گزارش تمرین عملی دوم درس هوش مصنوعی

نگین اسماعیل زاده ۹۷۱۰۴۰۳۴

توضیح کلی جهت اجرا :

برای اجرای برنامه در قسمت main یک متغیر به اسم Input قرار داده شده است. جهت اجرای سوال اول مقدار آن را برابر با "Q1" و جهت اجرای سوال دوم این مقدار را برابر با "Q2" قرار میدهیم. همچنین موارد دیگیری که رو به روی آن ها کامنت نوشته شده است قابل تنظیم هستند، مثل درصد دیتا های آموزش و تعداد بازه های تقسیم بندی :

```
if __name__ == '__main__':

    Input = "Q2"       # Q1 or Q2 ?
    if Input == "Q1" :
        attributes, data_strings = input_reader("Part1.csv")

        data = data_encoder(data_strings)
        root = Node()
        classification_name = ""
        decision = ""
        Plurality_Value = ""
        Gain = None
        decision_tree_creator(root, attributes, data, data_strings, classification_name, decision, Plurality_Value, Gain)
        visualization(root)

    elif Input == "Q2" :
        percentage = 80         # percentage of learn data
        Ranges =  2             # How many ranges?
        attributes, continuous_data_strings = input_reader("diabetes.csv")


        continuous_learn_data_strings, continuous_test_data_strings = learn_data_picker(continuous_data_strings,percentage)
        number_of_groups = [Ranges for _ in range(len(attributes)-1)]
        discrete_learn_data_strings  = continuous_to_discrete(continuous_learn_data_strings,number_of_groups)
        learn_data = data_encoder(discrete_learn_data_strings)
        root = Node()
        classification_name = ""
        decision = ""
        Plurality_Value = ""
        Gain = None
```

توضیح کلی روش حل :

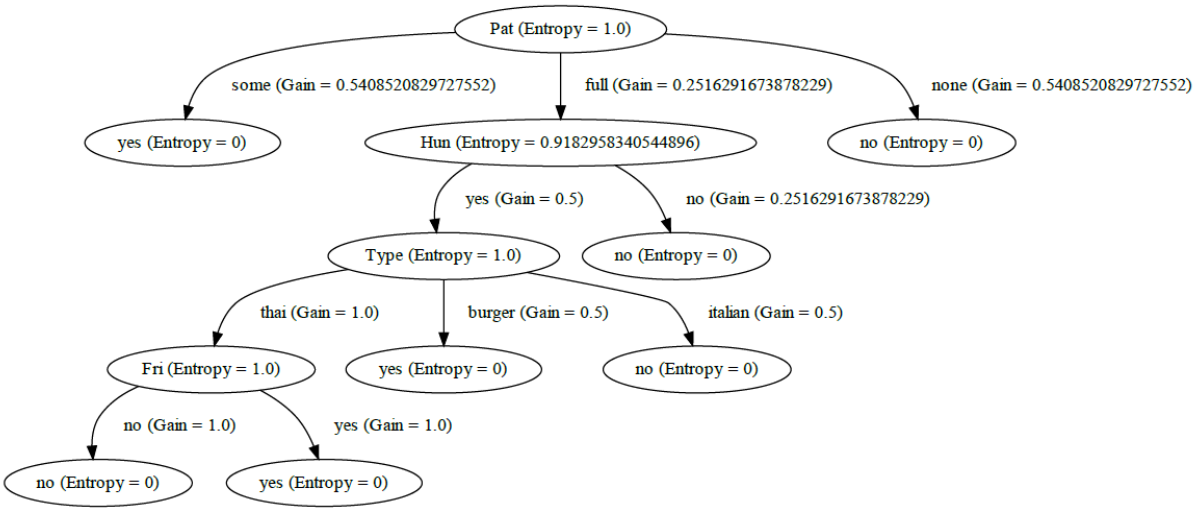
در سوال اول مقادیر گسسته بودند بنابراین درخت تقیربا بدون چالش تشکیل و رسم شده است و معیار انتخاب ارجح ترین ویژگی برای کاهش پیچیدگی درخت نیز میزان gain در نظر گرفته شده است.

در سوال دوم مقادیر پیوسته هستند بنابراین ابتدا برای تبدیل آن ها به نقاط گسسته هر بازه را معادل یک نقطه در نظر گرفتیم و بازه ها را برای مقادیر ۳ ، ۴ ، ۵ تست شدند. همچنین درصد داده های آموزش ۸۰ در نظر گرفته شده است و برای مقادیر ۹۰، ۷۰ و ۵۰ نیز تست شده است.

سوال اول)

در ابن سوال مطابق اسلاید های درس درخت تصمیم مسئله صبر کردن برای غذا رسم شده است و اگر کد اجرا شود خروجی آن را در قالب pdf مبینیم. (مقادیر gain و آنتروپی هم در نمودار به تفیکیک مشخص شده است)

به دلیل کوچک بودن اندازه ی این درخت در شکل زیر آورده شده است :



همانطور که میبینیم چون ویژگی ها خیلی خوب استخراج شده بودند (هم از جهت نویز کم هم از جهت ارتباط) در برگ های درخت شاهد آنتروپی صفر هستیم.

سوال دوم :

خروجی این سوال به دلیل بزر/ بودن درخت در گزارش آورده نشده است و در pdf خروجی حاصل از اجرای کد قابل رویت است. بهترین دسته بندی برای تعداد ۴ بازه ایجاد شد (با فرض ثابت بودن تعداد بازه ها و همچنین دو بازه ی ابتدای و انتهایی سقف و کف).

در این حالت برای درصد داده های آموزش ۸۰ (انتخاب داده ها رندم است برای نتیجه ی معقول تر) مقدار دقت داده های تست به شکل زیر است :

```
Accuracy is  70.12987012987013 %
```

برای درصد داده های آموزش ۹۰ (انتخاب داده ها رندم است برای نتیجه ی معقول تر) مقدار دقت داده های تست به شکل زیر است :

```
Accuracy is  66.23376623376623 %
```

برای درصد داده های آموزش ۷۰ (انتخاب داده ها رندم است برای نتیجه ی معقول تر) مقدار دقت داده های تست به شکل زیر است :

```
Accuracy is  67.09956709956711 %
```

برای درصد داده های آموزش ۵۰ (انتخاب داده ها رندم است برای نتیجه ی معقول تر) مقدار دقت داده های تست به شکل زیر است :

```
Accuracy is  66.92708333333334 %
```

همانطور که میبینیم میزان دقت تا یک جایی(حدودا ۸۰ به ۲۰ نسبت داده های اموزش به تست) افزایش پیدا میکند و از آن حد به بعد مجددا کاهش می یابد که علت آن هممطابق آن چه در درس بحث شد overfit شدن درخت است.

برای خروجی کد این بخش نیز خروجی به شکل تصویر است که مقادیر آنتروپی و گین هم علاوه بر شمای کلی درخت در آن قابل مشاهده است.

ایده های افزایش دقت :

ساده ترین ایده این هست که بر اساس تجمع داده ها تعداد بازه ها را برای هر ویژگی متفاوت در نظر بگیریم (البته این کار در کد فعلی ساده و قابل انجام است ولی نتایج آن برسی نشده)

ایده های بعدی این هست که روش گسسته سازی را عوض کنیم (با یکیاز روش هایی که در درس خواندیم)