ابتدا چهار الگوریتم را با روش bag of words چک کردم و adaboost و random forest دقت 85 درصد و logistic regression و neural network که در آن از لایه های dense استفاده کردم دقت 87 درصد میدادند. البته neural network به شکل قابل ملاحظه ای برای این روش سریع عمل میکرد.

سپس برای قسمت بعد سعی کردم در لایه ی اول از embedding استفاده کنم. که در Data cleaning برخی تغییراتی که دادم این بود که stopwords را حذف نکردم و تمامی کاراکترهای عددی را در ابتدا به “num” تبدیل کردم که در فیلتر حذف تمامی کاراکترهای غیر عددی حذف نشوند. برای من همچنین الگوریتم word2vec خیلی کند کار میکرد چون رو ویندوز بودم. اما اگر میتوانستم از آن استفاده کنم باید ابتدا جمله های یک پاراگراف را با tokenizer از هم جدا میکردم و سپس ارتباط بین کلمات را پیدا میکردم.

در این بخش من همچنین به دو روش سعی کردم مدل را بسازم که هر دو بار نتایج 75 76 درصد داد و موفقیت آمیز نبود. ضمن اینکه اجرای آن زمان زیادی میگرفت که مرا از ایجاد تغییرات بیشتر منصرف کرد. یک روش همان مدل Sequential بود که در آن بعد از لایه embedding از یک bidirectional lstm که به صورت many to one بود استفاده کردم. برای بار دوم هم از یک مدل فانکشنال استفاده کردم، که بعد از لایه embedding دو شاخه میشد و یک بار یک lstm با یک خروجی مانند دفعه قبل و بار دیگر یک lstm با 4 خروجی بود که فیچرهای بیشتری در بیاورد. در پایان این دو را بهم کانکت میکردم و بعد از دو لایه dense خروجی را میگرفتم.

برای روش های bag of words از optimizers نوع Sgd استفاده کردم که دقت مدل در هنگام تست به دقت که موقع train اعلام میکرد بسیار نزدیک بود بر خلاف adam که اختلاف بسیاری با هنگام تست کردن داشت.

اما برای روش های بعد sgd به خوبی جواب نمیداد و من به ناپار از همان adam استفاده کردم. همچنین به دلیل زمان زیادی که میگرفت من اکثر آن ها را با همان تعداد epoch 10 تست کردم که در صورت زیاد کردن مقدار ان قطعا نتایج حتی اندکی بهتر حاصل میشد.