Classification of Conditions and Warranties in Real Estate Purchase and Sale Contracts With Naïve Bayes Algorithm

I. Legal Test Statement

Promises in contracts are divided into big promises called ***conditions*** or promissory conditions and little promises called ***warranties***. Conditions are fundamental to the contract, but warranties are less important. When a contractual term is breached, whether the term is a condition or a warranty determines which remedies are available.

Term	Nature	Remedy of breach
Condition	Fundamental to the contract	The innocent party can rescind the contract and sue for damage, or sue for specific performance
Warranty	Less important than conditions	The innocent party must close the deal and sue for damage

Although many contracts may have clarified which terms are conditions and which are warranties, such differentiation may not be recognized by the court. In *Smale v. Van der Weer*, the court ruled that "it is not the use of the word 'warranty' or 'condition' which determines the nature of the provision in the contract but rather the character of the provision".

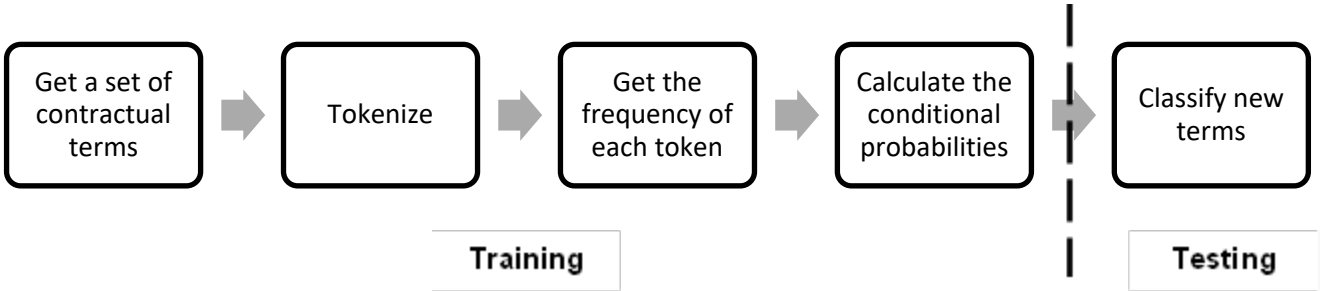
Moreover, the classification of some implied terms in the contract may be confusing and disputable. In *Jorian Properties Ltd v Zellenrath*, the Buyer was buying a five-plex but later discovered that the home was a legal triplex. The court's majority decision stated that the "five-plex" is a mere warranty and not a condition. In *Hatami v 1237144 Ontario Inc.* the court ruled that if the purchaser's ability to close the transaction is contingent upon a survey being provided, the term calling for a survey will likely be classified as a condition. Otherwise, it will likely be classified as a warranty.

Though many legal considerations are involved, previous decisions have shown that certain wording patterns of a contractual term may affect whether the court considers the term as a condition or a warranty. This plan attempts to differentiate conditions and warranties with text analysis in artificial intelligence.

II. Classification Model - Naive Bayes Algorithm

The differentiation of contractual terms between conditions and warranties is a binary text classification problem, which can be potentially solved by the Naive Bayes algorithm[1].

The Naive Bayes algorithm is based on conditional possibility and the Bayes' Rule. Simply put, the algorithm determines the categorization of a statement by comparing the probabilities of this statement belonging to either category. It generally comprises the following steps.



[1] Gutman, Alex J. in *Becoming a data head: How to think, speak, and understand data science, statistics, and Machine Learning*, ed (Wiley & Sons Canada, Limited, John, 2021) 164.

1. **Get a set of conditions and warranties**.

First, gather a series of contractual terms which have been categorized into conditions or warranties. Some of such contractual terms appearing in Ontario cases are shown below.

Text	Court's Classification
• "This sale is conditional for a period of 15 days from date of acceptance of same upon the Purchaser or his Agent being able to obtain a first mortgage in the amount of Ten Thousand Dollars…" (*Beauchamp v. Beauchamp*) • "All the development on Part Lot 7, Concession 3 in Township of Hamilton to be developed by Registered Plan of Subdivision in accordance with the Township of Hamilton applicable zoning regulations." (*Smale v. Van der Weer*)	Condition
• "They are the owners of the surface rights including all timber rights for the subject properties and the purchase price herein provided for includes all such rights" (*Phinny v Macaulay*) • "The premises contain 4 one bedroom apartments and 1 bachelor apartment being currently rented at the following rates" (*Jorian Properties Ltd v Zellenrath*)	Warranty

2. **Tokenize the text**

In analysis, we need to break down the text into individual words as the basic units or tokens. The process of tokenization include:

- Eliminates all punctuations,
- Eliminates stop words (such as "the", "a", and "this"),
- Lowercase all letters, and
- Stem the words[2].

Classes	Tokens (***Incomplete***)
Condition (*Beauchamp, Smale*)	{purchase, conditional, acceptance, obtain, mortgage, dollar, development, plan, regulation,...}
Warranty (*Phinny, Jorian*)	{purchase, surface, timber, right, bedroom, apartment, rent, ...}

3. **Get the frequencies of each word in each class**

After the tokenization, the user needs to find the frequency for the word in both classes respectively -- how many times a word appears in conditions and in warranties.

The frequencies need to be adjusted by adding 1 to avoid divided-by-zero problems in the following steps.

[2] Gutman, Alex J. in *Becoming a data head: How to think, speak, and understand data science, statistics, and Machine Learning*, ed (Wiley & Sons Canada, Limited, John, 2021) 156.

Tokens (*Incomplete*)	Frequency in Condition +1	Frequency in Warranty +1
purchase	1+1 = 2	1+1 = 2
mortgage	1+1 = 2	0+1 = 1
regulation	1+1 = 2	0+1 = 1
timber	0+1 = 1	1+1 = 2
apartment	0+1 = 1	2+1 = 3
Total	*8*	*9*

4. Calculate the conditional probability

The next step is to calculate the conditional probability of each token in both classes, which are $P(w_i|cond)$ and $P(w_i|warr)$. The calculation is to divide the frequency of a word by the total frequency of this class.

Tokens (*Incomplete*)	$P(w_i\|cond)$	$P(w_i\|warr)$
purchase	2/8 = 0.25	2/9 = 0.22
mortgage	0.25	0.11
regulation	0.25	0.11
timber	0.13	0.22
apartment	0.13	0.33
Total	*1*	*1*

5. Testing: Calculate the score

So far, the algorithm has been trained after the previous four steps. This step is about testing the algorithm by calculating a score related to the conditional probabilities calculated above. The mathematical formula for the score is as follows:

$$score = \frac{P(T_{\text{cond}})}{P(T_{\text{warr}})} \prod_{i=1}^{m} \frac{P(w_i|\text{cond})}{P(w_i|\text{warr})}$$

When score>1, it is a condition; when score<1, it is a warranty.

For example, if we have a contractual term reading "The **purchase** of this **apartment** is conditional of the acquisition of **mortgage** and subject to all applicable **regulations**". We can identify there are four keywords among our tokens, and we locate their corresponding probabilities and do the calculation as follows (Here we assume $P(T_{cond}) = P(T_{warr})$ since we have equal numbers of conditions and warranties in our training set):

$$score = \frac{P(\text{purchase}|\text{cond}) * P(\text{apartment}|\text{cond}) * P(\text{mortgage}|\text{cond}) * P(\text{regulation}|\text{cond})}{P(\text{purchase}|\text{warr}) * P(\text{apartment}|\text{warr}) * P(\text{mortgage}|\text{warr}) * P(\text{regulation}|\text{warr})}$$

Tokens	purchase	apartment	mortgage	regulation
P(w_i|cond)	0.25	0.13	0.25	0.25
P(w_i|warr)	0.22	0.33	0.11	0.11

$$score = \frac{0.25 * 0.13 * 0.25 * 0.25}{0.22 * 0.33 * 0.11 * 0.11} = 2.31 > 1$$

Since the score >1, we know it is likely to be a condition.

III. Collecting data

The data input of Naïve Bayes requires the text of contractual terms which have been categorized as a condition or a warranty by the court in Ontario. The ideal source of data will be the contractual texts in Ontario cases include *Jorian Properties Ltd v Zellenrath*, *Hatami v 1237144 Ontario Inc.*, *Beauchamp v. Beauchamp*, and *Smale v. Van der Weer*.

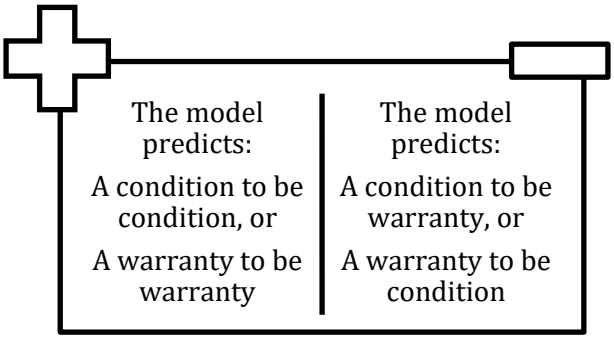
Texts in real court cases are representative of the courts' opinions on this topic, however, a disadvantage is that there may not be enough suitable cases to train the algorithm.

Alternatively, we can issue questionnaires to senior realtors or real estate lawyers to submit conditions and warranties from the real estate contracts they have handled with their clients' consent.

Questionnaires can generate a large amount of data, but the professional opinions may not be consistent with the courts' opinions.

IV. Test the model

The test of the model means the model should be able to make the correct judgement to classify a contractual term actual as a condition or a warrant rather than a mismatch. The test has two steps.



First, the model needs to pass the test with the testing set. Usually, the training set makes up 70% to 90% of the entire dataset, and the testing set makes up 30%-10%. We use the training set to train the model and use the testing set to test whether the model performs well.

Second, the model should be able to give accurate predictions in real cases. In this case, we need to find some new cases related to condition and warranty in real estate contracts and see if the model can correctly categorize them.

V. Significance and limitations

Considering the test of whether a contractual term is a condition or a warranty in the real estate context has not been fully established and may cause confusion, the Naive Bayes algorithm can potentially give a prediction of whether a contractual term is a condition or a warranty in the eyes of the court.

When the parties are composing and reviewing their real estate purchase and sale contracts, the algorithm will also provide suggestions about whether a term has the proper wording style to be categorized into condition or warranty as both parties agreed on.

Naive Bayes does not require a large amount of data to give a satisfactory result[3], which is suitable when there are not so much raw data like this case.

The limitation is that the algorithm only considers the wording of the term itself, excluding the context of the contract and other terms on the same contract. Context is important in differentiating conditions and warranties in practice. Therefore, the next step is to combine the context analysis, such as Topic Modeling, with Naive Bayes.

[3] https://theflyingmantis.medium.com/text-classification-in-nlp-naive-bayes-a606bf419f8c