**USER MANUAL**

Contents

[2 Overview of programs in APOLLO 4 3](#_Toc32142122)

[2.1 IMPORTANT: General instructions for using APOLLO 4 program 3](#_Toc32142123)

[2.2 Description of various tabs in APOLLO 4 3](#_Toc32142124)

[3 Input data format 4](#_Toc32142125)

[4 Train new Model 5](#_Toc32142126)

[4.1 At least five training examples for each class 5](#_Toc32142127)

[4.2 Machine learning models 6](#_Toc32142128)

[4.2.1 A brief description of the supervised learning models 6](#_Toc32142129)

[4.2.2 A brief description of unsupervised learning models 7](#_Toc32142130)

[4.2.3 Optimizing model parameters for performance measures 7](#_Toc32142131)

[4.3 Description of Performance Measures 7](#_Toc32142132)

[4.3.1 Accuracy 7](#_Toc32142133)

[4.3.2 Precision 8](#_Toc32142134)

[4.3.3 Recall 8](#_Toc32142135)

[4.3.4 F1 8](#_Toc32142136)

[4.3.5 Macro Precision 8](#_Toc32142137)

[4.3.6 Macro Recall 8](#_Toc32142138)

[4.3.7 Macro F1 8](#_Toc32142139)

[4.4 Additional stop words for supervised and unsupervised learning 8](#_Toc32142140)

[4.5 Finally, running the document classification program 8](#_Toc32142141)

[4.6 Monitoring the progress of program runs 9](#_Toc32142142)

[4.7 Outputs from the program run 9](#_Toc32142143)

[4.7.1 Output from Supervised Learning 9](#_Toc32142144)

[4.7.2 Output from Unsupervised Learning 13](#_Toc32142145)

[4.8 Saving the model 13](#_Toc32142146)

[5 Use Existing Model 14](#_Toc32142147)

[5.1 Select existing projects and models 14](#_Toc32142148)

[5.2 Use Existing Model for supervised and unsupervised learning 14](#_Toc32142149)

[5.2.1 Use Existing Model for supervised learning 14](#_Toc32142150)

[5.2.2 Use Existing Model for unsupervised learning 15](#_Toc32142151)

[6 Incremental Learning 16](#_Toc32142152)

[6.1 Incremental learning for supervised learning model 16](#_Toc32142153)

[6.1.1 Additional data contains new training examples labeled with existing class labels 16](#_Toc32142154)

[6.1.2 Additional data does not contain any new training examples 16](#_Toc32142155)

[6.1.3 Additional data contains training examples with a new class 16](#_Toc32142156)

[6.2 Incremental learning for unsupervised learning model 17](#_Toc32142157)

[6.2.1 Additional data contains new training examples 17](#_Toc32142158)

[6.2.2 Additional data does not contain any new training examples 17](#_Toc32142159)

[7 Patent Scoring 17](#_Toc32142160)

[7.1 Input Data Format for Patent Scoring Program 17](#_Toc32142161)

[7.2 Program Usage 17](#_Toc32142162)

[8 Further questions? 18](#_Toc32142163)

# Overview of programs in APOLLO 4

APOLLO 4 program ([www.apollo-ip.com](http://www.apollo-ip.com)) is hosted on Amazon Web Services (AWS). Each user will be provided with login credentials by the IP Group Analytics Team.

## IMPORTANT: General instructions for using APOLLO 4 program

Please use only Google Chrome browser for running the analytics programs.

Please do not upload any Samsung confidential data (e.g., invention disclosures) as input to the program.

Please do not upload any NASCA encrypted files on the program. If you happen to accidently upload a NASCA encrypted file, the program will notify you that it is a NASCA encrypted file and will not upload that file to run the program.

APOLLO 4 program can be accessed from only within the Samsung corporate network. If working remotely, please first connect to VPN and then use then login to [www.apollo-ip.com](http://www.apollo-ip.com).

## Description of various tabs in APOLLO 4

APOLLO 4 App has four tabs for four different programs. Here is a brief description of all the programs in APOLLO 4.

* **Train new Model**

This tab is used for training a new model. When starting a new project where there is no previous model or data available, you will need to train a new model. You will be able to create a new project and save the new model, which will be available for later use.

* **Use Existing Model**

This tab is used for classifying the unlabeled data by using a previously trained model. The previously trained model can be a supervised learning model or an unsupervised learning model.

* **Incremental learning**

This tab is used for incrementally updating a model that was previously trained and saved. Consider, for example, that project ‘Demo’ was created and trained on 1000 training examples. If you are not satisfied with the model’s performance, you may label a few more training examples for this project, and quickly update this model using 50 more training examples. Incrementally updating the model is much faster than training the model from scratch on 1050 examples.

* **Patent Scoring**

This tab is used for search for patents that are similar to some search keywords or a list patents, based on a list of related patents provided by the user. The program outputs similarity scores between the input (search keywords or list of patents) and the list of related patents.

# Input data format

The **input files** must contain **appropriate headers** as described below. The input data MUST be provided in a **TAB delimited** text file with a **.txt** extension. If the input files do not contain the appropriate headers, the program will show you an error message to inform you.

**Patent Data Format**

The columns in the input patent file should contain the following data and header:

**Column A:** Identification Number

**Column B:** Title

**Column C:** Abstract

**Column D:** Claims

**Column E:** Application Number

**Column F:** Application Date

**Column G:** Current Assignee

**Column H:** UPC

**Column I:** Category (only applicable to training patent data)

**NOTE: Please do not use the label “Others” in your training data.** APOLLO App outputs Threshold Analysis excel sheet, which automatically predicts some patents as “Others” based on chosen threshold. If you have a category “Others” in your data, then APOLLO App will get confused between the “Others” in YOUR data and “Others” generated by the App.

**Journal Data Format**

The columns in the input journal file should contain the following data and header:

**Column A:** Meta data (e.g. publication name, identification number, publisher type, etc.)

**Column B:** Title

**Column C:** Abstract

**Column D:** Author

**Column E:** Affiliation

**Column F:** Published Year

**Column G:** Category (only applicable to training journal data)

**NOTE: Please do not use the label “Others” in your training data.** APOLLO App outputs Threshold Analysis excel sheet, which automatically predicts some journals as “Others” based on chosen threshold. If you have a category “Others” in your data, then APOLLO App will get confused between the “Others” in YOUR data and “Others” generated by the App.

**Input Data Format for Patent Scoring Program**

If the input file contains search keywords, then the first line of the text file must contain the text “**Search Keywords**” as the very first line of the document. An example of the text is provided below:

Search Keywords

fpga, data center, acceleration, edge, etc.

If the input file contains a list of patents, instead of the search keywords, then the input file format should be exactly the same as **Patent Data Format** (as shown above).

**What if the input file format does not meet the above specification?** APOLLO 4 App allows you to train the model using patent or journal data, depending on the input training (labeled) data file chosen. You may choose the testing data to be either patent data or journal data. Note that the App automatically determines whether the data is patent data or journal data depending on the file headers, and shows the type of data (patent/ Journal) in the GUI.

**NASCA encrypted file:** APOLLO program cannot read any NASCA encrypted files. If the input file is NASCA encrypted, the program will show an error message to inform you that the file is NASCA encrypted. You will need to remove the NASCA encryption from the file. Removing the NASCA encryption from a file is not straightforward. One way to work around this problem is to open Notepad and copy-paste the data from the NASCA encrypted file into the Notepad file. Saving the file using Notepad does not automatically encrypt the file with NASCA.

# Train new Model

Please use this tab whenever you want to train a new model using supervised learning or unsupervised learning.

**Supervised learning mode:** In this mode, you need to provide labeled training data to train the model.

* **Training (labeled) Data File:** This file should contain the sample patent/journal data with the category (labels).
* **Unlabeled Data File:** This file should contain the sample patent/journal data without any category. The program will automatically predict the category for the patents/journals in this file based on the training data.

**Unsupervised learning mode:** In this mode, you need to provide only the data, but NOT the labels.

Click on the radio button “Unsupervised Learning” to run the program in unsupervised learning mode. In this mode, you will need to provide only **one** file containing the patent/journal data WITHOUT any categories (or labels). ***Figure 9*** below shows where to select the files for **unsupervised** learning mode.

* **Training (unlabeled) Data File:** This file should contain the sample patent/journal data without the category (labels).
* **Number of clusters and top words:** You need to provide the number of clusters and the number of top words that you want the model to output.

**Additional stop words:** Please provide any words or phrases that you want the model to ignore while clustering and determining the topic for the clusters.

## At least five training examples for each class

**\*\*\*An important note for supervised learning: You need to provide at least 5 training examples for each class.**

Please note that in the supervised learning mode, the program requires at least 5 training examples for each class in the labeled data. If you have less than 5 training examples for a class and try to run the program, the system will prompt an error message to inform you of this error. If you see this error message, please provide more training examples for the classes which have less than 5 training examples. When you try to run the program, the App will show the # Training examples for each class in the “DATA STATISTICS” section.

**\*\*\*Please note that the program will NOT run until you provide at least 5 training examples for each class in the sample training data for supervised learning.**

## Machine learning models

In this App, you will be able to choose one of the many available models for supervised learning. There is no single model that will work well for all the data sets and tasks, and therefore, you may want to try a few different models to determine which model works best for YOUR data set.

Currently, the App supports **six** models, which work well for text classsification task and with many classes.

### A brief description of the supervised learning models

#### **Logistic regression**

Logistic regression model works quite well in general and takes relatively less time for training, and hence, it is a good idea to first try using logistic regression model.

#### **Multinomial Naïve Bayes**

Multinomial naïve Bayes model also takes relatively less time for training and may sometimes be better than logistic regression. This would be the second choice of models to try.

In general, if you have a very large dataset, then it is a good idea to first try logistic regression and then multinomial naïve Bayes models.

#### **Support vector machines**

Support vector machines model also works quite well, and is often better than multinomial naïve Bayes and logistic regression, however, if your training data set is too large (i.e., more than 500 examples in the TRAINING data file), then it could take a very long time to train this model. Hence, use this model if you have fewer than ~500 examples in your training data OR when multinomial naïve Bayes and logistic regression models do not work well for your data set.

#### **One vs Rest approaches**

The One vs Rest approach trains one model per class, where each model learns to distinguish between examples of one class vs examples of all the other classes in the data set. When you have more than two or three classes in your training data, it is worth experimenting with One vs. Rest approaches, as this approach works better when there any several classes in the training dataset. Since this strategy builds one model per class, it may take a long time to train the One vs Rest models.

NOTE: when you have only **two classes** in your training dataset, then One vs Rest approaches **cannot** be supported. In this case, the program will show an error message to inform you to choose a different model.

In general, when you have more than two classes in your training data set, and both logistic regression and multinomial naïve Bayes do not give good results, then you should try to use One vs Rest approaches.

Please note that “One vs Rest (Logistic Regression)” and “One vs Rest (Multinomial Naïve Bayes)” models run way faster than “One vs Rest (Support vector machines)” model. Try using “One vs Rest (Support vector machines)” model only when none of the other five models work well for your data set.

#### **Automatically determine the best model and use the best model for training**

If you are not sure which model to try first and would like the App to choose the best model for your data set, then click on the radio button “Automatically determine the best model and use the best model for training”. Please note that when you choose the option to automatically determine the best model, the App will train and **compare** only **four** models: (1) Multinomial Naive Bayes, (2) Logistic Regression, (3) One vs Rest (Multinomial Naive Bayes), and (4) One vs Rest (Logistic Regression). This is because training the Support Vector Machines model takes a long time. If you want, you can still run the program with Support Vector Machines and One vs Rest (Support Vector Machines) models separately.

### A brief description of unsupervised learning models

#### **K-Means Clustering**

Currently, APOLLO 4 App supports only the “K-means clustering” model to group similar documents into the number of clusters that you choose.

### Optimizing model parameters for performance measures

Each supervised learning model has its own parameters that need to be optimized for each data set and the performance of the model heavily depends on how well these parameters are tuned for a particular task or data set. Fortunately, the APOLLO 2 App automatically optimizes these parameters specifically for your data set.

There are many ways in which a model’s performance can be evaluated e.g., based on accuracy, precision, recall, etc. Please refer to the next section titled “Description of Performance Measures” in this user manual to learn and understand the various performance measures. The App lets you decide which performance measure is suitable for your data set or task and optimizes the parameters of the model to maximize the performance measure that you choose.

It is OK if you do not fully understand which performance measure to choose for your task. If you cannot make a decision on which performance measure to choose, please contact the IP Group Analytics Team (Send an email to IP Group Analytics Team: [apolloip@ssi.samsung.com](mailto:apollo-ip@ssi.samsung.com) for further help). You might also want to experiment with a few different performance measures on your own.

## ­**Description of Performance Measures**

### **Accuracy**

Accuracy is the number of documents correctly predicted divided by the total number of documents. The accuracy measure might not be suitable when you have too many examples in one class and only a few examples for the other class.

Consider for example a scenario where **95%** of the instances belong to **category A** and only **5%** of the instances belong to **category B**. Consider a model that simply categorizes every example in the data to category A, then such a model will technically be “95% accurate”, however, such a model will be useless for predicting instances of category B. This is an example scenario which shows that optimizing for accuracy is not a good idea for such cases where there are too many examples of one class and too few examples of another class. Please keep reading this section further to learn about other performance measures that might be more meaningful in such scenarios.

**NOTE: Accuracy** is computed **over all the classes** and for all the documents. **Precision, recall,** and **F1** measures are defined for **ONLY one particular class.**

### **Precision**

Considering a particular class ‘A’, precision for class ‘A’ is defined as the number of class ‘A’ documents in the test file that actually belong to class ‘A’ divided by the total number of documents that are predicted as class ‘A’ in the test file.

### **Recall**

Considering a particular class ‘A’, recall for class ‘A’ is defined as the number of documents in the test file that are predicted as class ‘A’ divided by the actual number of class ‘A’ documents in the actual/correct label file.

### **F1**

F1 score is the harmonic mean between the recall and precision; it is taking some sort of an average of the precision and recall measures.

### **Macro Precision**

Macro precision first computes precision for each class individually, and then takes a simple average of the precision for all the classes.

### **Macro Recall**

Macro recall first computes recall for each class individually, and then takes a simple average of the recall for all the classes.

### **Macro F1**

Macro F1 first computes F1 for each class individually, and then takes a simple average of the F1 scores for all the classes.

## Additional stop words for supervised and unsupervised learning

This App enables you to enter any stop words that you would like the model to ignore during classification, clustering, or determining the topic for ‘OTHERS’ category.

**Note**: you may use “Ctrl+c” and “Ctrl+v” options using your keyboard to copy-paste text from anywhere into this textbox. However, please note that it is YOUR responsibility to make sure that the additional words are **separated by commas**.

## Finally, running the document classification program

Once you are satisfied with all the settings, click on “**RUN DOCUMENT CLASSIFIER**” button to actually start running the program. Please note that when you click the “RUN DOCUMENT CLASSIFIER” button, it will show the “OUTPUT AND ANALYSIS” page.

APOLLO 4 App shows the progress bar to show roughly how much the program run has completed, and roughly how much time is remaining for the program run to complete.

## Monitoring the progress of program runs

The progress bar shows exactly when the program run was started. You may want to periodically view and monitor the PROGRESS frame to check the status of the program run. The progress frame is updated instantly as soon as the program run makes any progress. You may log out and log back in at any time. If the program was running when you log out, you will be able to log back in any time to monitor the progress of program run without losing any work.

Once the program run finishes, the program will pop up a message box to inform you that the program has finished running successfully. APOLLO 4 App also shows exactly how much time the program run took.

In case there is any problem with running the program, the program will pop up an error message and will also update the PROGRESS frame to inform you about any errors that were encountered while running the program. In case of errors, please contact the IP Group Analytics Team (Send an email to [apolloip@ssi.samsung.com](mailto:apollo-ip@ssi.samsung.com)) to report any issues.

**Note that while the program is running, you will not be allowed to change any inputs, as this might affect the program run.**

## Outputs from the program run

The program output can be seen in the OUTPUT AND ANALYSIS frame. Once the program run finishes successfully, you can see the basic results of the program in the GUI and the detailed results in the output files generated by the program.

**Once the program run finishes, the output files will be available for downloading. Please click on the results file(s) in the web browser and it will automatically download and store the downloaded files into your default download folder.**

### Output from Supervised Learning

Once the program run finishes successfully, check the “OUTPUT AND ANALYSIS” frame in the App to see some of the basic results of the program run.

The App displays basic statistics for the examples in both the training data and the testing data. The OUTPUT AND ANALYSIS frame contains several frames as described below.

1. **DATA STATISTICS frame:**

In this frame, for the training data, the App displays the number of training examples (**# Training examples**) and the number of classes in the training data (**# Classes**).

For the testing data, it displays the number of examples in the testing data (**# Testing examples**).

For both training and testing data, the App also displays the names of the classes (**Class**), the number of examples in each class (**# Examples**), and the percentage of training examples in each class (**Class %**). Note that for the testing data, the ‘**# Examples**’ and ‘**Class %**’ are based on the predictions made by the model.

In this frame, you can click on any of the column headings to display the results in a sorted order.

1. **MODEL SELECTION frame:**

The App displays the “Selected Model” and the Hyper-parameter setting that was chosen for the model. Note that if you choose the option to “Automatically determine the best model and use the best model for training”, then the “Selected Model“ will display the name of the best model that was automatically chosen by the App.

1. **MODEL EVALUATION frame**

This is a **very important frame**, because it allows you to evaluate the model’s average performance on the training data set using an approach known as 5-fold Cross Validation.

Note: The App evaluates the performance of the model by performing a procedure known as “Five-fold cross-validation” on the training data. This procedure provides an **estimate** of how well the model can perform on the **training** data. This does not mean that the model will have the same (or even similar) performance on the testing data. However, this evaluation provides you with some idea about how well the model could perform on the testing data.

It is important for you to carefully analyze the results displayed in this frame and see whether the model’s performance is “okay” for your task. This frame displays the average accuracy, AUC, Macro Precision, Macro Recall, and Macro F1 scores, as well as the standard deviations. Please refer to the last section titled “Description of Performance Measures” in this user manual to learn and understand the various performance measures.

If the accuracy of a model is very low, there are **TWO possible choices** to improve the model performance:

1. Re-run the program using some other machine learning models and see if any other model performs better for your data set or task.
2. If none of the machine learning models give you the desired model performance, then you should consider providing more labeled data in your training set and re-run the program to see if it improves the model performance.
3. **PROGRESS frame**

This frame shows the progress made by the program.

**EXCEL SHEET TO DETERMINE THE BEST THE RELEVANCE THRESHOLD VALUE**

All the output files from the program are by default saved in the folder where your testing data file is located. Open the folder were your testing data file is located.

Once the program run finishes, the App outputs an excel sheet from the web browser. The output file is named “Your\_testing\_data\_file\_name\_**Threshold\_Analysis**.xlsx”, and it provides you with the capability to play with various threshold values for OTHERS. Once you have finalized a particular threshold value using the excel sheet, you may need to run the program one more time using the instances in the ‘OTHERS’ category and using unsupervised learning mode.

How to play with the relevance threshold in the excel sheet?

Please open the excel sheet named “Your\_testing\_data\_file\_name\_**Threshold\_Analysis**.xlsx”. You will see two tabs in this excel sheet:

1. **Results** tab: you will need to use only this tab to play with relevance thresholds
2. **Raw\_Data** tab: please **do NOT change** **any data** in this tab, as the data in this tab is used to evaluate which documents should be categorized as ‘OTHERS’.

The figure below shows the Results tab of the excel sheet. The first few columns in this tab contain the testing data that you provided (e.g., title, abstract, claims, …, etc.). The excel sheet has two more columns named “PREDICTED\_CATEGORY” and “PREDICTED\_CATEGORY\_WITH\_OTHERS”.

1. **Column “PREDICTED\_CATEGORY”**: this column contains the category for the patent/journal based on the relevance threshold of 0, i.e., no document is categorized as ‘OTHERS’.
2. **Column “PREDICTED\_CATEGORY\_WITH\_OTHERS”:** this column contains the category for the patent/journal based on the relevance threshold value that you enter in the excel sheet.
3. **The cell “THRESHOLD” highlighted in yellow color:** In the cell next to the THRESHOLD cell, you can enter **ANY** threshold value that you want, and it will instantly update the column named “PREDICTED\_CATEGORY\_WITH\_OTHERS” to show you which patents/journals have been categorized as ‘OTHERS’.
4. **The cell “COUNT (OTHERS)” highlighted in yellow color:** As soon as you change the threshold value, the cell next to the “COUNT (OTHERS)” is instantly updated to tell you how many of the patents/journals are categorized into ‘OTHERS’ based on the threshold value you enter.

Please play with several threshold values in this excel sheet and finalize a threshold that makes sense to you for YOUR data. Once you have finalized a threshold value, **please re-run the program one more time** using the documents in ‘OTHERS’ category.

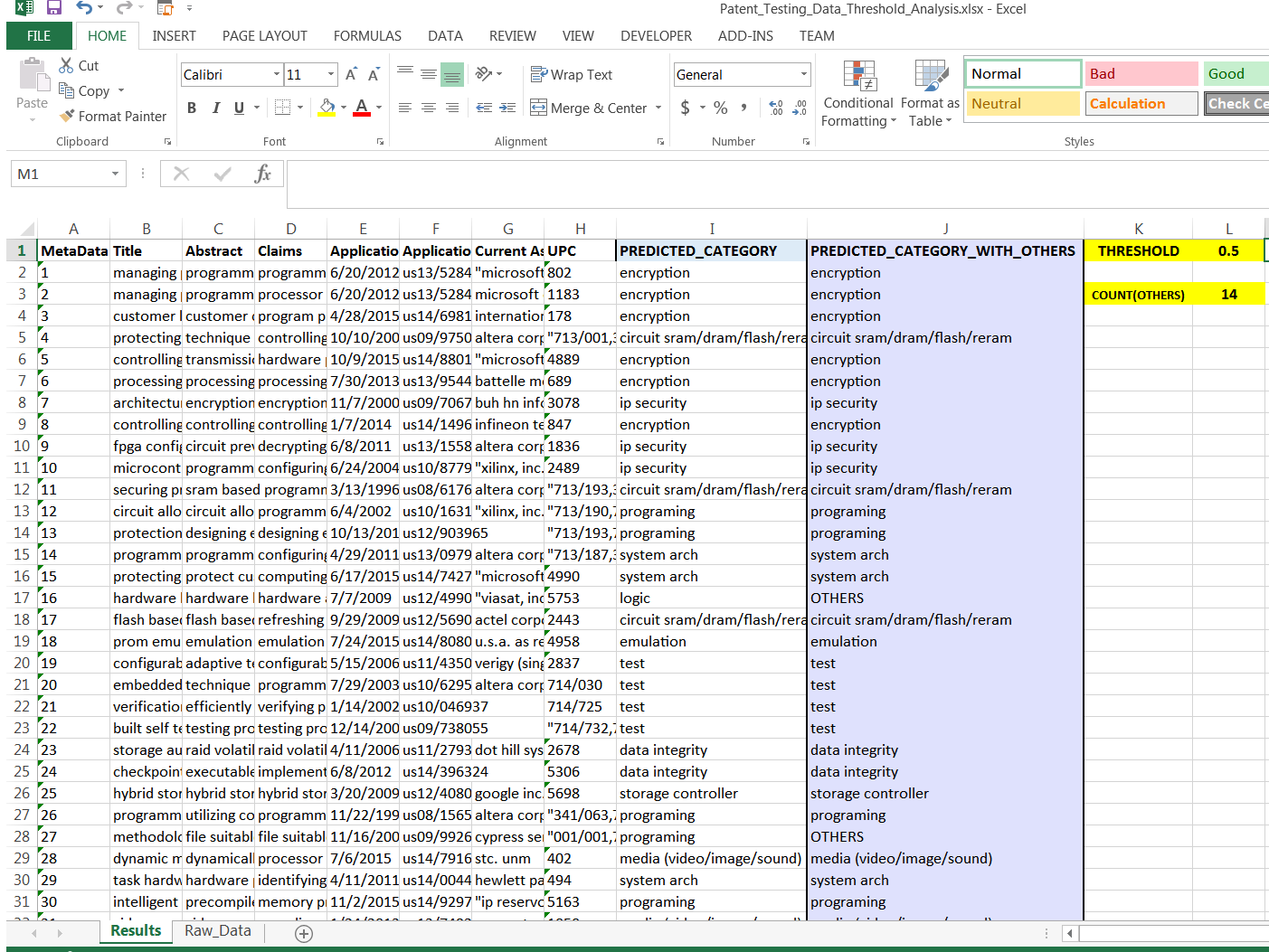


Figure: Excel sheet to play with various threshold values

**EXCEL SHEET TO PERFORM DETAILED ANALYSIS ON MODEL’S PERFORMANCE ON THE TRAINING DATA**

The App also outputs an excel sheet named “Your\_testing\_data\_file\_name\_**Threshold\_Analysis\_Training\_Data**.xlsx”, which provides you with the capability to view model’s performance on the training data. The figure below shows the output excel sheet. This excel sheet shows the accuracy on training data in Column ‘K’ of the excel sheet. This excel sheet also shows in Column ‘J’ whether the model has predicted the document in the training data correctly.

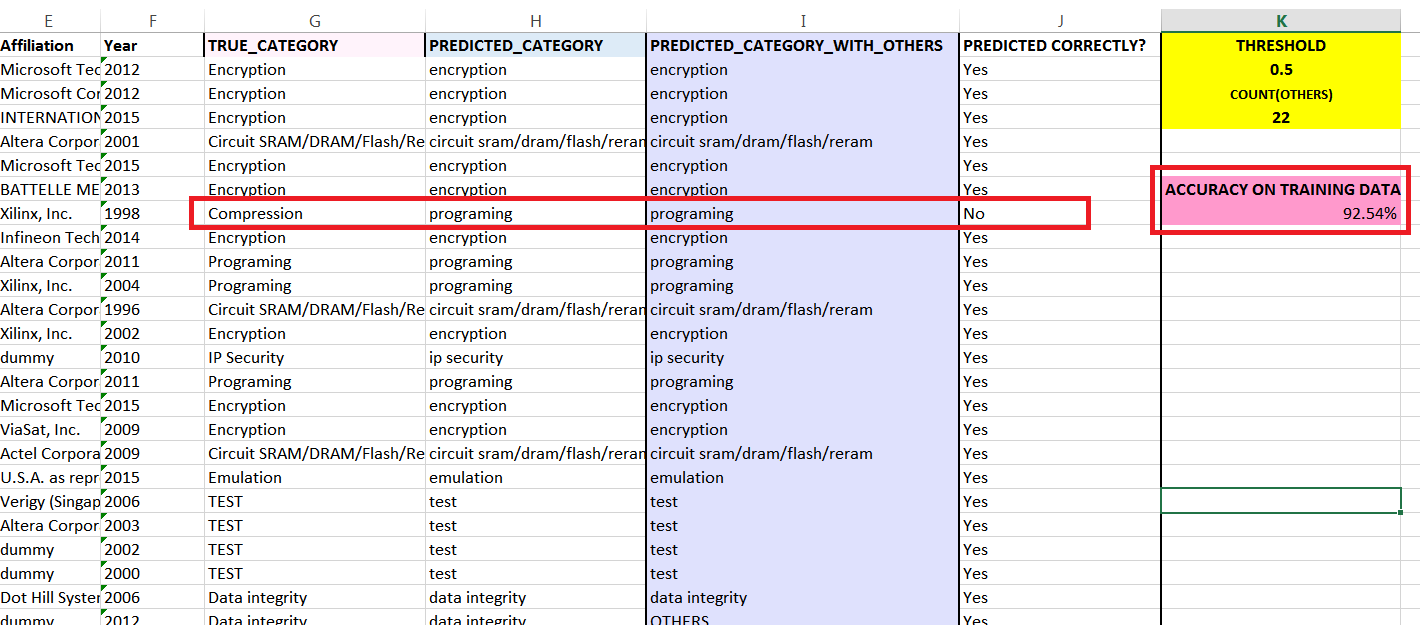


Figure: Excel sheet for Threshold Analysis on Training Data

**DETAILED RESULTS FROM SUPERVISED LEARNING AVAILABLE IN THE RESULTS FILES:**

The detailed results from the program can be found in the output file named “Your\_testing\_data\_file\_name**\_Prediction\_Results**.txt” which contains all the classified patents/journals with the categories in the last column.

### Output from Unsupervised Learning

Once the program run finishes successfully, check the “OUTPUT AND ANALYSIS” frame in the App to see some of the basic results of the program run including the topic modeling results. The OUTPUT AND ANALYSIS frame displays the basic topic modeling results in a table within the “TOPIC MODELING RESULTS” frame showing the “Topics” and the number of examples (# Examples) that belong to the topic.

The App allows you to sort the data in the columns by clicking on the headers. Clicking the header once will sort in ascending order, and clicking the header again will sort the column in descending order.

Once the program run finishes successfully, the App will also allow you to download the file named “Your\_data\_file\_name\_**Topic\_Modeling**.txt” which contains the topic modeling results, i.e. the topics assigned to each of the patent/journal in the data file that you provided to the program. The file contains all the classified patents/journals with the categories in the last column.

## Saving the model

Once a model is trained successfully, you will be able to click on “SAVE MODEL” button to save the model. Note that the “SAVE MODEL” button is enabled only after a model has been successfully trained. Clicking on the “SAVE MODEL” button will open another dialog box that will allow you to create a new project for which you can save the model.

If a project already exists, you may choose to save the project to an existing project by clicking on the radio button “Save the model to an existing project” and select a project from the drop-down menu. Alternatively, you may want to create a new project by clicking on “Create a new project and save the model to the new project. In this case, you will need to provide a project name and a project description of the new project. Please provide at least 2-3 lines of project description that will help you to remember why you created the project and the task that you are working on.

Note that you will also need to provide a short description of the model that you have trained, so that you will remember the purpose and task that the model was built to perform.

Once the model is saved, the program will show a message box to inform that the model has been saved successfully. The program will also show how much time it took for the program to save the model. Usually, saving the model takes very little time. Note that the program saves the model as well as all the data associated with training the model.

# Use Existing Model

Please use this tab when you have unlabeled data that you would like to quickly classify using an existing model.

## Select existing projects and models

From the first dropdown list, select an existing project. The App will show a second dropdown list containing all the existing models for the selected project. Note that there may be more than one existing model for the same project. In the case where there is more than one existing model, please use the second dropdown list to select a model that you would like to use for classifying the unlabeled data. APOLLO 4 App displays the performance of each model, which might help you in selecting a suitable model for classification.

## Use Existing Model for supervised and unsupervised learning

Note that you may use an existing supervised learning model or an existing unsupervised learning model for classifying the unlabeled data. When the selected model is a supervised learning model, the program will perform classification. When the selected model is an unsupervised learning model, the program will perform clustering.

### Use Existing Model for supervised learning

When the selected model is a supervised learning model, the program will perform classify the unlabeled data into classes that the model was trained on. Select the unlabeled data file by using the “Browse” button. Once you are happy with the selection of model and have provided the input data file in the GUI, please click on “RUN DOCUMENT CLASSIFIER” button to run the program and perform classification using the selected model. The App will show the progress during the program run.

Once the program finishes successfully, the App will display a “Success” message box. All the output files from the program will be available to download from the web browser. The output of classification results will be two files named:

* “Your\_testing\_data\_file\_name\_**Threshold\_Analysis**.xlsx”
* “Your\_testing\_data\_file\_name\_**Prediction\_Results**.txt”

### Use Existing Model for unsupervised learning

Once you are happy with the selection of unsupervised learning model and have provided the input data file in the GUI, please click on “RUN DOCUMENT CLASSIFIER” button to run the program for clustering new documents.

For unsupervised learning using an existing model, the App performs clustering as described below.

1. Suppose that the existing unsupervised learning model was trained using 10 clusters. The additional training data will be first categorized into one of the 10 clusters, depending on which cluster is closest to the document.
2. The old clusters will now have some more training examples. The App will re-compute the topics for each cluster based on the new data in each cluster.
3. The new topics can be seen in the OUTPUT AND ANALYSIS frame in the “TOPIC MODELING OUTPUT RESULTS” table.

Once the program finishes successfully, the App will display a “Success” message box. All the output files from the program will be available in the web browser. Please click on the output file to download the results. The results will be saved into your default download folder. Open the folder were your testing data file was saved. The classification results will be output in the file named:

* “Your\_testing\_data\_file\_name\_**Results\_Topic\_Modeling**.txt”

# Incremental Learning

Use this tab to incrementally update an existing model. This tab allows you to view all the previously saved models. Consider, for example, that a project related to ‘Data Center’ was created. Let’s say the previous model was trained on 1000 training examples. Let’s say you labeled a few more training examples (50 examples) for this project. You can choose the previous model that was originally trained on 1000 examples, and quickly update this model using 50 more training examples. Incrementally updating the model is much faster than training the model from scratch on 1050 examples!

## Incremental learning for supervised learning model

In order to incrementally update a supervised learning model, you need to select a supervised learning model for a project. From the first dropdown list in this tab, select an existing project. From the second dropdown list, select an existing model for the selected project.

Select the file containing additional training data using the “Select File” button. Click on “RUN DOCUMENT CLASSIFIER” button to update the model incrementally. There are three possible scenarios for incrementally updating a supervised learning model:

### Additional data contains new training examples labeled with existing class labels

If the additional data contains new training examples, and the classes or categories in the new training examples are the same as in the original data, then the program will incrementally update the model with additional training examples. Note that the new training examples do not need to have examples of all the classes. The new training examples may contain documents belonging to only one or two classes, however, those classes must be present in the original training dataset. When the incremental learning is successfully completed, the App shows the “Success” message box. The model’s updated 5 fold cross-validation performance is displayed in the GUI.

### Additional data does not contain any new training examples

APOLLO 4 App automatically de-duplicates the new training data based on the old training data. If the additional training data file does not contain any new patent or journal data, the program will display an error message to inform you that there is no new data to update the model. In case when there is no new data available for training, incremental learning cannot be performed.

### Additional data contains training examples with a new class

In case the additional training data contains a new class, the program will inform you that there is a new class in the training data. In case when the additional training data contains a new class, incremental learning is not possible, and the model has to be trained from scratch, i.e., regularly training the model instead of incrementally updating the model. In this case, the App will display the error message to inform you about this case. If you want to train the model with the ‘new class’ added to the training dataset, click on “Yes” button to retrain the model from scratch.

Note that the program requires at least five training examples for each class. If the new class does not contain at least five training examples, the program will show an error message to inform you that the ‘new class’ contains less than five examples. In this case, incremental learning is not possible and you need to update the additional training data file to provide at least five training examples of the new class.

## Incremental learning for unsupervised learning model

To incrementally update an unsupervised learning model, select an unsupervised learning model from the drop-down lists. Select file with additional data using the “Browse” button. After selecting the file, click on the “RUN DOCUMENT CLASSIFIER” button to perform incremental learning. There are two possibilities in this case:

### Additional data contains new training examples

If the additional training data file contains new training examples, then the App will perform incremental learning to incrementally update the unsupervised learning model. Once the incremental learning finishes successfully, the App will show a “Success” message box to inform that incremental learning process is finished successfully.

### Additional data does not contain any new training examples

In case the additional training data does not contain any new training examples, the App will show an error message to inform you that there are no new training examples for training the model. In this case, incremental learning is not possible, since there are no new examples to update the model.

# Patent Scoring

Please use this tab to search for patents that are similar to the content in the input document. The input document can contain either search keywords or a list of patents. Please note that you will also need to provide a list of related patents which will be used by the program for comparison against the search keywords or patents.

## Input Data Format for Patent Scoring Program

If the input file contains search keywords, then the first line of the text file must contain the text “**Search Keywords**” as the very first line of the document. An example of the text is provided below:

Search Keywords

fpga, data center, acceleration, edge, etc.

If the input file contains a list of patents, instead of the search keywords, then the input file format should be exactly the same as **Patent Data Format** (as shown above).

## Program Usage

This program has multiple uses, as described below.

**Prior Art Search:** This program can be used to perform prior art search. If you would like to search for patents that are related to an invention disclosure, you will need to provide search keywords in the input file.

Please note: do not upload the entire invention disclosure document, as this is Samsung confidential information. You may, however, provide some search keywords that are most related to the invention disclosure document.

**Searching for Infringement Cases:** This program can also be used to search for patents that might be potential infringement cases. If you would like to search for potential infringement cases, then the input file should contain the patent (or a list of patents) for which you want to search potential infringement cases.

The program takes as input two files:

* File containing Search Keywords or a List of Patents
* File containing a list of related patents

Please use the “Browse” buttons in this tab to upload the two files.

Please click on the button “Compute Similarity b/w Input and Related Patents” to start the program run. Once the program run finishes, the program will output all the results in a zip file. Please download the zip file and extract the files within the zip file to view results.

If the input file contains search keywords, the zip file will have one file containing the prior art search results (only the top 5 most related patents). This is to ensure that IP Managers do not view more than top 5 most related patents.

If the input file contains a list of patents, then each output file within the zip file will contain the “Application number of input patent” in the file name and contain the similarity scores (similarity percentage value between 0 and 100) between a particular input patent and all related patents. Please note that in the output files, only certain patents will be displayed based on the following criteria:

* Related patents that have application date later than or equal to the application date of the Input patent
* Related patents that are NOT assigned to Samsung.

Let’s consider for example, that there are 200 related patents. Let’s say that for a particular patent with application number US12345678 in the input list, if only 100 related patents have application date later than the application date of the particular patent (US12345678). Then only those 100 patents will be displayed in the output file named US12345678\_rank.txt

# Further questions?

If there are any further questions that are not answered in this user manual, please contact the IP Group Analytics Team (Send an email to IP Group Analytics Team at [apolloip@ssi.samsung.com](mailto:apollo-ip@ssi.samsung.com)) for help.