eighty seven')
        parser.add_argument('-188', '--hundred_eighty_eight', type=str, choices=['yes', 'no'], default='no', help='Hundred eighty eight')
        parser.add_argument('-189', '--hundred_eighty_nine', type=str, choices=['yes', 'no'], default='no', help='Hundred eighty nine')
        parser.add_argument('-190', '--hundred_ninety', type=str, choices=['yes', 'no'], default='no', help='Hundred ninety')
        parser.add_argument('-191', '--hundred_ninety_one', type=str, choices=['yes', 'no'], default='no', help='Hundred ninety one')
        parser.add_argument('-192', '--hundred_ninety_two', type=str, choices=['yes', 'no'], default='no', help='Hundred ninety two')
        parser.add_argument('-193', '--hundred_ninety_three', type=str, choices=['yes', 'no'], default='no', help='Hundred ninety three')
        parser.add_argument('-194', '--hundred_ninety_four', type=str, choices=['yes', 'no'], default='no', help='Hundred ninety four')
        parser.add_argument('-195', '--hundred_ninety_five', type=str, choices=['yes', 'no'], default='no', help='Hundred ninety five')
        parser.add_argument('-196', '--hundred_ninety_six', type=str, choices=['yes', 'no'], default='no', help='Hundred ninety six')
        parser.add_argument('-197', '--hundred_ninety_seven', type=str, choices=['yes', 'no'], default='no', help='Hundred ninety seven')
        parser.add_argument('-198', '--hundred_ninety_eight', type=str, choices=['yes', 'no'], default='no', help='Hundred ninety eight')
        parser.add_argument('-199', '--hundred_ninety_nine', type=str, choices=['yes', 'no'], default='no', help='Hundred ninety nine')
        parser.add_argument('-200', '--two_hundred', type=str, choices=['yes', 'no'], default='no', help='Two hundred')
        parser.add_argument('-201', '--two_hundred_one', type=str, choices=['yes', 'no'], default='no', help='Two hundred one')
        parser.add_argument('-202', '--two_hundred_two', type=str, choices=['yes', 'no'], default='no', help='Two hundred two')
        parser.add_argument('-203', '--two_hundred_three', type=str, choices=['yes', 'no'], default='no', help='Two hundred three')
        parser.add_argument('-204', '--two_hundred_four', type=str, choices=['yes', 'no'], default='no', help='Two hundred four')
        parser.add_argument('-205', '--two_hundred_five', type=str, choices=['yes', 'no'], default='no', help='Two hundred five')
        parser.add_argument('-206', '--two_hundred_six', type=str, choices=['yes', 'no'], default='no', help='Two hundred six')
        parser.add_argument('-207', '--two_hundred_seven', type=str, choices=['yes', 'no'], default='no', help='Two hundred seven')
        parser.add_argument('-208', '--two_hundred_eight', type=str, choices=['yes', 'no'], default='no', help='Two hundred eight')
        parser.add_argument('-209', '--two_hundred_nine', type=str, choices=['yes', 'no'], default='no', help='Two hundred nine')
        parser.add_argument('-210', '--two_hundred_ten', type=str, choices=['yes', 'no'], default='no', help='Two hundred ten')
        parser.add_argument('-211', '--two_hundred_eleven', type=str, choices=['yes', 'no'], default='no', help='Two hundred eleven')
        parser.add_argument('-212', '--two_hundred_twelve', type=str, choices=['yes', 'no'], default='no', help='Two hundred twelve')
        parser.add_argument('-213', '--two_hundred_thirteen', type=str, choices=['yes', 'no'], default='no', help='Two hundred thirteen')
        parser.add_argument('-214', '--two_hundred_fourteen', type=str, choices=['yes', 'no'], default='no', help='Two hundred fourteen')
        parser.add_argument('-215', '--two_hundred_fifteen', type=str, choices=['yes', 'no'], default='no', help='Two hundred fifteen')
        parser.add_argument('-216', '--two_hundred_sixteen', type=str, choices=['yes', 'no'], default='no', help='Two hundred sixteen')
        parser.add_argument('-217', '--two_hundred_seventeen', type=str, choices=['yes', 'no'], default='no', help='Two hundred seventeen')
        parser.add_argument('-218', '--two_hundred_eighteen', type=str, choices=['yes', 'no'], default='no', help='Two hundred eighteen')
        parser.add_argument('-219', '--two_hundred_nineteen', type=str, choices=['yes', 'no'], default='no', help='Two hundred nine
```