프랑스 영화 리뷰 데이터 구축과 감정분석

By Sungwoo Stanislas Kwon (권성우) 2023-82030

주제 변경 이유:

- 형태소 분석 경험 부족으로 인한 어려움.
- 보여주고자 했던 바를 증명하기에는 부족했던 술어들의 다양성.
- 복잡한 분석 보다는 그런 분석에 사용 될 수 있는 새로운 데이터를 스스로 구축 해보는 것도 의미있는 일 일것 같다고 생각.

새로운 주제:

- 프랑스 텔레비전 산업 관련 사이트 allociné.fr
- 그 중 '영화' 목록에있는 영화들의 제목, 별점, 리뷰 등등 스크랩해와 데이터셋 구축, 감정분석.

주요 tasks:

- BeautifulSoup를 이용한 데이터 스크래핑과 데이터셋 구축, 전처리 & 간단한 수치들 시각화.
- TF-IDF embedding을 이용해 간단한 감정 분류기 구축(로지스틱 회귀)과 분.

1. 데이터 스크래핑과 전처리

allociné.fr에서 십만개가 넘는 영화 url들을 추출하고 그 url에 접근해서 해당 영화 리뷰 (최대) 30개, 별점등을 스크래핑해와 데이터셋 구축

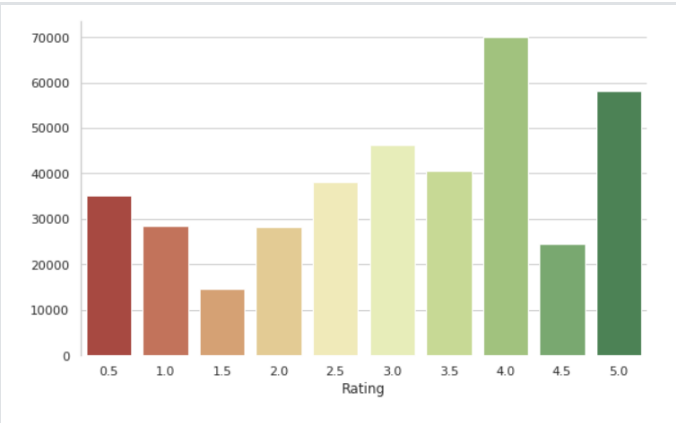
- 데이터셋을 1부터10까지 만들어야 했기에 사실상 가장 복잡했던 부분.
- 또한 처리해야하는 데이터가 상당히 많아서 코드 돌리는데도 시간이 매우 오래걸림.

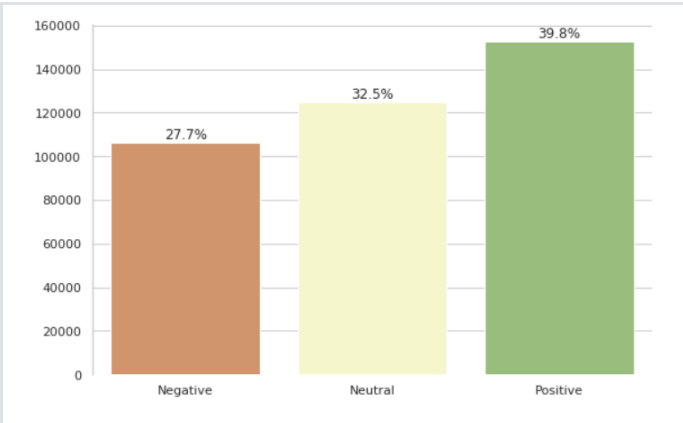
	film-url	rating	review	date	helpful	unhelpful
53894	http://www.allocine.fr/film/fichefilm-111632/critiques/spectateurs	1.0	Sans grande surprise, le premier opus de la franchise (il en existe trois actuellement) est dépourvu d'originalité et d'un quelconque intérêt. Sexy Dance (2006) est ultra conventionnel, prévisible au bout de dix minutes et pour un film qui avoisine les 105 minutes, il y en a clairement 45 en trop, entre les répétitions inutiles, les séquences musicales et de danses à rallonges, les répliques mièvres, les disputes et les réconciliations, bref, une sortie en DTV aurait largement suffit ! Espérons que le second volet (2008) et le troisième (2010) seront relever le niveau.	2010-09-21	1	1
366314	http://www.allocine.fr/film/fichefilm-209558/critiques/spectateurs	3.5	enfin un téléfilm Gulli qui traite des vacances ! un téléfilm très sympathique, bien joué et divertissant pour les moins de 12 ans.	2014-11-14	1	2
14939	http://www.allocine.fr/film/fichefilm-109788/critiques/spectateurs	5.0	En voyant la multiplication de critiques négatives envers ce monument du cinéma modernes, j'ai re-regardé There Will Be Blood une troisième fois. Et le verdict est toujours le même, on est toujours face à une leçon magistrale de cinéma. Les acteurs sont plus qu'excellents (Day-Lewis comme à son habitude fait exploser le talentomètre, mais aussi le sous-estimé Paul Dano). Les dialogues sont écrits à la perfection. La réalisation est irréprochable, offre certains plans très beaux plans mais arrive à asseoir une ambiance angoissante et poisseuse, appuyé par la musique qui déroute au début mais arrive à nous influencer inconsciemment et instaurer chez le spectateur un profond malaise à certains moments (les dangers du métier de mineur dans une exploitation pétrolière sont parfaitement retranscris à l'écran). Cette ambiance sert une histoire passionnante, celle d'un exploitant misanthrope (de plus en plus détestable) sombrant dans la folie et qui ne cherche qu'une chose: être assez riche pour pouvoir vivre coupé du monde. Tout est parfait, rien que la scène de fin (grand moment, interrogation sur la foi religieuse, le dévoilement de la vraie nature des personnages principaux, le milk-shake, etc) devrait suffire nous faire aimer There Will Be Blood, ou au moins lui reconnaître ses indéniables qualités.	2017-01-11	3	0

Polarity coding:

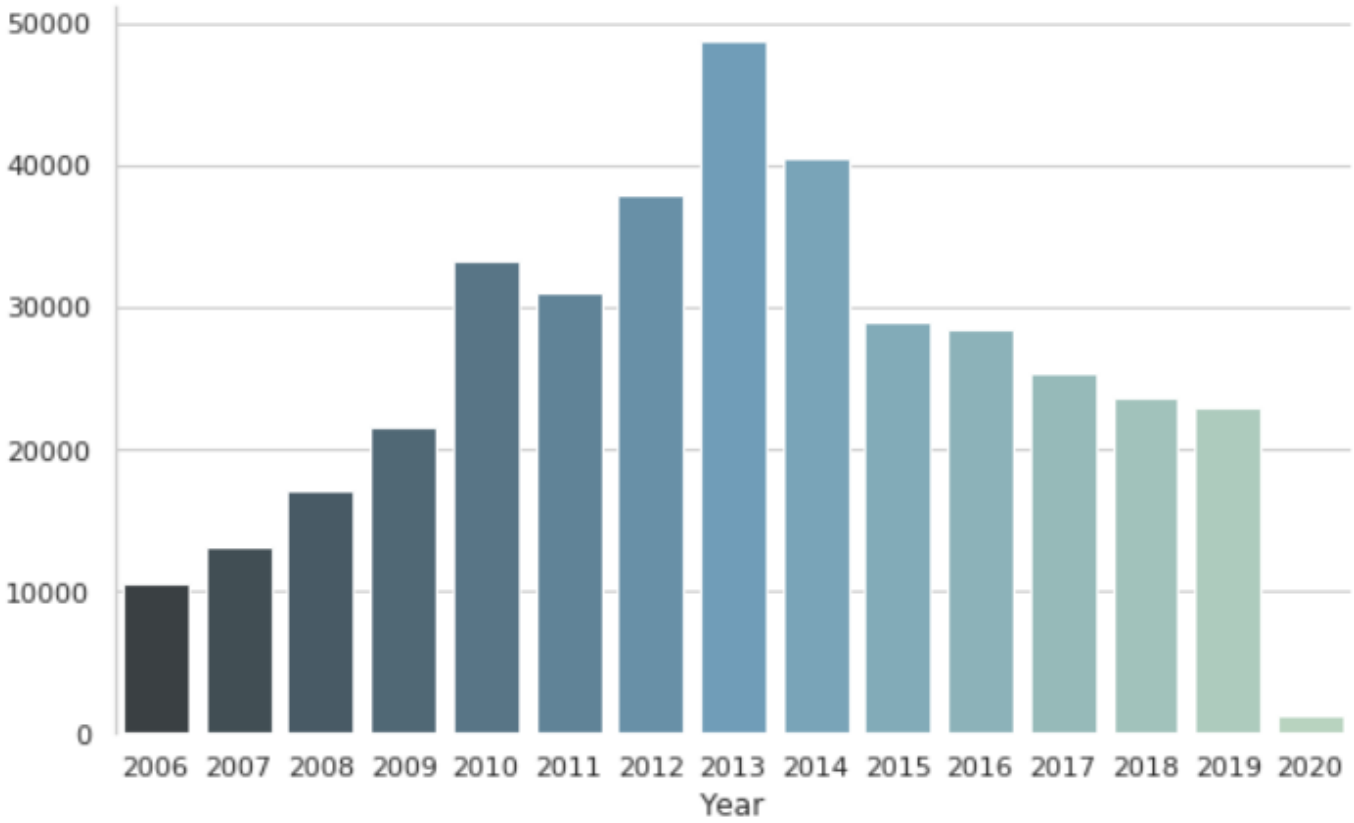
- 1: positive: 4-5점
- 0: neutral: 2점 초과 4점 미만
- -1: negative: 0-2점

평점 별 리뷰 분표



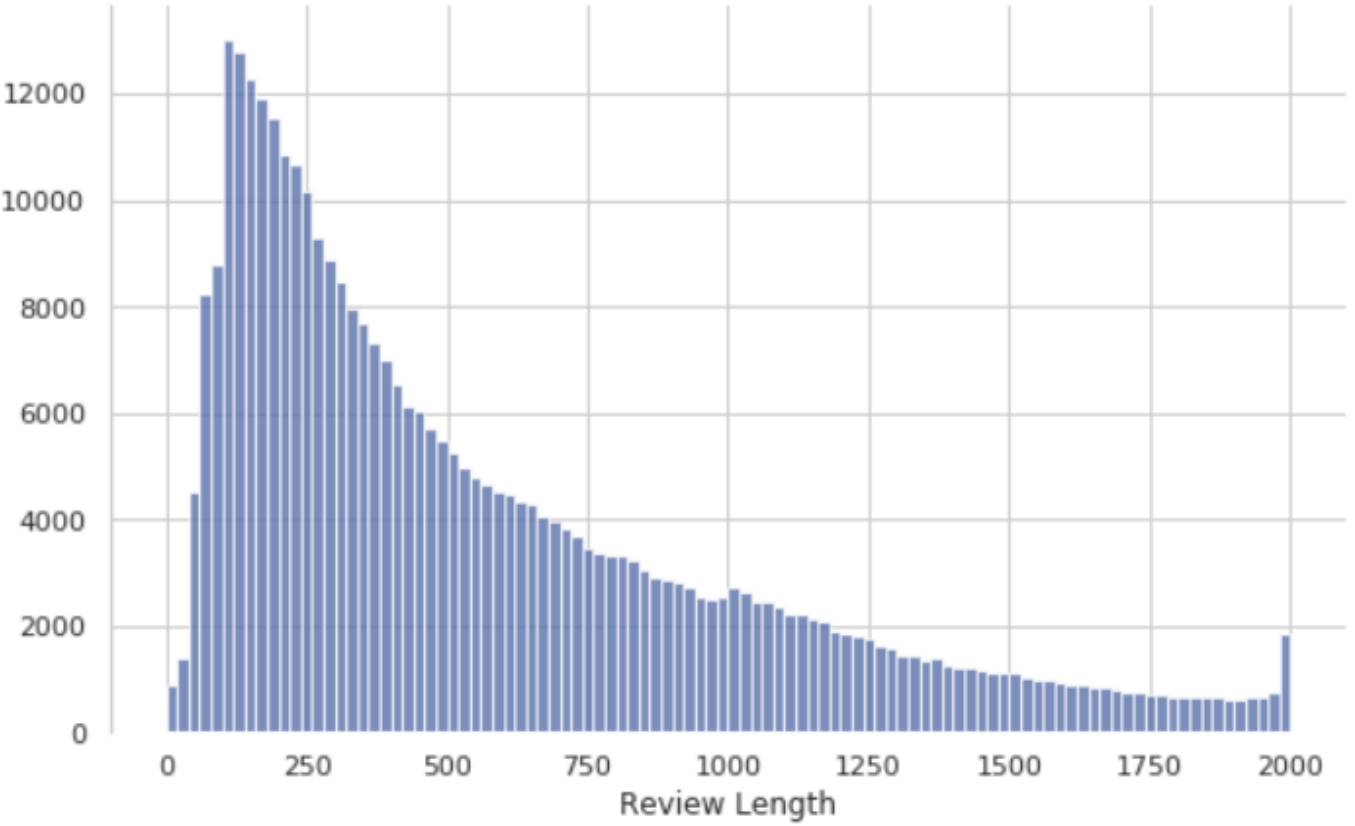


년도별 리뷰 개수



리뷰 길이 별 분포

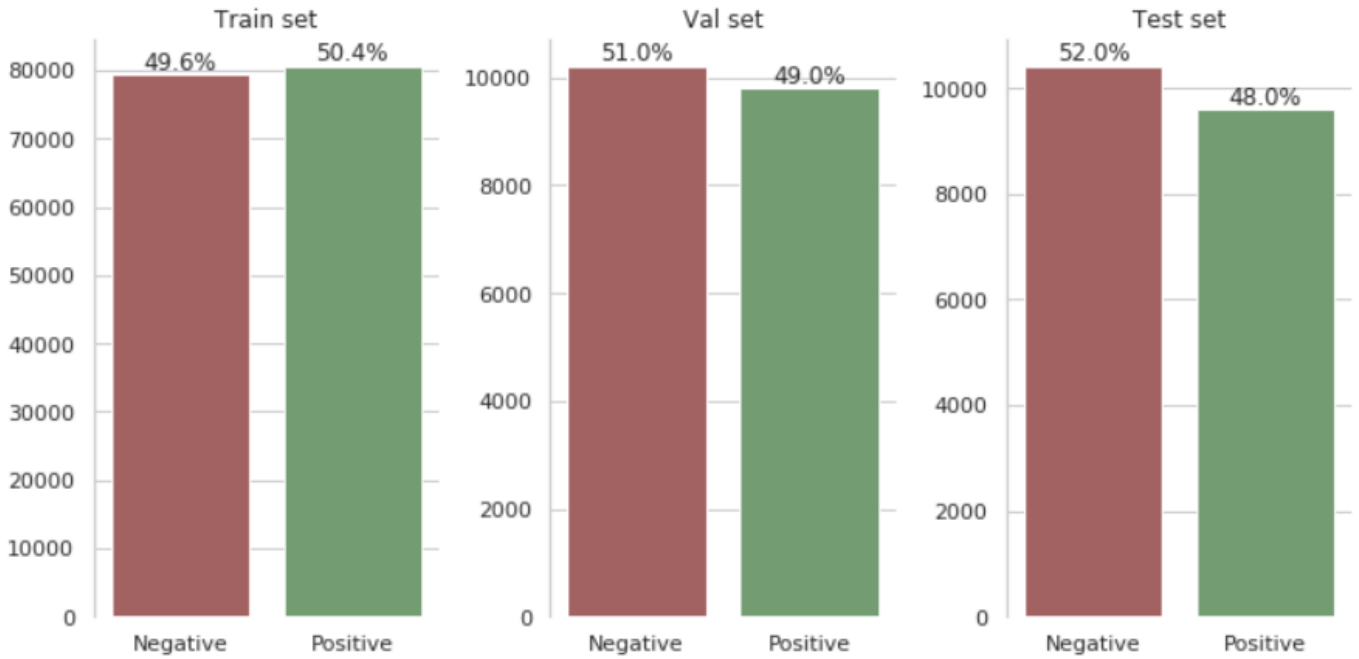
- 2000자 이상으로 구성되어 있는 리뷰(전체 데이터의 약 6%)는 잘라냄.



```python
def loss_percentage(df, previous_length):
    new_length = len(df)
    percentage = 100*(1-(new_length/previous_length))
    return new_length, percentage

# 1. 필요없는 열 삭제
dataset_df = dataset_df.drop(columns=['rating', 'date', 'helpful', 'unhelpful'])

# 2. 필요없는 행 삭제 (polarity가 neutral한 행들)
dataset_df = dataset_df[dataset_df['polarity'] != 0]
length, percentage = loss_percentage(dataset_df, initial_len)
print("Length: {} (-{:.1f} %)".format(length, percentage))

# 3. 2000자 넘는 리뷰 삭제
LENGTH_THRESH = 2000
dataset_df = dataset_df[dataset_df['review'].str.len() <= LENGTH_THRESH]
length, percentage = loss_percentage(dataset_df, length)
print("Length: {} (-{:.1f} %)".format(length, percentage))

# 4. 각 영화별 리뷰 30개로 제한
MAX_REVIEWS_PER_FILM = 30
grouped = dataset_df.groupby('film-url')
for ids in grouped.groups.values():
    num_reviews = len(ids)
    if num_reviews > MAX_REVIEWS_PER_FILM:
        sampling_size = num_reviews - MAX_REVIEWS_PER_FILM
        ids_to_drop = random.sample(list(ids), sampling_size)
        dataset_df = dataset_df.drop(ids_to_drop)

length, percentage = loss_percentage(dataset_df, length)
print("Length: {} (-{:.1f} %)".format(length, percentage))

length, percentage = loss_percentage(dataset_df, initial_len)
print("Total loss: (-{:.1f} %)".format(percentage))
```

그 밖의 전처리

- negative한 리뷰를 -1에서 0으로 코딩
- 텍스트 클리닝 (공백, 줄 바꾸기 제거 등.)
- 비어있는 리뷰 삭제
- 긍정 & 부정 리뷰 개수 맞추기



2. TF-IDF를 이용한 감정분류 머신러닝 모델 구현

데이터 사이즈

```
import pickle

PICKLE_PATH = "C:/Users/sungw/Downloads/data/data/allocine_dataset.pickle"

with open(PICKLE_PATH, 'rb') as reader:
    data = pickle.load(reader)

X_train, y_train = np.array(data["train_set"]['review']), np.array(data["train_set"]['polarity'])
X_val, y_val = np.array(data["val_set"]['review']), np.array(data["val_set"]['polarity'])
X_test, y_test = np.array(data["test_set"]['review']), np.array(data["test_set"]['polarity'])
class_names = data['class_names']

print("LEN TRAIN: "+ str(len(X_train)))
print("LEN VAL: "+ str(len(X_val)))
print("LEN TEST: "+ str(len(X_test)))
```

```
LEN TRAIN: 160000
LEN VAL: 20000
LEN TEST: 20000
```

기본 모델

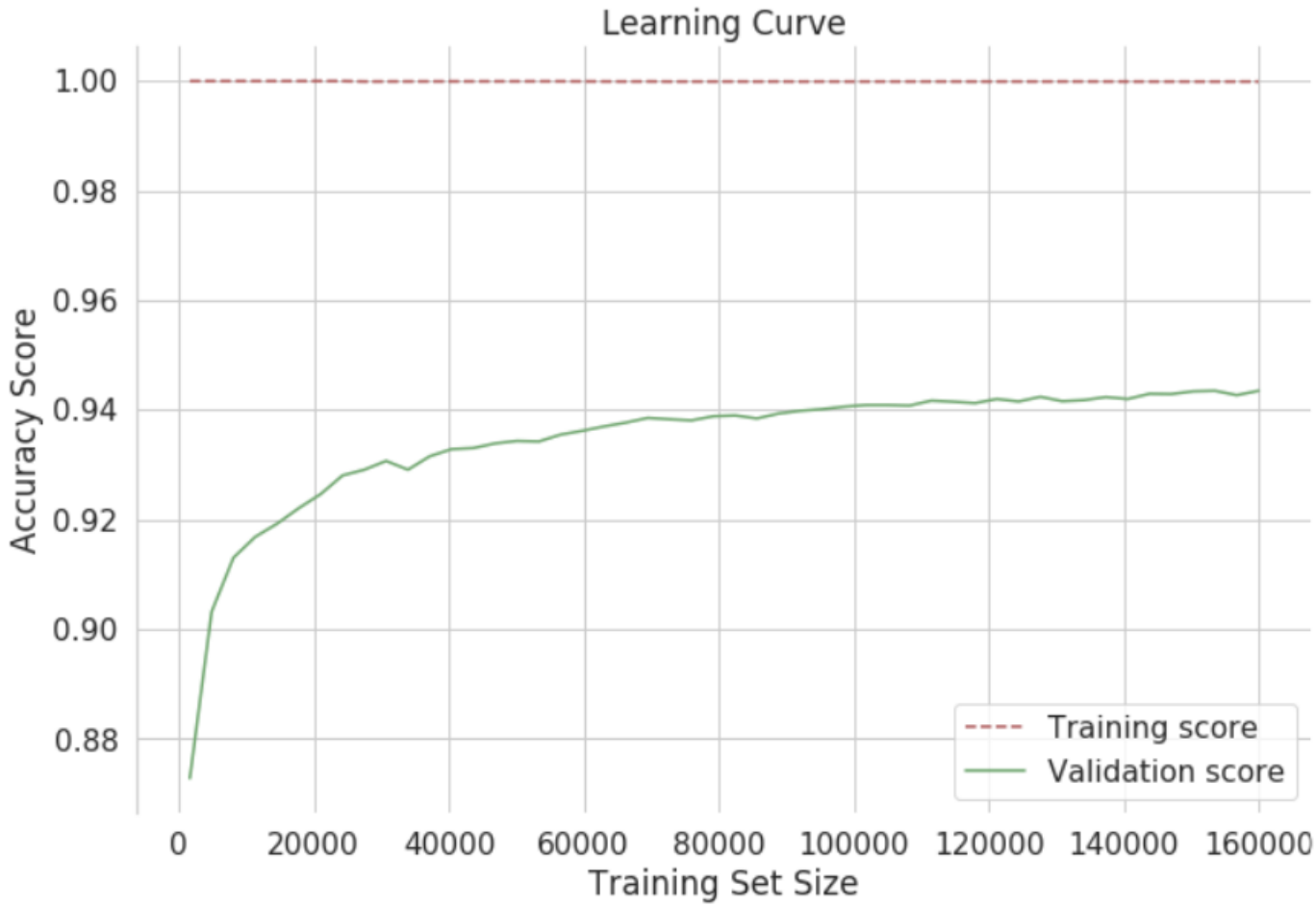
```python
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline

tfidf_clf = Pipeline([
    ('tfidf', TfidfVectorizer()),
    ('clf', LogisticRegression(n_jobs=-1, verbose=1)),
])

tfidf_clf.fit(X_train, y_train)
```

결과

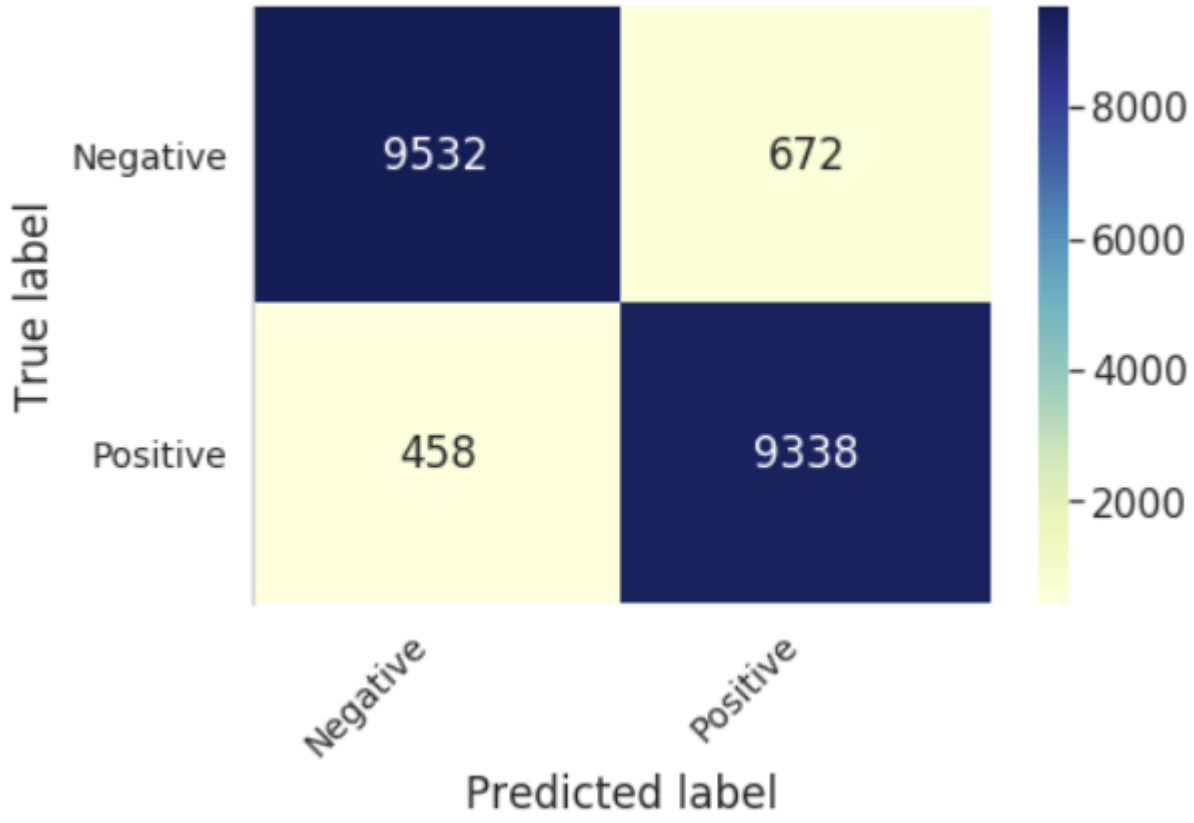
```python
from sklearn import metrics

y_pred = best_clf.predict(X_val)

print("Val Accuracy: {:.2f}".format(100 * metrics.accuracy_score(y_val, y_pred)))
print("Val F1-Score: {:.2f}".format(100 * metrics.f1_score(y_val, y_pred)))
print()

report = metrics.classification_report(
    y_val, y_pred,
    target_names=class_names.values()
)
print(report)
```

```
Val Accuracy: 94.35
Val F1-Score: 94.29

              precision    recall  f1-score   support

    Negative       0.95      0.93      0.94     10204
    Positive       0.93      0.95      0.94      9796

    accuracy                           0.94     20000
   macro avg       0.94      0.94      0.94     20000
weighted avg       0.94      0.94      0.94     20000
```

Learning curve



Confusion matrix를 통한 오차 분석



오차 분석에서 잘못 예측한 데이터 기반 말풍선 시각화

```
false_pos = X_val[(y_val == 0) & (y_pred == 1)]
false_neg = X_val[(y_val == 1) & (y_pred == 0)]
```