Naïve Bayes Binary Classification for Film Review

Iwa Ovyawan Herlistiono[1], Sriyani Violina[2]

[1,2] Informatics Department, Engineering Faculty, Widyatama University
Jalan Cikutra No 204A Bandung

Email : ovyawan.herlistiono@widyatama.ac.id

Abstract

Online streaming services provide thousands of movie collections that can be watched by customers, viewers usually choose films based on reviews and ratings. To save time in the selection of good programs, there needs to be a tool to classify the various reviews available to choose which films are worth watching. Classification is an important topic in machine learning and data mining. In this study we use the naïve bayes algorithm which is one of the most efficient and effective algorithms for classification. The data set used in this study is the IMDB film review data taken from [4], as many as 50000 data that has been given 2 types of classification labels, namely "negative" and "positive".It can be concluded that the accuracy of the Naïve Bayes algorithm by applying the Multinomial and Beroulli methods is able to classify film reviews well in all test scenarios with the best accuracy achieved is 84%.

Keywords: naïve bayes, Gaussian Naïve Bayes, film review classification

1. Introduction

The covid-19 global pandemic caused many changes in human behavior both in the world of work and daily life. The term "new normal" relates to new habits that must be adapted to the demands of the changes that occur. Society must adapt to "new normal"

New habits cover various aspects of life. Work that was done at the office, now a lot is done at home. Awareness of cleanliness such as the use of masks, washing hands with soap, the use of hand sanitizers and consumption of vitamins and exercise to maintain endurance are also increased. This change of habit also applies when looking for entertainment (leisure), before the pandemic, the entertainment carried out usually meets with many people and public places. At present this type of entertainment should be avoided to cut off the spread of covid-19.

One type of entertainment that must be avoided is watching films to the cinema, Cinema is one of the places that are prone to transmission, so the government closes all cinemas during this pandemic. An alternative entertainment for film lovers is watching at home by using television and online streaming services. At present the online streaming service provided in Indonesia has become a popular trend for people's entertainment needs. Netflix, an online streaming service, announced an increase of up to 15.8 million paid subscribers in the first quarter of 2020. So that as a whole to 182.9 million paid customers [1].

Online streaming services provide thousands of movie collections that can be watched by customers, viewers usually choose films based on reviews and ratings. To save time in the selection of good programs, there needs to be a tool to classify the various reviews available to choose which films are worth watching. Classification is an important topic in machine learning and data mining. There are many algorithms that have been developed for classification. In this study we use the naïve bayes algorithm which is one of the most efficient and effective algorithms for classification.

2. Research Method

2.1 Naïve Bayes

Naïve Bayes is a collection of supervised learning algorithms that apply the Bayes theorem with the assumption of independence between each pair of features given a class variable value.
The Bayes theorem can be formulated as follows, given the value of the variable class y and the dependent features x_1 to x_n:

$$P(y \mid x_1, \dots, x_n) = \frac{P(y) P(x_1, \dots, x_n \mid y)}{P(x_1, \dots, x_n)}$$

the naive conditional independence assume that: :

$$P(x_i | y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i | y)$$

for all *I*, simplified to:

$$P(y \mid x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \dots, x_n)}$$

Since *P(x1,..., xn)* is constant given the input, then following classification rule applied:

$$P(y \mid x_1, \dots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y),$$ [1]

2.1.1 **Gaussian Naïve Bayes**

This Method implement the Gaussian likelihood of the features to the native Naïve Bayes classification algorithm. Hence:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

A maximum likelihood is used to estimate the σy and μy parameters. The following snippet code shows that.

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
X, y = load_iris(return_X_y = True)
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                   test_size=0.5,
                                               random_state=0)
gnb = GaussianNB()
y_pred = gnb.fit(X_train, y_train).predict(X_test)
print("Number of mislabeled points out of a total %d points : %d"
       % (X_test.shape[0], (y_test != y_pred).sum()))
```

The above snippet would yield an output:

```
Number of mislabeled points out of a total 75 points : 4
```
[2]

2.1.2 **Multinomial Naïve Bayes**.

The second method, implements multinomial distributed data to the original Naïve Bayes Classification Algorithm. The distribution is determined by the vector θy = (θy1, ..., θyn) for each class y, where n is the vocabulary size and θyi is the probability of the feature P (xi | y) that appears in the sample class y. Whereas the θy parameter is filled with an approach to maximum similarity:

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

Dengan:

$N_{yi} = \sum_{x \in T} x_i$ is the number of times feature *I* appears in a sample of class *y* in the training set *T*, and

$N_y = \sum_{i=1}^{n} N_{yi}$ is the total count of all features for class *y*.

2.1.3 **Bernoulli Naïve Bayes**

The third method, Bernoulli Naïve Bayes, implements training and Naïve Bayes classification algorithms for data distributed according to the Bernoulli multivariate distribution. The decision rules for Bernoulli Naïve Bayes are based on :

$$P(x_i \mid y) = P(i \mid y) x_i + (1 - P(i \mid y))(1 - x_i)$$

it penalizes the non-occurrence of a feature *i*, an indicator for class *y*. while the multinomial variant would just ignore it. [3]

2.2 Data Set

The data set used in this study is the IMDB film review data taken from [4], as many as 50000 data that has been given 2 types of classification labels, namely "negative" and "positive". The data sample can be seen in Table 1.

Table 1
Film Review Data Sample

No	Review	Sentiment
1.	One of the other reviewers has mentioned that after watching just 1 Oz episode you'll be hooked. They are right, as this is exactly what happened with me.<br /><br />The first thing that struck me about Oz was its brutality and unflinching scenes of violence, which set in right from the word GO. Trust me, this is not a show for the faint hearted or timid. This show pulls	positive

No	Review	Sentiment
	no punches with regards to drugs, sex or violence. Its is hardcore, in the classic use of the word.<br />
It is called OZ as that is the nickname given to the Oswald Maximum Security State Penitentary. It focuses mainly on Emerald City, an experimental section of the prison where all the cells have glass fronts and face inwards, so privacy is not high on the agenda. Em City is home to many..Aryans, Muslims, gangstas, Latinos, Christians, Italians, Irish and more....so scuffles, death stares, dodgy dealings and shady agreements are never far away.

I would say the main appeal of the show is due to the fact that it goes where other shows wouldn't dare. Forget pretty pictures painted for mainstream audiences, forget charm, forget romance...OZ doesn't mess around. The first episode I ever saw struck me as so nasty it was surreal, I couldn't say I was ready for it, but as I watched more, I developed a taste for Oz, and got accustomed to the high levels of graphic violence. Not just violence, but injustice (crooked guards who'll be sold out for a nickel, inmates who'll kill on order and get away with it, well mannered, middle class inmates being turned into prison bitches due to their lack of street skills or prison experience) Watching Oz, you may become comfortable with what is uncomfortable viewing....thats if you can get in touch with your darker side.	
2.	A wonderful little production.

The filming technique is very unassuming- very old-time-BBC fashion and gives a comforting, and sometimes discomforting, sense of realism to the entire piece.

The actors are extremely well chosen- Michael Sheen not only "has got all the polari" but he has all the voices down pat too! You can truly see the seamless editing guided by the references to Williams' diary entries, not only is it well worth the watching but it is a terrificly written and performed piece. A masterful production about one of the great master's of comedy and his life.

The realism really comes home with the little things: the fantasy of the guard which, rather than use the traditional 'dream' techniques remains solid then disappears. It plays on our knowledge and our senses, particularly with the scenes concerning Orton and Halliwell and the sets (particularly of their flat with Halliwell's murals decorating every surface) are terribly well done.	positive
3.	I thought this was a wonderful way to spend time on a too hot summer weekend, sitting in the air conditioned theater and watching a light-hearted comedy. The plot is simplistic, but the dialogue is witty and the characters are likable (even the well bread suspected serial killer). While some may be disappointed when they realize this is not Match Point 2: Risk Addiction, I thought it was proof that Woody Allen is still fully in control of the style many of us have grown to love.

This was the most I'd laughed at one of Woody's comedies in years (dare I say a decade?). While I've never been impressed with Scarlet Johanson, in this she managed to tone down her "sexy" image and jumped right into a average, but spirited young woman.

This may not be the crown jewel of his career, but it was wittier than "Devil Wears Prada" and more interesting than "Superman" a great comedy to go see with friends.	positive
4.	Basically there's a family where a little boy (Jake) thinks there's a zombie in his closet & his parents are fighting all the time.

This movie is slower than a soap opera... and suddenly, Jake decides to become Rambo and kill the zombie.

OK, first of all when you're going to make a film you must Decide if its a thriller or a drama! As a drama the movie is watchable. Parents are divorcing & arguing like in real life. And then we have Jake with his closet which totally ruins all the film! I expected to see a BOOGEYMAN similar movie, and instead i watched a drama with some meaningless thriller spots.

3 out of 10 just for the well playing parents & descent dialogs. As for the shots with Jake: just ignore them.	negative

No	Review	Sentiment
5.	Petter Mattei's "Love in the Time of Money" is a visually stunning film to watch. Mr. Mattei offers us a vivid portrait about human relations. This is a movie that seems to be telling us what money, power and success do to people in the different situations we encounter. <br /><br />This being a variation on the Arthur Schnitzler's play about the same theme, the director transfers the action to the present time New York where all these different characters meet and connect. Each one is connected in one way, or another to the next person, but no one seems to know the previous point of contact. Stylishly, the film has a sophisticated luxurious look. We are taken to see how these people live and the world they live in their own habitat.<br /><br />The only thing one gets out of all these souls in the picture is the different stages of loneliness each one inhabits. A big city is not exactly the best place in which human relations find sincere fulfillment, as one discerns is the case with most of the people we encounter.<br /><br />The acting is good under Mr. Mattei's direction. Steve Buscemi, Rosario Dawson, Carol Kane, Michael Imperioli, Adrian Grenier, and the rest of the talented cast, make these characters come alive.<br /><br />We wish Mr. Mattei good luck and await anxiously for his next work.	positive

2.3 Data Preprocessing

Just by looking to the contents of the dataset, we immediately realize that we need to apply some preprocessing to the data set. The first one that comes to mind was, to remove all of the HTML Tags, the special characters, and all of the stop words. By the second look, we also find that we need to tokenize, force to lowercase all of the words and to replace the classification class in the dataset. And we did exactly that.

2.3.1 Replacing Sentiment Class

In our discussion, for easier reading we choose to replace all the sentiment classes by assigning “1” instead of “positive”, and “0” instead of “negative”. It is done by writing a snippet below:

```
# 1st preprocessing the dataset, replacing:
# "positive" with "1" and “negative” with “0”
data.sentiment.replace('positive', 1, inplace = True)
data.sentiment.replace('negative', 0, inplace = True)
data.head(10) #  print dataset  first 10
```

2.3.2 **The HTML Tags Removal**.

Next, would be the removal of all HTML Tags. Those are commonly known as the characters of “<”, “.”, “*”, “?” and “>” characters. We applied those characters to the regex rules as it can be seen in the snippet below.

```
# removing HTML tags
# regex rules: "<", ".", "*", "?", ">"
def hapus_HTML(text):
    cleaned = re.compile(r'<.*?>')
    return re.sub(cleaned, '', text)
```

2.3.3 The Special Characters Removal.

The special characters, basically, are the characters not available on the keyboard. Some of them are parts of the above HTML Tags removal leftover. Next, we applied a special characters removal to the dataset. Special characters removal snippet, can be seen below.

```
# Removing special characters
def spesial_karakter(text):
    rem = ''
    for i in text:
        if i.isalnum():
            rem = rem + i
        else:
            rem = rem + ' '
    return rem
```

2.3.4 The Force Lowercase.

Lowercasing is one of standard operation for text preprocessing. It is applicable to most text mining and NLP problems, mostly to avoid sparsity issue, and it can help in the case of not so big of data and it also helps the expected output consistency. The lowercasing function be seen below.

```
# convert to lowercase
def lowerisasi(text):
    return text.lower()
```

2.3.5 The Stop Words Removal and Tokenizer.

A stop word is a word that is so common that can be ignored (c.q. removed) without losing the meaning of said word. Example of this kind of words are the word “the”, “a”, “an”, “in”, etc. The idea of this removal of low information words from text, is so that we only focused on the important words instead.

Meanwhile, the tokenization is a process which splits longer strings of text into smaller pieces, called tokens. Token can be in a form of just one word or a smaller chunk of text as part of bigger text.

Those two processes, are also the part of standard text preprocessing. In below snippet, the stop word removal is directly followed by a tokenizer.

```
# stop-word removal and tokenizer
def tokenisasi(text):
   stop_words = set(stopwords.words('english'))
   words = word_tokenize(text)
   return [w for w in words if w not in stop_words]
```

2.3.6 The Stemming of the Words.

Stemming is a process that removes all prefixes from words back to their root word. For example: “running” would be stemmed into “run”, “invaluable” would be stemmed into “value”, and so on. In snippet below, we show the stemmer.

```
# Stemmer for words
def stemisasi(text):
    ss = SnowballStemmer('english')
    return " ".join([ss.stem(w) for w in text])
```

2.3.7 Bag of Words

A bag-of-words model (BoW) is a simple and flexible method to extract features from document or text for use in machine learning algorithms. In the BoW model, the structures, the grammar and the order information of the words in the document are ignored. It only deal with if a known words exist in the text or document rather than where the words reside in the text or document.

The snippet code below, show the creation of the Bag-of-Word.

```
# Create Bag of words
X  = np.array(data.iloc[:, 0].values)
y  = np.array(data.sentiment.values)
cv = CountVectorizer(max_features = 1000)
X  = cv.fit_transform(data.review).toarray()
```

2.4 Testing Scenario

The test is carried out with the aim to analyze the effect of the percentage of training data and testing data on accuracy. The test scenarios can be seen in table 2. The test scenarios will be applied to three methods, namely Gaussian, Multinomial and Bernouli.

Tabel 2 Testing Scenario

Scenario	% Training Data	% Testing Data
1	70	30
2	75	25
3	80	20
4	85	15
5	90	10

3. Result

After testing, the results of the test are as follows. The test results are described based on the test scenario that is the percentage of training data and testing data. Accuracy is defined as the percentage of prediction accuracy. The test results are illustrated in graphical form in Figure 1 to Figure 6.

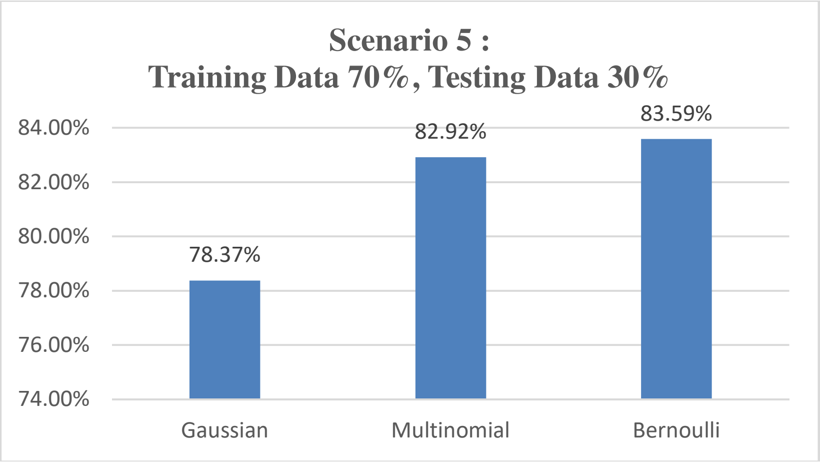

Figure 1 Accuracy Graph (Scenario 1)

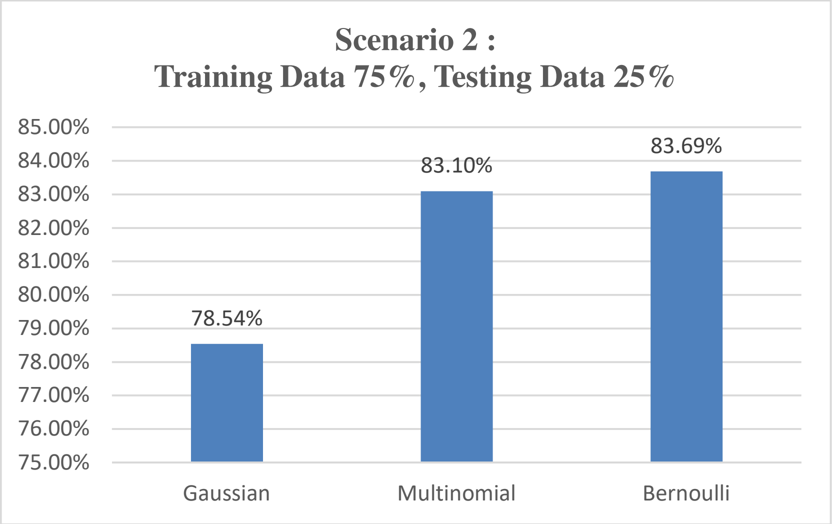

Figure 2 Accuracy Graph (Scenario 2)

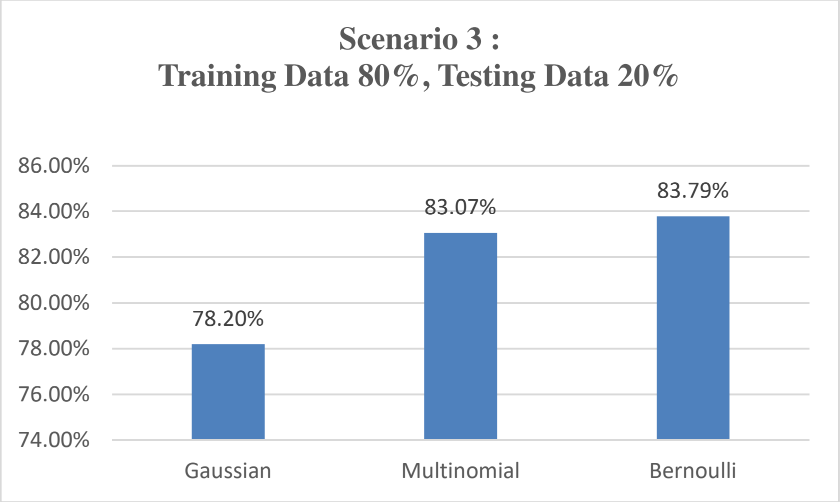

Figure 3 Accuracy Graph (Scenario 3)

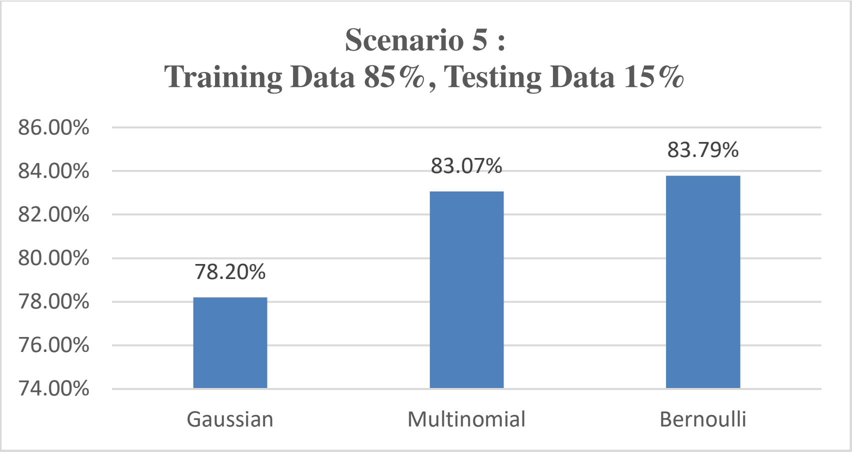

Figure 4 Accuracy Graph (Scenario 4)

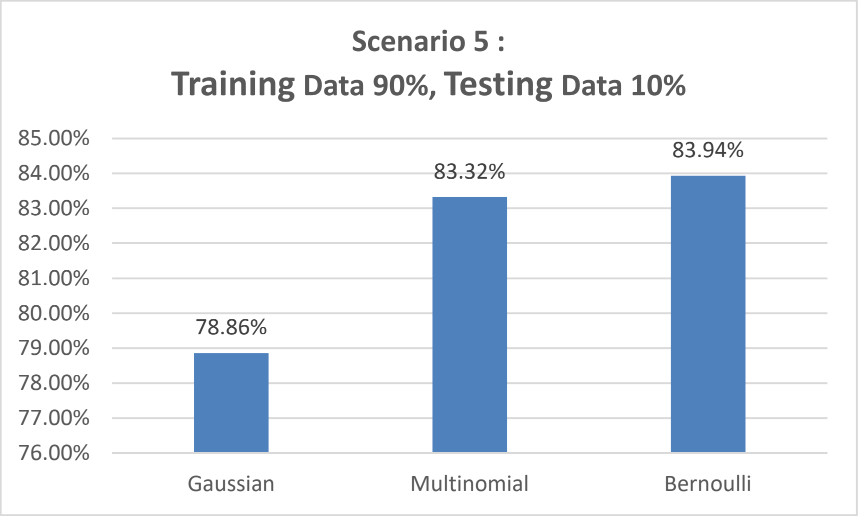

Figure 5 Accuracy Graph (Scenario 5)

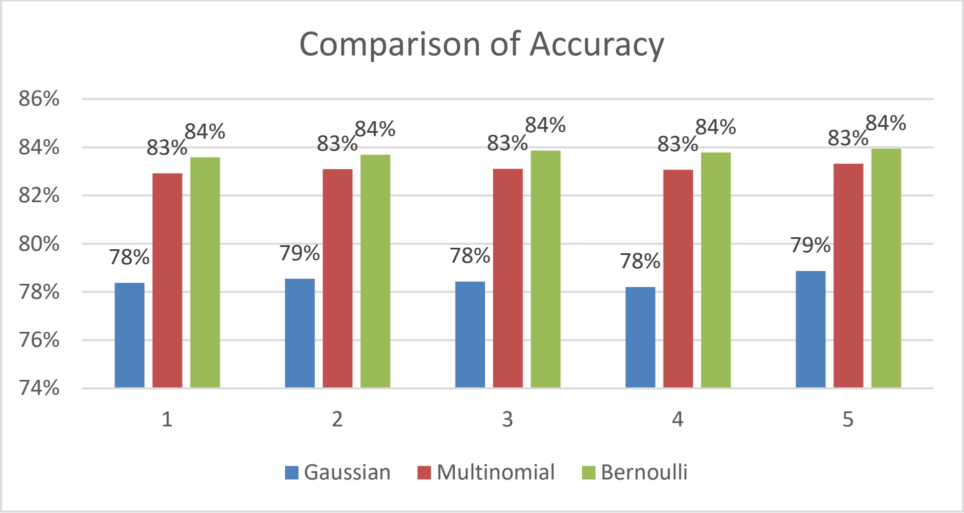

Figure 6 Comparison of Accuracy

4. Conclusions

After testing and analysis of the schemes that have been designed, it can be concluded that the accuracy of the Naïve Bayes algorithm by applying the Multinomial and Beroulli methods is able to classify film reviews well in all test scenarios with the best accuracy achieved is 84%. There was no significant result when seen in changes in the percentage of testing data and training data.

5. References

[1] H.Zhang, "The optimality of Naive Bayes", Proc. FLAIRS., https://www.cs.unb.ca/~hzhang/publications/FLAIRS04ZhangH.pdf, accessed 07/20/2020

[2] Pedregosa et al., "Scikit-learn: Machine Learning in Python", http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html, https://scikit-learn.org/stable/modules/naive_bayes.html, JMLR 12, pp. 2825-2830, 2011, accessed 07/20/2020.

[3] A. McCallum and K. Nigam, "A comparison of event models for Naive Bayes text classification.”, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.46.1529, accessed 07/20/2020

[4] ______, the IMDb review Dataset, https://thecleverprogrammer.com/wp-content/uploads/2020/05/IMDB-Dataset.csv, accessed 07/20/2020.