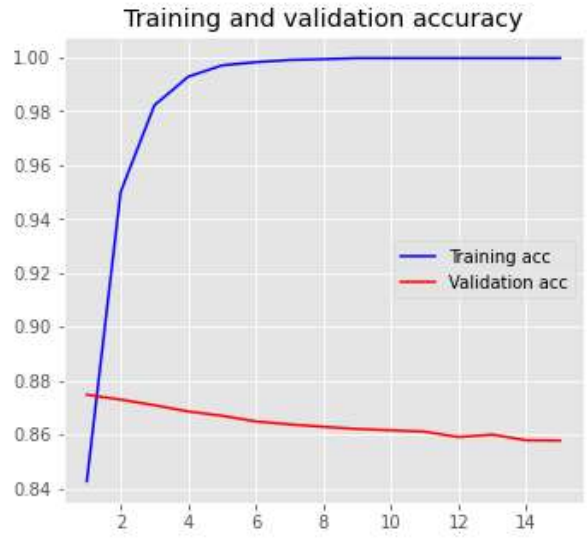
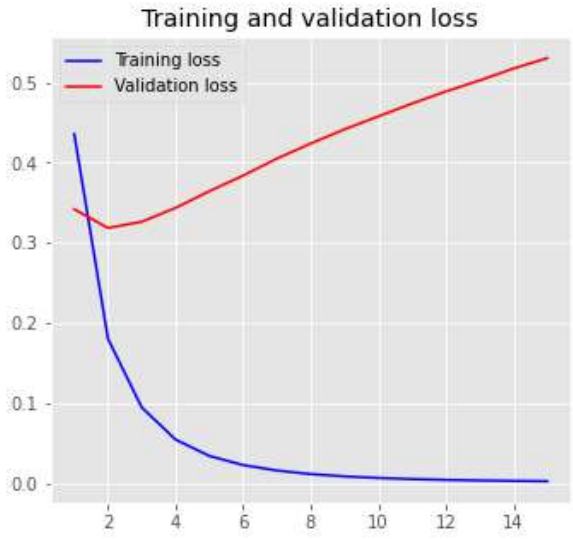
در این مجموعه دادگان دو تا دسته ی تست و آموزش هست و هر کدام شامل دو دسته فایل txt. هستند که یک مجموعه شان مثبت و دیگری منفی ست این دسته بندی در حکم متغیر هدف است. هر متن در یک فایل جداگانه ی txt. ذخیره شده. برای خواندن تمام این متن ها و تشکیل یک دیتافریم خام اولیه، ادرس و نام تمام فایل ها در هر فولدر pos و neg را در لیستmylist به کمک glob قرار میدهم. سپس هر کدام از این ها را با برچسب ۰ یا ۱ به ترتیب برای فایل های پوشه ی neg و pos در یک تاپل در دو لیست poss و negs قرار می‌دهم و نهایتا تمام این ها را به یک دیتافریم با دو ستون sentences و target تبدیل می‌کنم. تمام این مراحل برای داده های تست و آموزش به طور جداگانه و یکسان انجام شده. دیتافریم ها شافل شده اند تا داده های با برچسب ۰ و ۱ درهم تلفیق شوند.

از sklearn با استفاده از CountVectorizer یک ساک کلمات از تمام جملات df_train می‌سازم. این قابلیت در این مدل وجود دارد که برای متون انگلیسی کلمات stopword و علایم نگارشی را در ساک کلمه لحاظ نکند و نیاز به پیش پردازش های این چنینی از نداریم. جملات دیتافریم ها را به صورت نمایشی از مدل ساک کلمات ساخته شده در X_train و X_test ذخیره می‌کنم. این ها باید به صورت آرایه های numpy تبدیل شوند تا بتوان در شبکه ی عصبی بخش بعد ازشان استفاده کرد. من به دلیل اینکه در کامپیوتر شخصی خودم خافظه ی کافی پردازش کل این داده ها را به صورت بردار نداشتم و امکان استفاده از پردازش های آنلاین هم برایم فراهم نبود، هر دو مجموعه تیت و آموزش را از ۲۵۰۰۰ به ۱۰۰۰۰کوچک تر کردم و بردارهای X_train_arr و X_test_arr برای ورود به شبکه عصبی مرحله بعد آماده اند.

با استفاده از keras از tensorflow یک شبکه عصبی با یک لایه ورودی ۱۰۰۰۰ تایی، با یک لایه پنهان ۱۰ تایی با تابع فعالسازی relu و یک تک واحد خروجی با تابع فعالسازی سیگموید تعریف کردم. این شبکه با معیار accuracy آموزش می‌بیند و برای تابع loss از binary_crossentropy ساخته می‌شود و با ۱۵ ایپاک و در دسته های ۶۴ تایی آپدیت وزن و آموزش صورت می‌گیرد.

نمودار تست و آموزش روی loss و accuracy آمده و نتیجه ی بهترین مدل به دست آمده و نهایی روی داده های تست و آموزش هم چاپ شده. این نتیجه روی ایپاک ۲ در ابتدای کار به دست آمده (با معیار loss) و بعد هرچه اموزش ادامه پیدا می‌کند بیش برازش بیشتر می‌شود و مدل قابلیت generalization از دست می‌دهد. دقت ۸۶ دسته بندی درصد



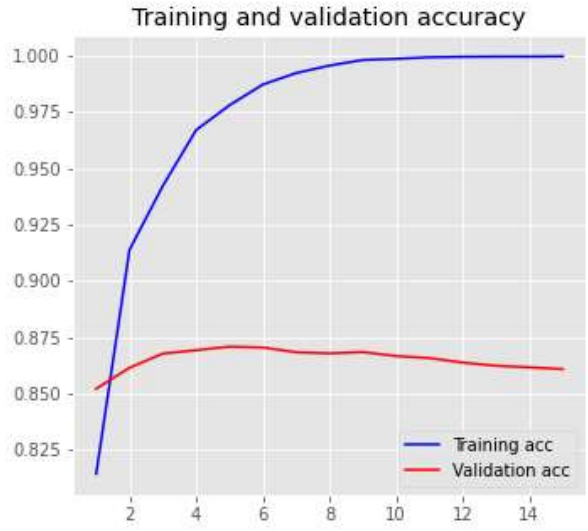


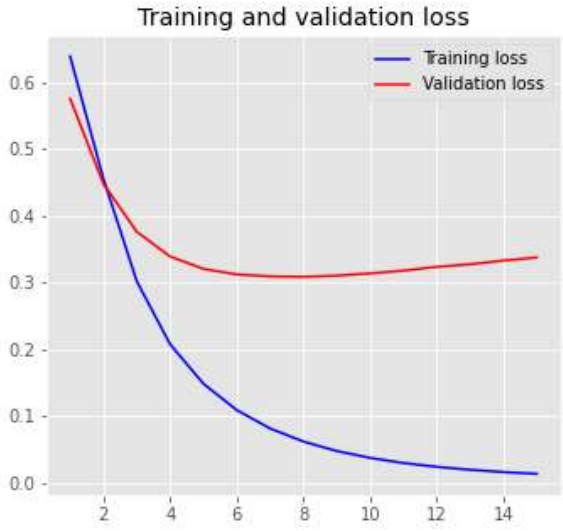

```
Training Accuracy: 0.9997
Training Loss: 0.0021

Testing Accuracy:  0.8578
Testing Loss:  0.5302
```

یک مدل دوم می‌سازم که به جای داده های ساک کلمات از روش tfidf استفاده کند. با استفاده از TfidfTransformer از sklearn فرکانس تکرار کلمات را در هر متن پیدا می‌کند. دوباره داده ها را به صورت آرایه ی numpy تبدیل می‌کنم و به همان علت محدودیت سیستم پردازش که دارم با داده ها ی ۱۰۰۰۰ تایی کار می‌کنم. همان معماری شبکه‌ی قبلی را روی این داده ها آموزش می‌دهم . نحوه ی آموزش هم همان است.

نمودار تست و آموزش آورده شد. بهترین نتیجه در حدود ایپاک ۵ به دست آمده و بعد افت نتیجه روی تست را داریم. (بیش برازش) تغیری با مدل قبلی در نتیجه و عملکرد مدل به دست نیامد.

```
Training Accuracy: 0.9997
Training Loss: 0.0116
Testing Accuracy:  0.8608
Testing Loss:  0.3376
```





مدل نهایی به دست آمده از این شبکه با دقت ۸۶ درصد قابل استفاده ست.