Handout Sentiment Analysis of Android App Data

Sentiment Analysis Studio

Case Study: Android App Reviews (recall we have done text analytics on this data)

Data

We have created artificial data of 600 reviews by modifying and anonymizing actual customer reviews posted online. Of the 600 reviews, 500 are used for building models, and the remaining 100 are used for testing models. Raw textual data have been categorized into positive and negative groups based on 5-star numerical ratings given by a consumer on the review site at the time the review was written by the same consumer. Comments greater than or equal to 4 stars are considered as positive and less than or equal to 2 stars are considered as negative for the purpose of this case study. For modeling, we have two directories (folders) for sentiment mining as described below.

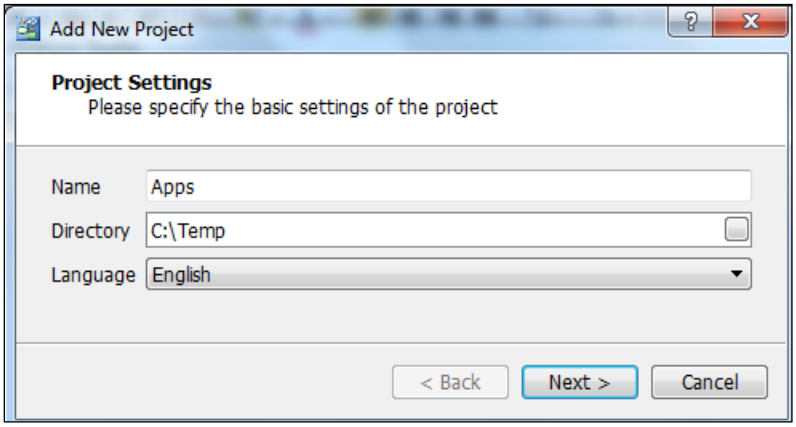
Sentiment Mining Model Data Folder (APP_SM):

- Positive reviews in the subfolder: APP_SM\model\pos
- Negative reviews in subfolder: APP_SM\model\neg

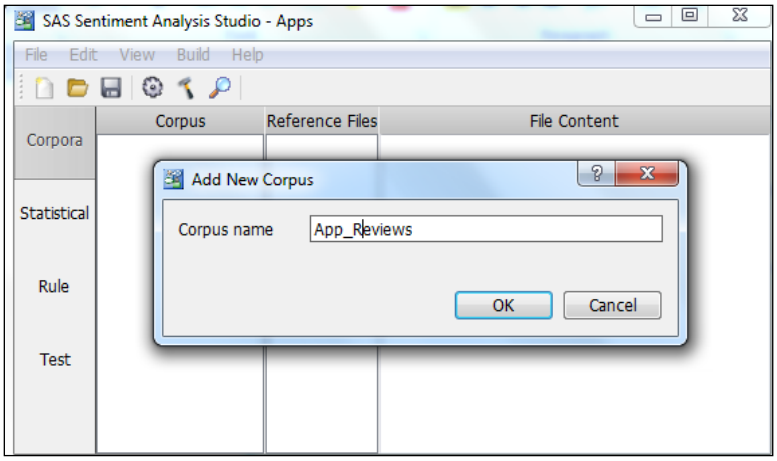
For testing sentiment mining models, we have two directories.

- Positive reviews in subfolder: APP_SM\test\pos
- Negative reviews in subfolder: APP_SM\test\neg

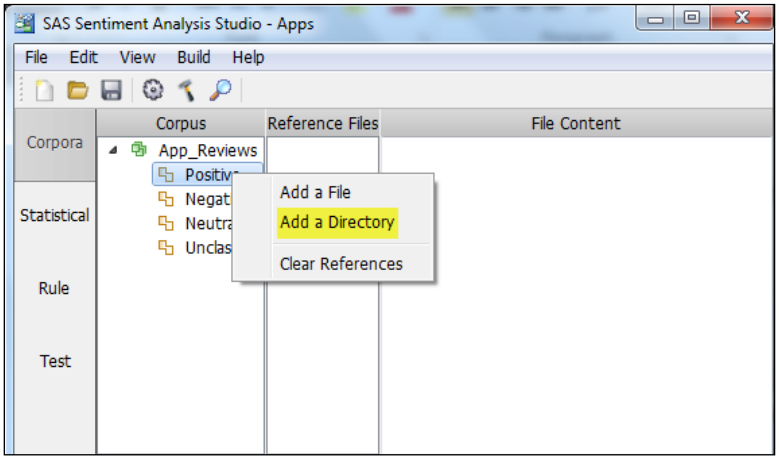
1. Open SAS Sentiment Analysis Studio.
2. Start a new project by selecting **File** ⇨ **New**. Select an appropriate name and a path for your project. Click **Next**.



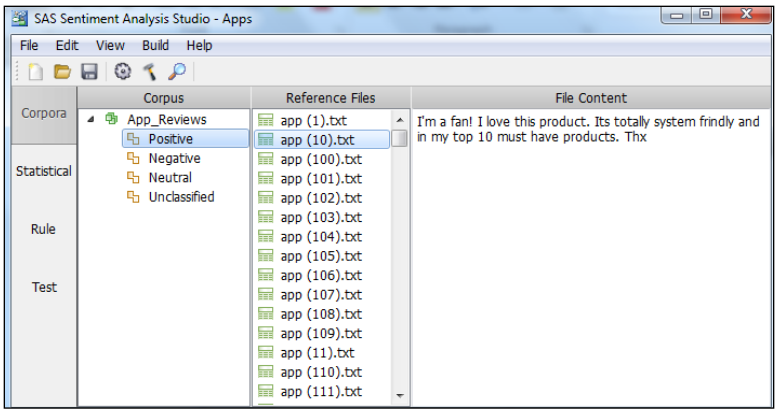
3. Accept default settings for a rule-based model. Click **Next**.
4. Accept default settings for the statistical model. Click **Next**.
5. Click **Finish** to complete project creation.
6. In the Corpus white workspace, right-click and select **New Corpus**. In the pop-up box, enter a name such as **App_Reviews**. Click **OK**.



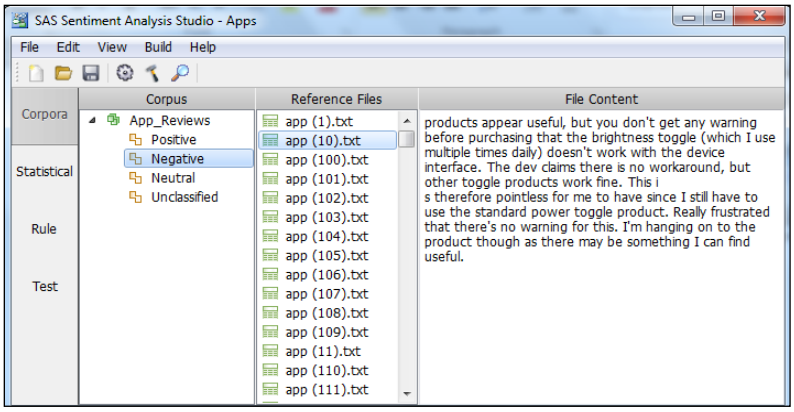
7. Right-click **Positive** and select **Add a Directory**.



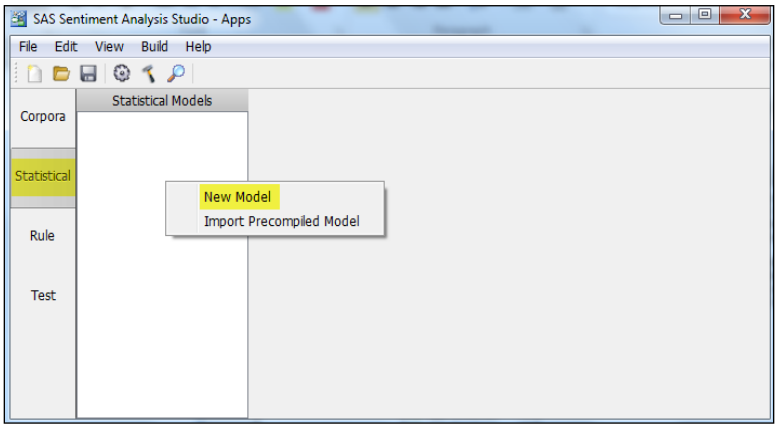
8. Go to the folder (APP_SM\model\pos) where the files for modeling the positive sentiments are stored and click **Select Folder**. You find that 250 positive reviews (text files) are imported and listed in the Reference Files workspace. To see any particular file (such as app(10).txt), click and select it in the Reference Files space.



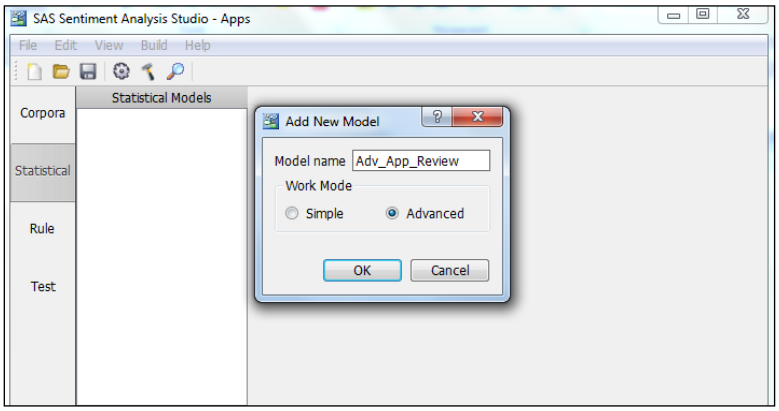
9. Follow similar steps to add to the Negative directory folder (APP_SM\model\neg) where files for modeling the negative sentiments are stored. We do not have “Neutral” and “Unclassified” documents in this case study, so leave those blank.



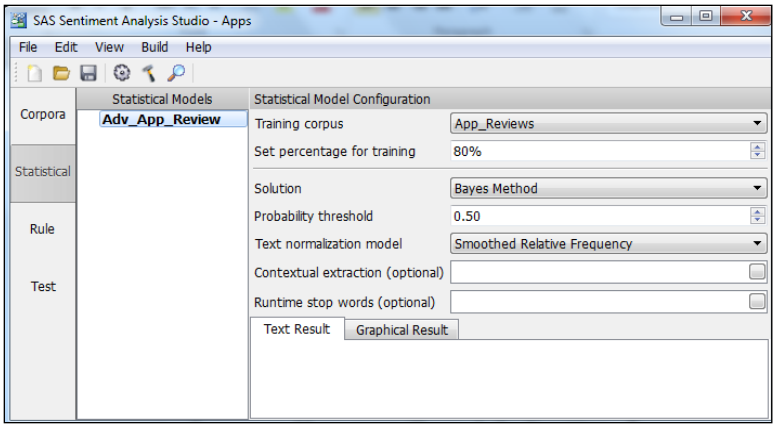
10. Click **Statistical** in the left panel. Right-click in the Statistical Models white workspace and select **New Model**.



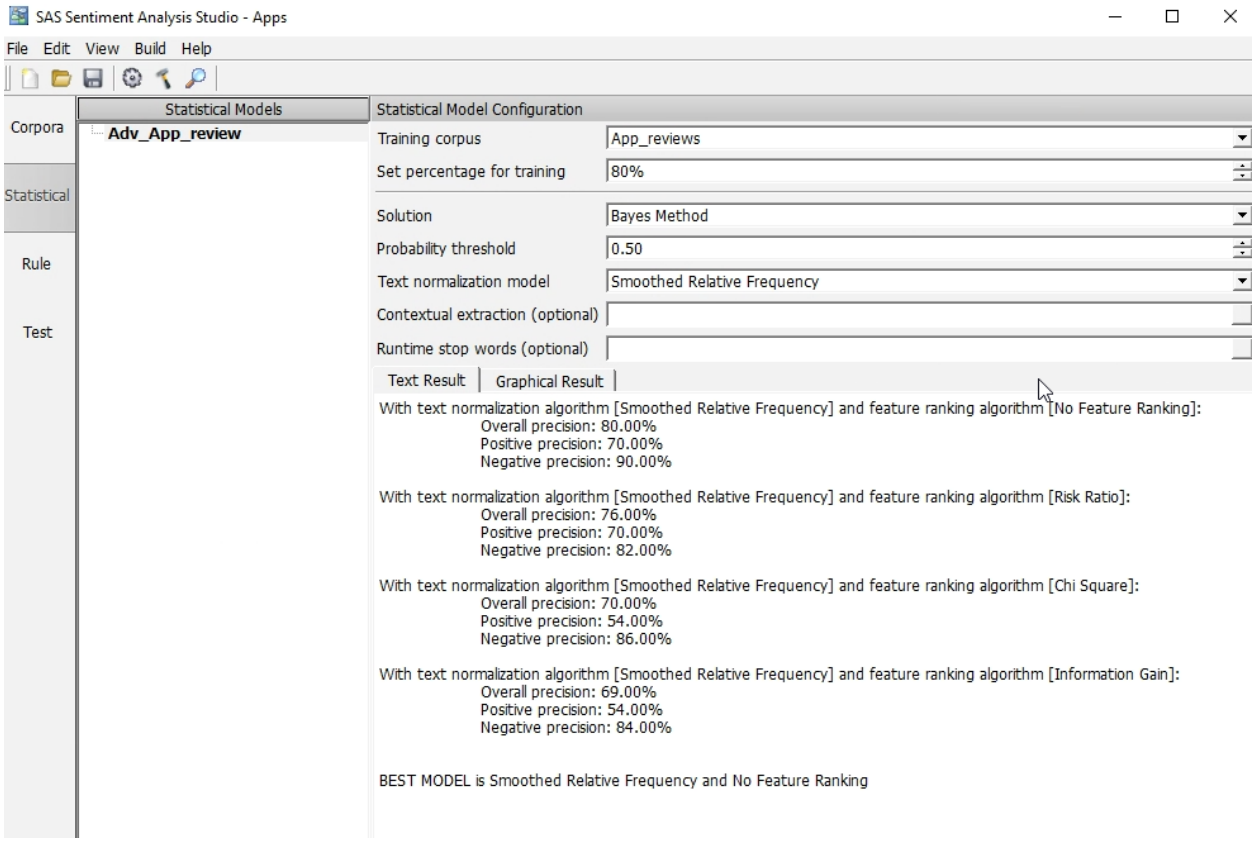
11. Name the new model (**Adv_App_Review** is used below), select **Advanced**, and click **OK**.



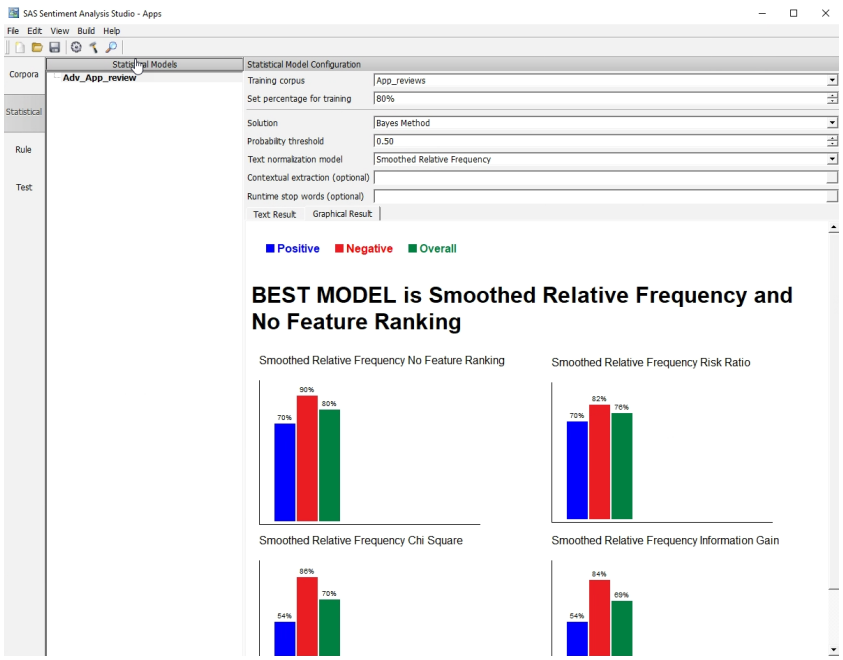
12. In the Statistical Model Configuration panel (below), you can make changes to improve your statistical model performance. Here, we use default settings:



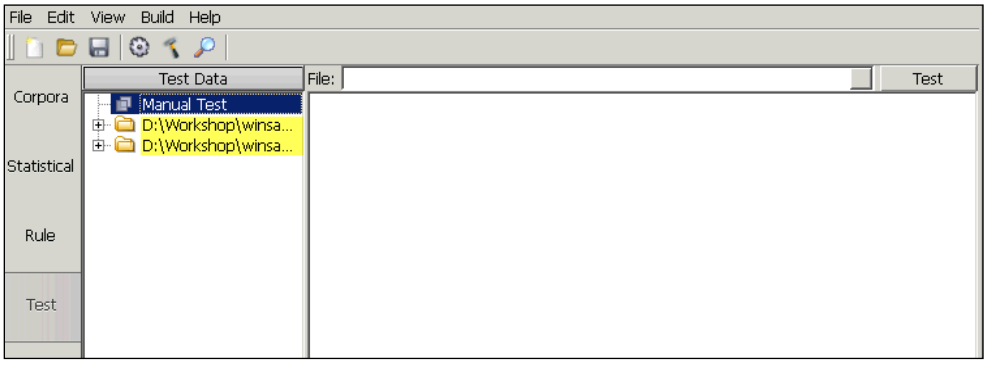
13. From the top menu, click **Build** ⇨ **Build Statistical Model**. Select the name of the model that you just created and click **OK** to run the model. Examine the results.



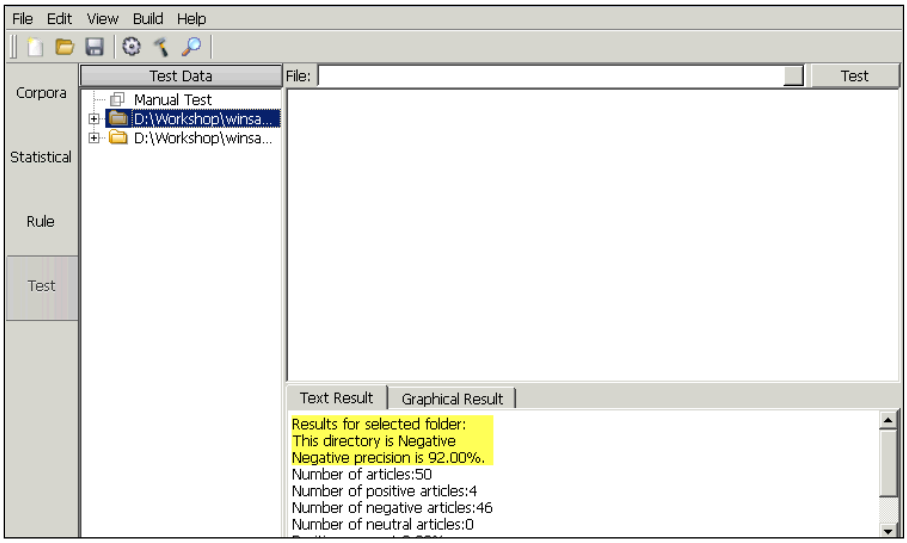
14. Click the **Graphical Result** tab to see bar charts of different metrics.

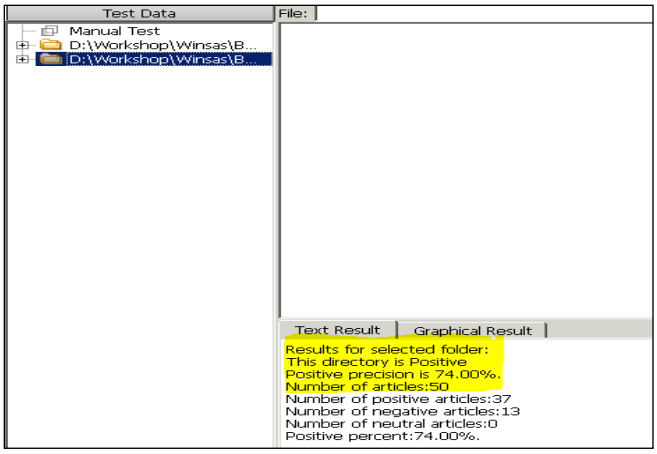


15. In this case study, one of the reasons to build a statistical model is to gain a list of the most often used words and terms so that we can import this list to a rule-based model. From the result, you can see that the overall accuracy of the statistical model is reasonable but perhaps can be improved by playing with options in statistical models. We compare the accuracy of both statistical models and rule-based models later in this demonstration.
16. Click **Test** in the left panel. Right-click in the Test Data white workspace and select **New Test Directory**. Browse to the test folders for sentiment mining data and select the **neg** folder (APP_SM\test\neg).
17. Repeat the step above for selecting the **pos** folder in the test directory.

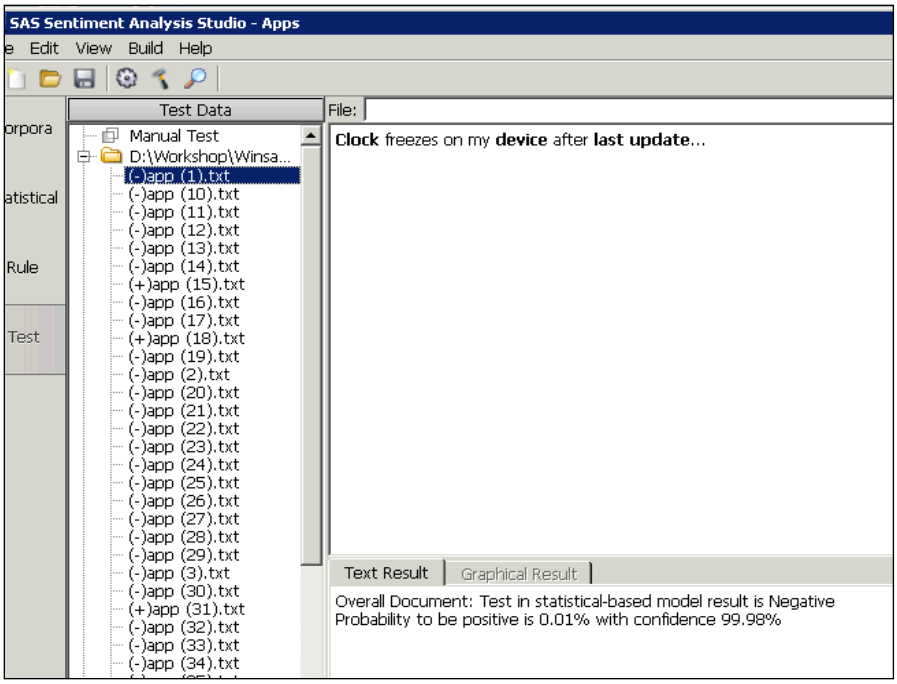


18. Right-click on each folder under Manual Test and select **Test in Statistical Model**. Examine the results.

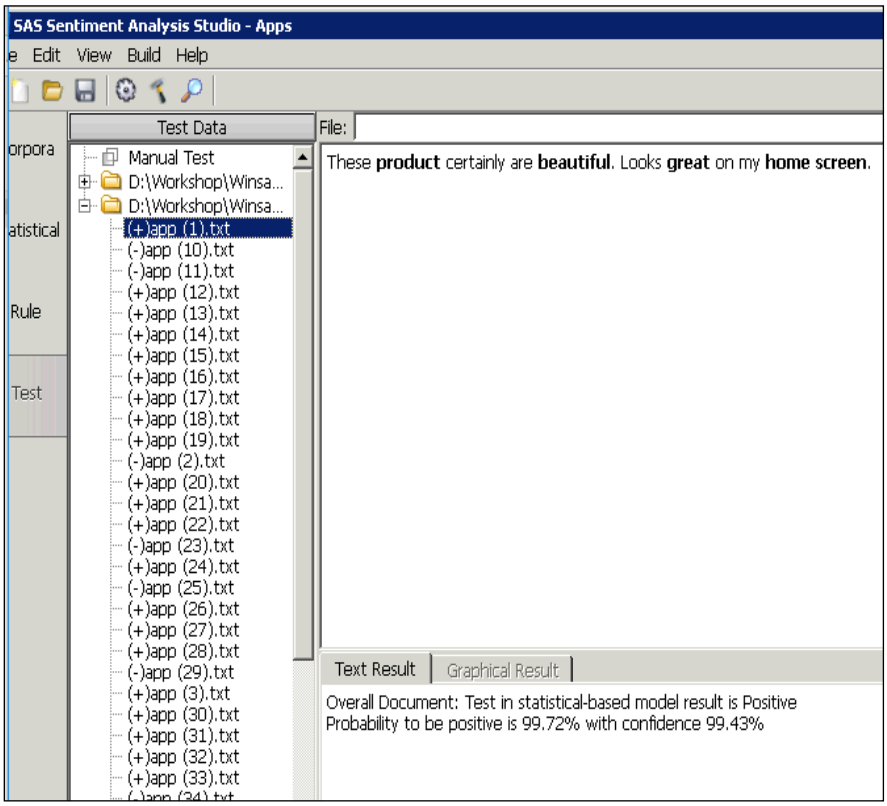




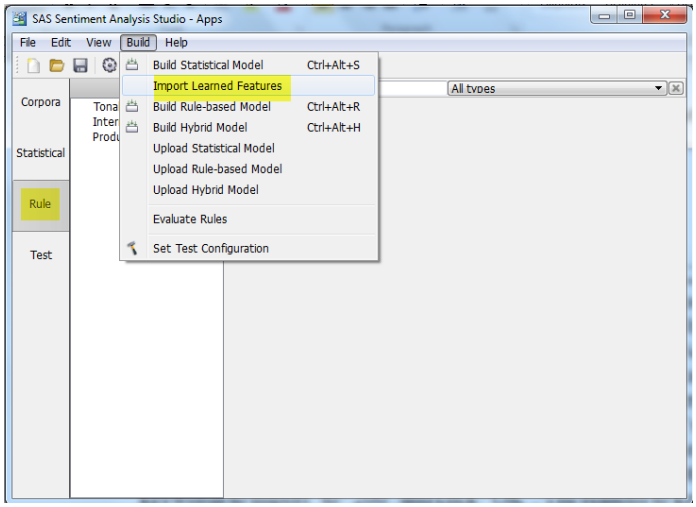
19. Double-click to expand the negative testing directory. Right-click the text file **app(1).txt** and select **Test in Statistical Model**. You see that this file has been predicted as a negative comment. But the exact reason for classification as to why it is negative is unclear.



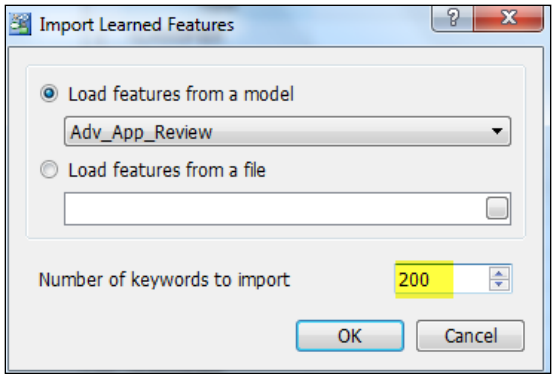
20. Double-click to expand the positive testing directory. Right-click the text file **app(1).txt** and select **Test in Statistical Model**. You see that this file has been predicted as a positive comment. But the exact reason for classification as positive is unclear.



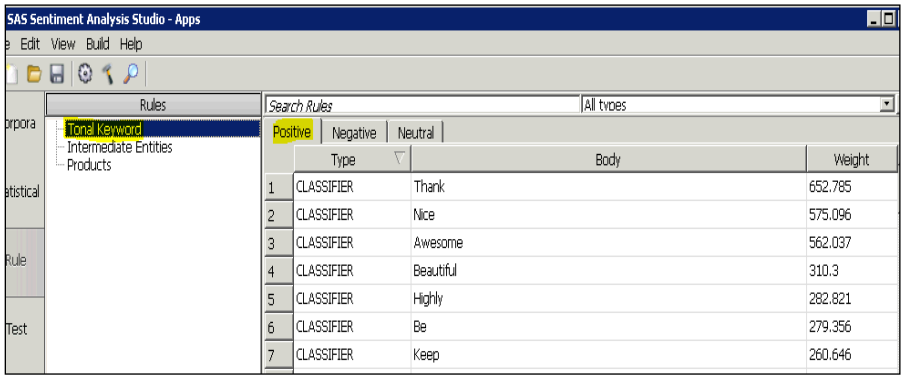
21. Click **Rule** in the left panel. On the top menu, click **Build** ⇨ **Import Learned Features** to import all keywords to start building a rule-based model.

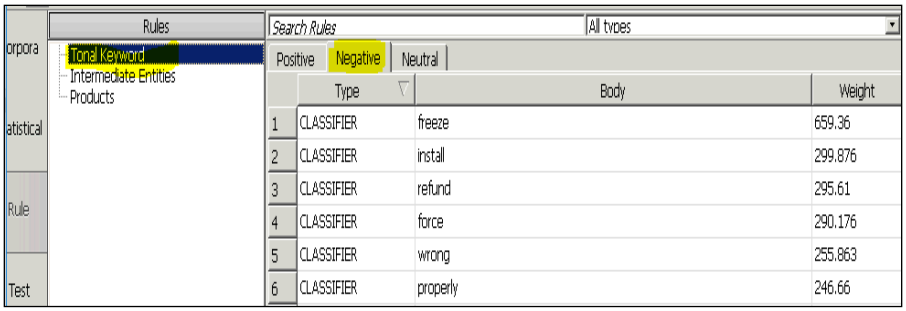


22. You can load the features from a model or a file. Here we will first load features from the statistical model. Change the number of keywords to import to **200** and click **OK**. The reason to keep 200 keywords is because it is easier to review, edit, and change them in later steps.



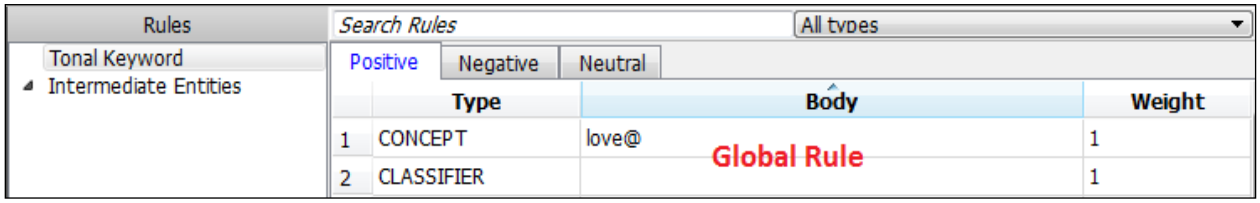
23. Click **Tonal Keyword** and you see 200 positive words and 200 negative words. You can scroll down to see different keywords for positive or negative reviews (toggle using the Positive and Negative tabs). Many of these keywords look reasonable, but some do not. As an analyst, we often study the keywords and then do some editing to create or modify our own list.



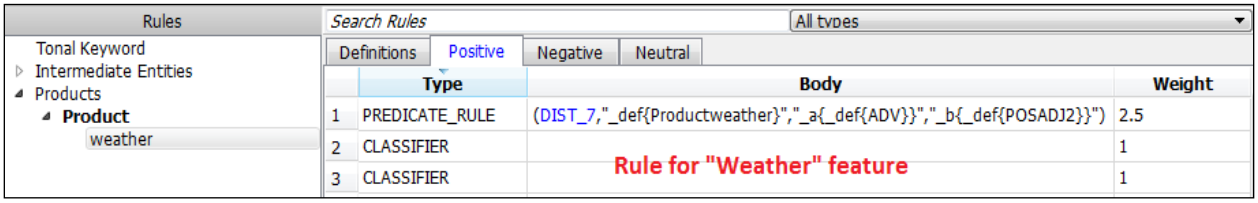


24. We are going to create multiple rules to make this model work better. For now, all terms and words are serving globally. If you select **Build Rule-based Model now** and test the testing data, the number of documents for each feature will be the same. *It often works better if we divide words into different categories. We will create different categories under Intermediate Entities*. The reason to do this is because we want each sentiment word along with its feature to be detected correctly and thoroughly during the sentiment mining process.

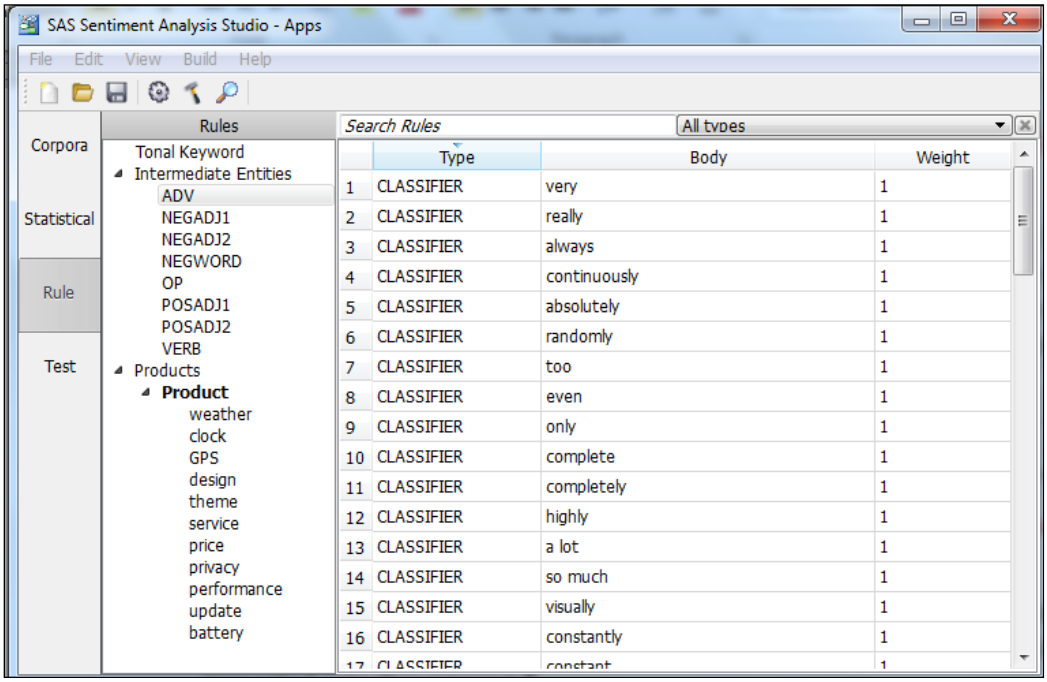
25. If you want to create your own rules, you can either create rules ***globally*** or create rules ***for each feature***. Suppose you want to create a ***global positive*** CONCEPT rule for the word *love* and all of its forms. Click **Tonal Keyword**. On the Positive tab, first edit the body part and then change the type and weight as you deem appropriate, as shown below. In this example, all forms of the word *love*, such as *love*, *loves*, *loved*, and *loving* will be detected and counted for all features in this analysis.



26. Here is an example of creating rules for a specific feature. In this example, this rule will be triggered only if, within a distance of seven words, a feature name (**weather**), an adverb, and a positive sentiment adjective all appear in the document. (We have defined a group of positive sentiment adjectives in the intermediate entities under **POSADJ2**.)

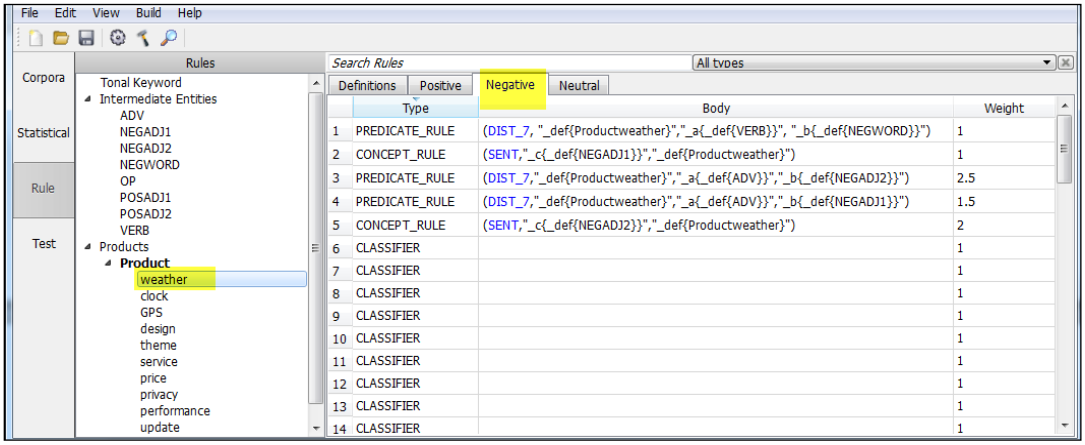


27. For this case, we have created a set of rules and entities. Select **File** ⇨ **Import Rules**. Select the XML document rule file **App_Rule.XML** (App_SM) and click **OK**. Now all rules have been successfully imported into this project.



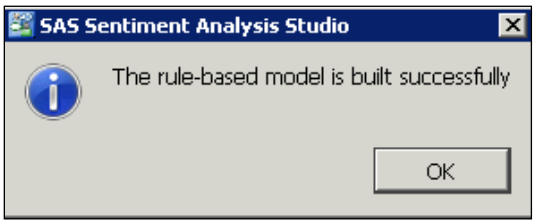
28. From text mining, we have discovered features that we are going to implement in this part. Now you can see all these features are under Product. To create a new product, right-click **Products** and select **New Product**. If you want to add features to Product, right-click **Product** and select **New Feature**.
29. Click some of the other intermediate entities to get a sense of the entities created.
30. Under Intermediate Entities, you can see ADV, NEGADJ1, NEGADJ2, NEGWORD, OP, POSADJ1, POSADJ2, and VERB. Click **ADV** and you see a list of words that are used as adverbs. Words in NEGADJ1 are negative adjectives with a sentiment weight of 1. Similar rules apply to NEGADJ2 but with weights of 2. POSADJ1 is a list of positive adjectives with a sentiment weight of 1. Similar rules apply to POSADJ2 but with weights of 2. NEGADJ1 and POSADJ1 are lists with all adjective words that we considered to have less negative/positive sentiments than the words in NEGADJ2/ POSADJ2. When ADV and NEGADJn/POSADJn happen together, then the sentiment weight becomes n+0.5. Sentiment weights are often subjective difficult to judge as mentioned earlier. OP contains other products that can be seen as competitors. In VERB, it contains verbs that are often used in these reviews.
31. Click **Tonal Keyword** to examine different rules created in this project

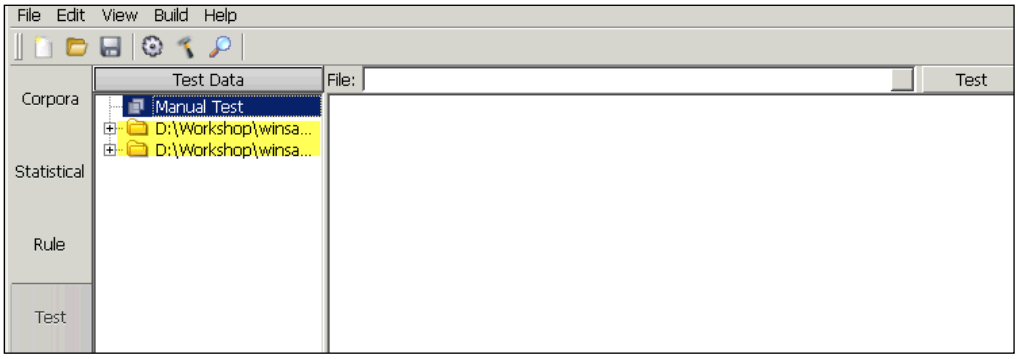
32. If you click through some of the rules, you will find some that have the type of VERB, CONCEPT, and the symbol @ added to the verbs. The symbol **@** assures that all verb forms can be detected. CLASSIFIER, CONCEPT, CONCEPT_RULE, and PREDICATE_RULE types are used for this case study. CLASSIFIER rules are used to match a term or a phrase. We used CLASSIFIER rules to match the words that can be used only for a feature. For example, *expensive* can be used only for the feature **price**. The window below shows an example of PREDICATE_RULE and CONCEPT_RULE. In **DIST_n**, **n** is the number of words between matches on rules. The first match is tagged as position 1 until the last match (n). **_def** matches definition for products or features. **_def{Productweather}** is a definition for the feature **weather** of Product. **_a** and **_b** are arguments that match when these two arguments match in a document. **SENT** will match the words and definition that are only within same sentence.



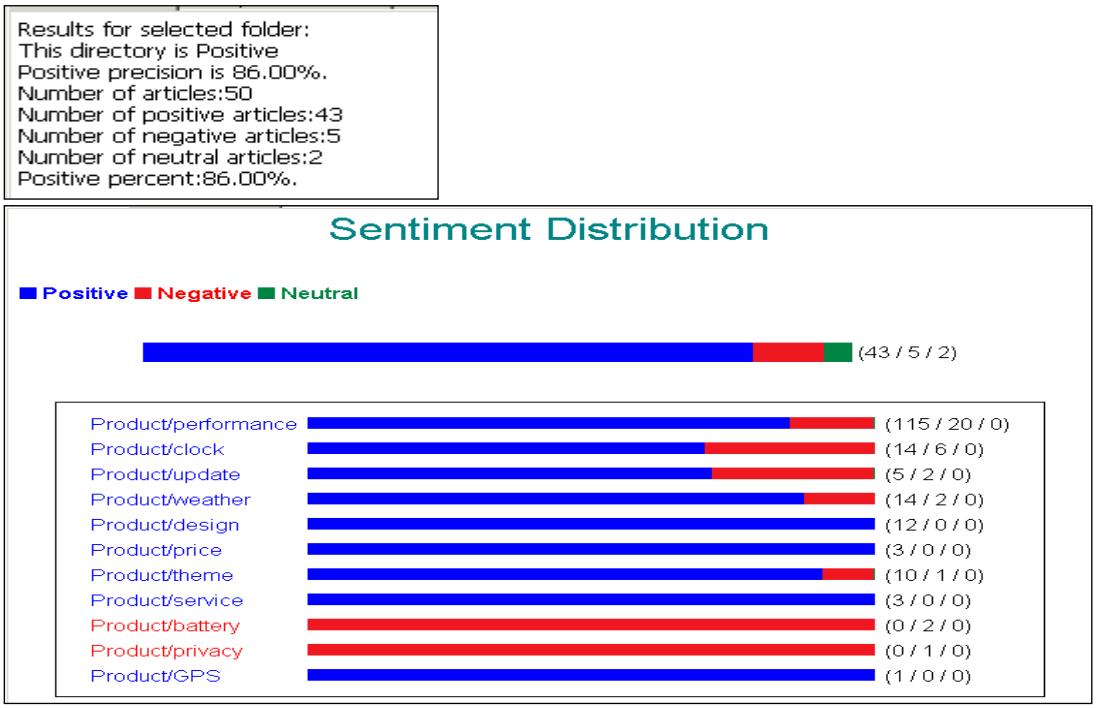
33. Explore more feature-specific rules (positive and negative) on your own by clicking different features and then clicking the Positive or Negative tab.

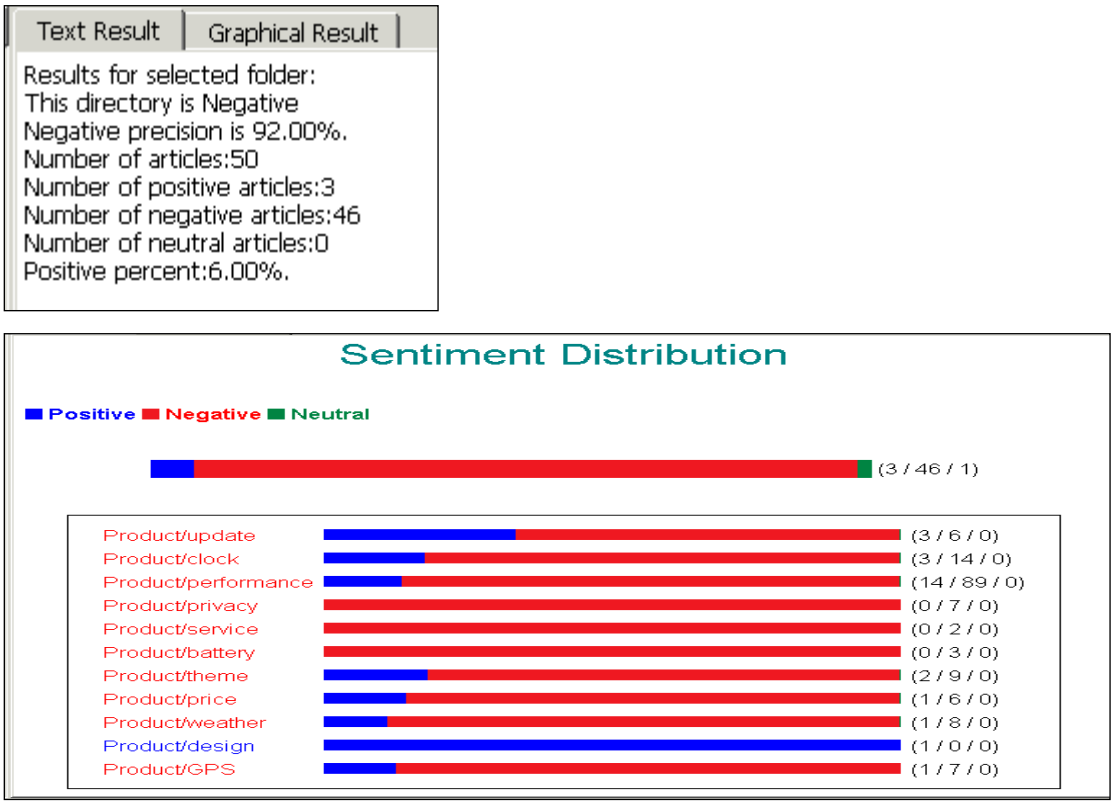
34. Click **Build** ⇨ **Build Rule Based Model**. After a few moments, you will be notified with a “built successfully” message but no visible results. Click **OK**.



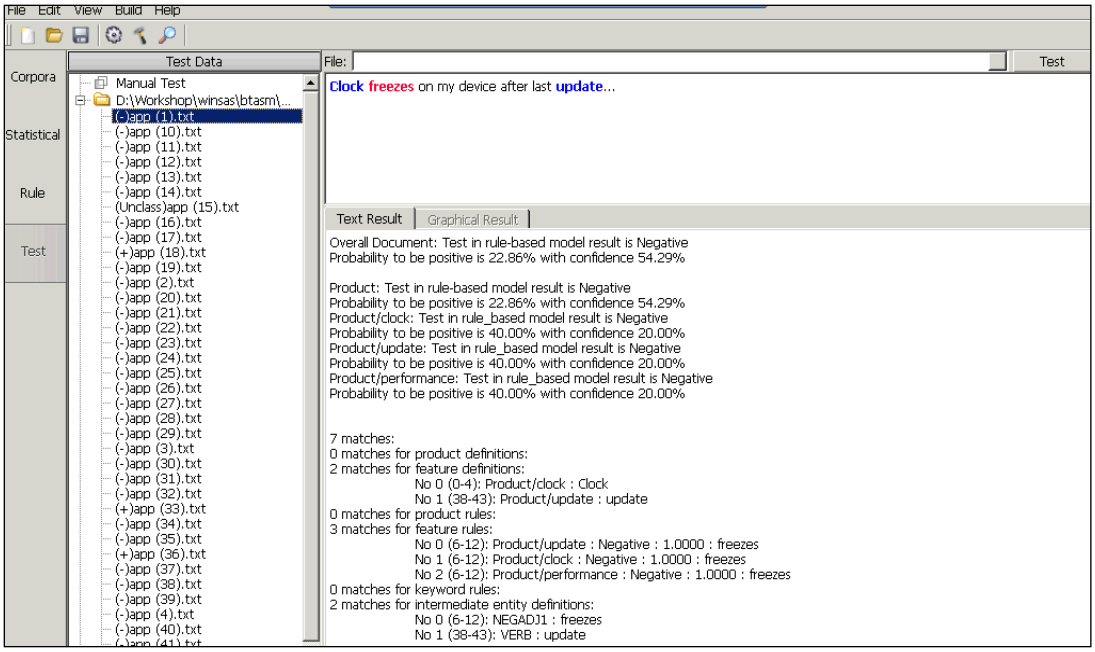


35. Right-click each folder under Manual Test and select **Test in Rule-based Model**. Examine the results.

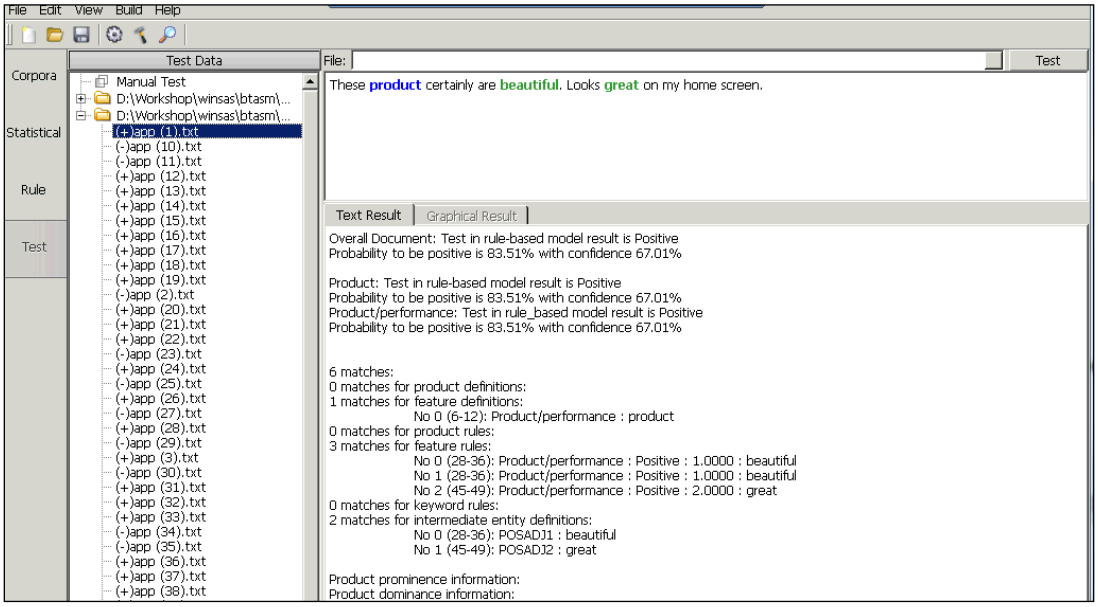




36. Double-click to expand the negative testing directory. Right-click the text file **app(1).txt** and select **Test in Rule-based Model**. You see that this file has been detected as a negative comment. Two features have been detected. Words in blue are detected as features, words in green are detected as positive sentiments, and words in red are detected as negative sentiments.



37. Double-click to expand the positive testing directory. Right-click the text file **app(1).txt** and select **Test in Rule-based Model**. This file has been detected as a positive comment. One feature has been detected.



38. Overall, the rule-based model provides deeper insights into understanding consumers' sentiments.