محمدرضا تاجیک 400521198

پروژه‌ی درخت تصمیم

1. تحلیل دیتا:

در ابتدای کار من شروع کردم و یک سری تحلیل روی فایل csv انجام دادم. روش انجام این تحلیل هم به این صورت بود که با استفاده از کتابخانه های پایتون مثل pandas و numpy فایل csv رو باز کردم و شروع کردم به پیدا کردن تمام مقدار های ممکن برای یک attribute خاص. مثلا متوجه شدم که type فقط 5 مقدار مختلف داره یا اینکه step 10 مقدار مختلف داره. بعد از بررسی های بیشتر متوجه شدم که تمامی مقادیر nameOrig با همدیگه فرق میکنن و یونیک هستن، در نتیجه نباید توی درخت تصمیم همچین راسی داشته باشیم.
همچنین اینکه تمامی مقادیر بقیه‌ی attribute ها پیوسته هستند و باید گسسته سازی بشن.
بعد از بررسی attribute ها سراغ label رفتم و متوجه شدم که 98 درصد این دیتا منفی هستند و فقط 2 درصد مثبت!!!

2. گسسته سازی و data cleaning:

بعد از مرحله‌ی تحلیل داده سراغ گسسته سازی دیتا رفتم، که در این مرحله اومدم و attribute های nameOrig و nameDest رو کلا پاک کردم از روی دیتا ست.

برای 5 مقدار مختلف type، یک عدد مشخص بین 0 تا 4 درنظر گرفتم و اون ها رو مپ کردم.

سپس مقادیر کمترین و بیشترین رو برای attribute های amount و oldbalanceorg و newbalanceorg و oldbalancedest و newbalancedest محاسبه کردم و برای هرکدوم 8 دسته مختلف درنظر گرفتم که با توجه به مقدار مپ بشن به یه عددی بین 0 تا 7.

در نهایت دو تا لیست درست کردم و دیتای مثبت و منفی رو از هم جدا و شافل کردم و 200 دیتای مثبت و 9800 تا دیتای منفی رو جدا کردم برای دیتای train درخت و همچنین دقیقا همین مقدار دیتا برای تست .

3. ساخت درخت تصمیم :

در این مرحله با استفاده از 2 معیار entropy و gini index که با استفاده ازین دوتا میتونیم بهترین attribute رو برای این راس خاص پیدا کنیم، 2 تا درخت متفاوت ساختم که در ساختار این درخت به دلیل اینکه یک راس خاص ممکن بود بیشتر از 2 تا فرزند داشته باشه، از دیکشنری استفاده کردم، به این صورت که هر راس یک key در دیکشنری است که value آن یک دیکشنری از فرزندان آن راس است و به همین شکل تا به leaf برسیم. این leaf 2 مقدار 0 و 1 دارد که به معنی مثبت یا منفی بودن نتیجه کلاهبرداری میباشد.

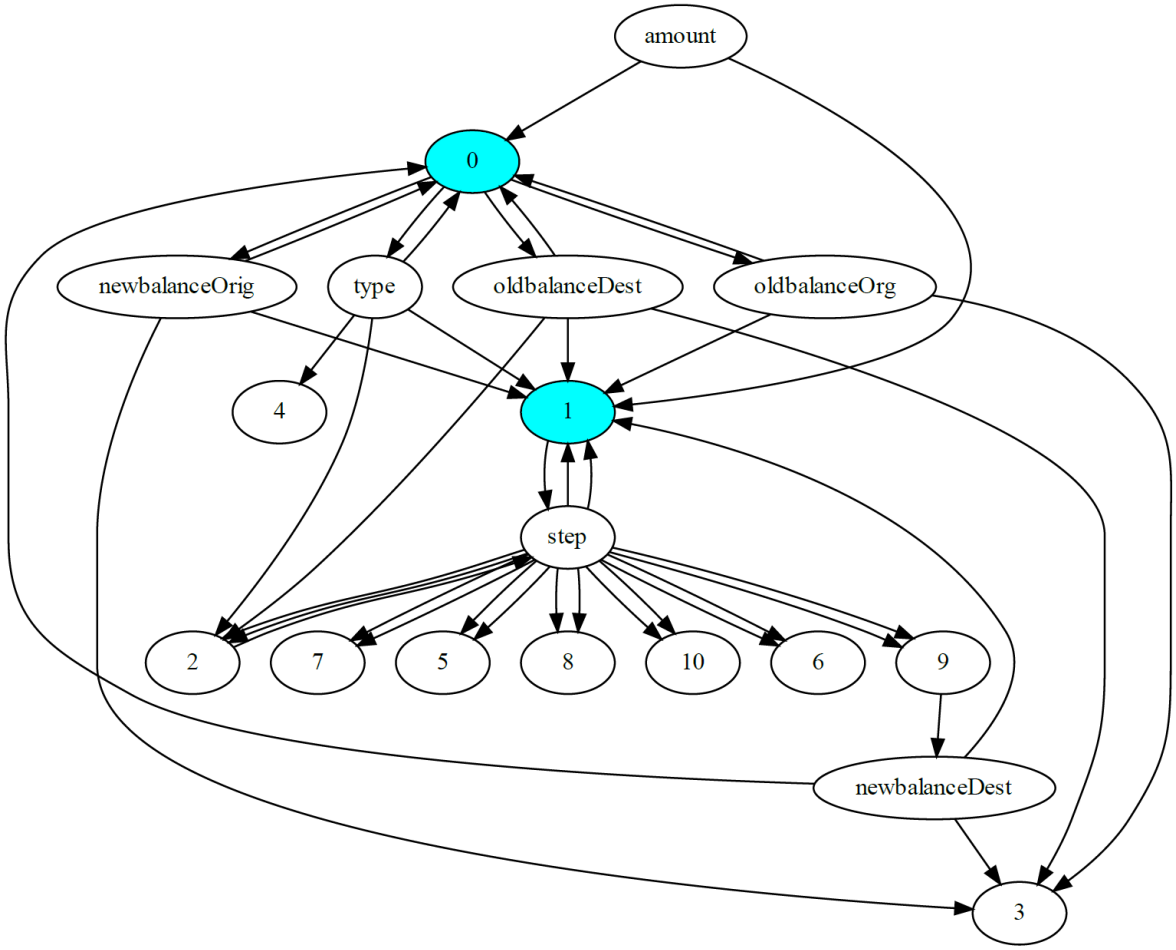
4. تست کردن درخت تصمیم:

توی این مرحله با استفاده از همون دیتایی که توی مرحله 2 بدست آوردیم هر دو درخت تصمیم که با entropy و gini index ساخته شدن رو تست میکنیم و دقت هرکدوم رو محاسبه میکنیم.
که نتیجه تقریبا 60 به 40 به نفع entropy هست و دقت بیشتری به ما میده.

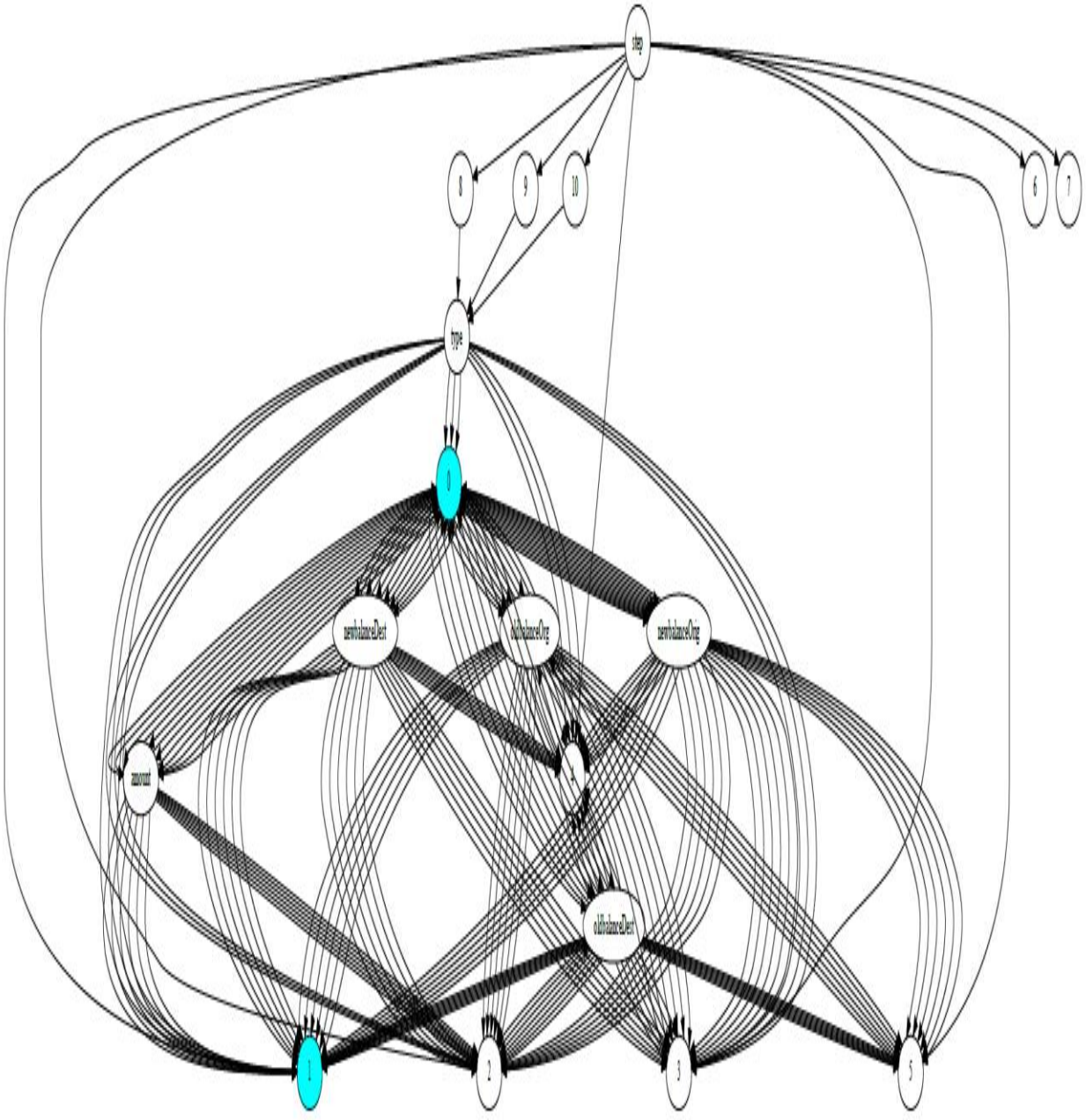
5. نمایش درخت تصمیم:

در نهایت با استفاده از کتابخانه graphviz در پایتون، هر دو درخت رو در digraph که توی کتابخانه graphviz هست ذخیره کردم و رندر گرفتم و عکس این درخت ها رو در قالب فایل های svg و pdf ذخیره کردم که این پایین میتونید مدل های مختلف با تعداد دیتای train حدودا 100 و 500 ببینید. (بیشتر از این تعداد به سختی قابل تشخیص هست)

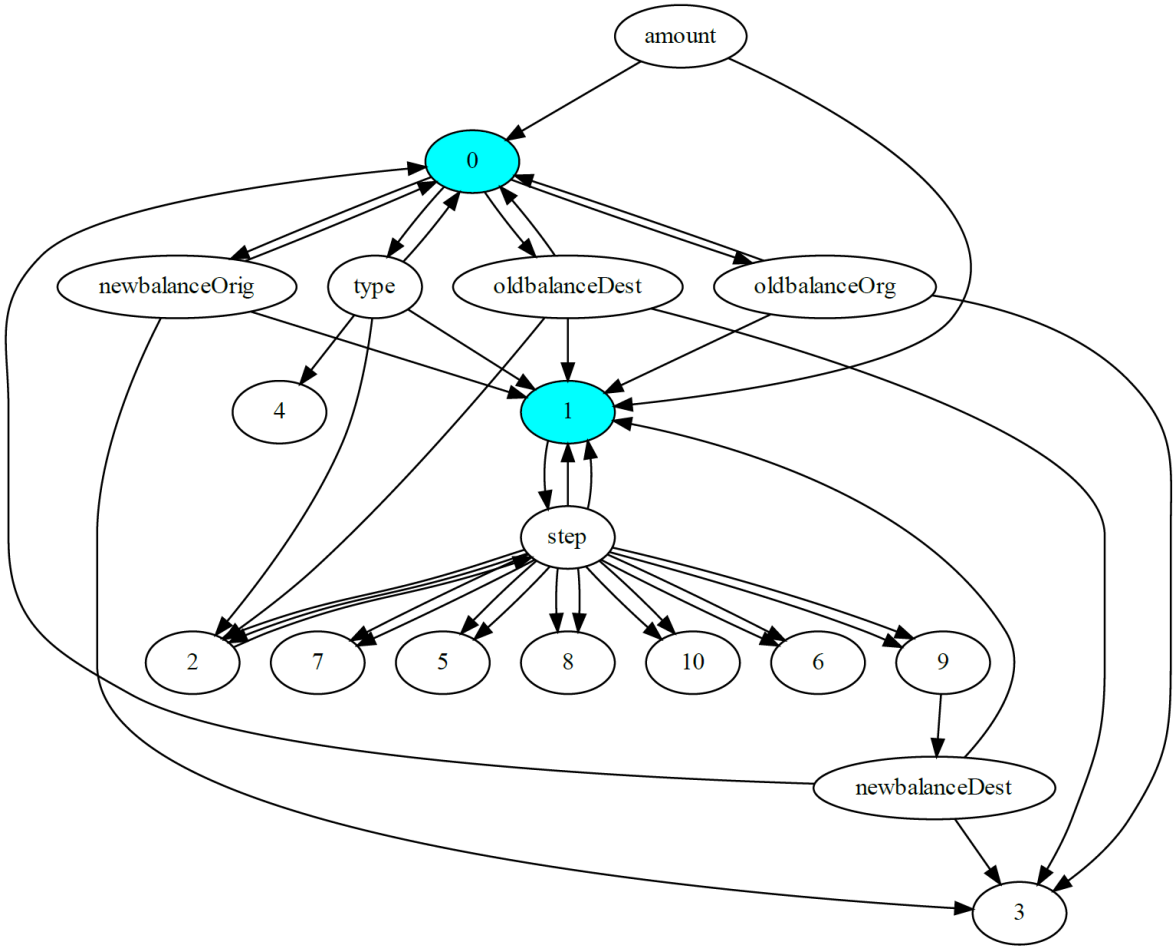
Entropy tree with 100 data train:



Entropy tree with 500 data train:



Gini tree with 100 data train:



Gini tree with 500 data train: