Step-by-step guide

From the Apps interface, select Splunk Machine Learning (Figure 1).

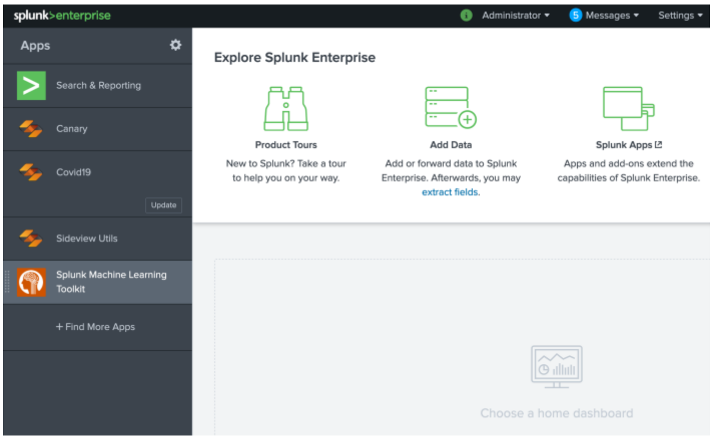


Figure 1: Splunk Apps

We will now analyse a firewall log for malware. For this select "Predict Numeric Fields", and then "Create New Experiment":

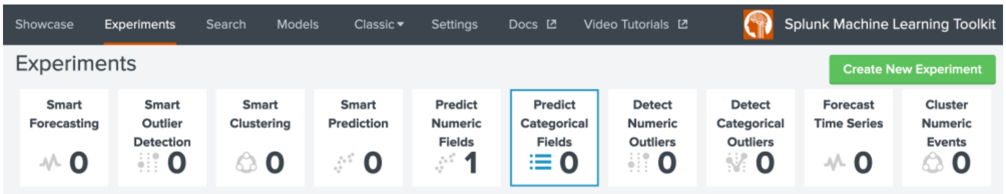


Figure 2: Selecting a Predict Categorical Fields experiment

Provide the name of "Firewall" to the experiment title (Figure 3).

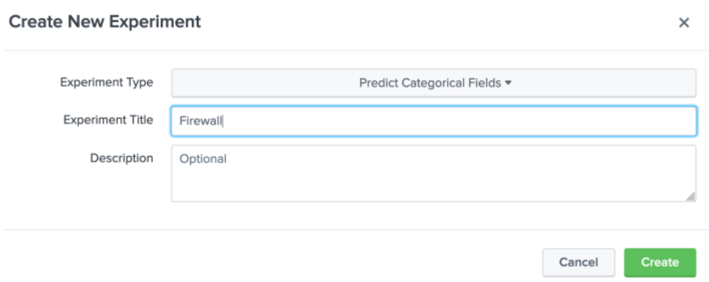


Figure 3: Defining a new experiment

Next (Figure 4) entered the search of **"| inputlookup firewall_traffic.csv"** and select the green search button. It will then populate the data set in the page. Scroll down to the populated dataset and define the following:

Number of results in the dataset: Parameters used in the dataset: Which field do you think we are likely to train on:

Outline four IP addresses for source addresses:

Outline four IP addresses for destination addresses:

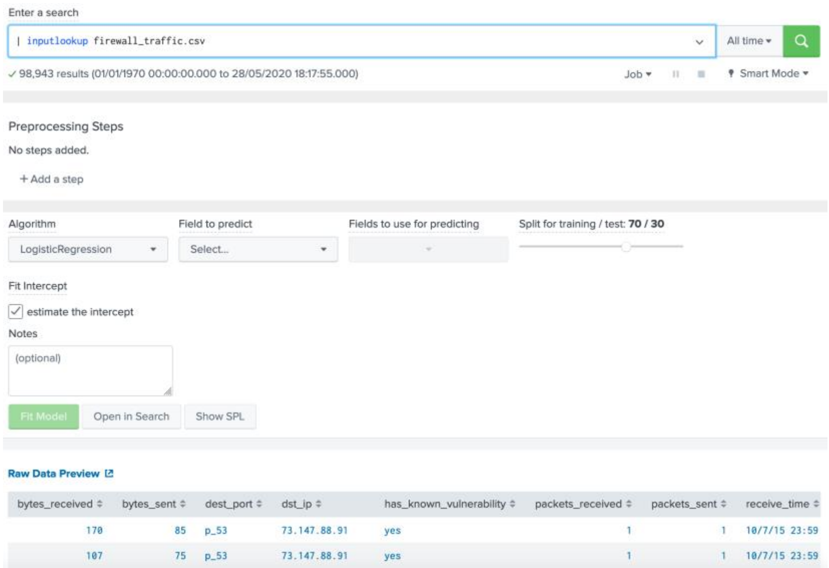


Figure 4: Defining the dataset

There are 98,943 results, which is rather large for processing so reduce it to 50,000 (Figure 5).

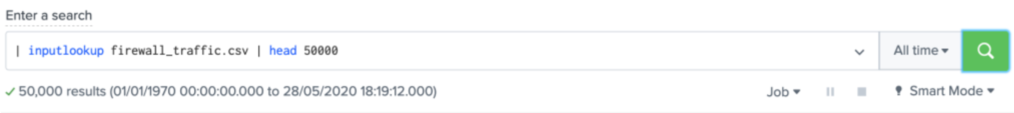


Figure 5: Filtering to 50,000 records

Next we will use Logistic Regression to predict a value for “used_by_malware” (Figure 6).

Which values are possible for the “used_by_malware” parameter?

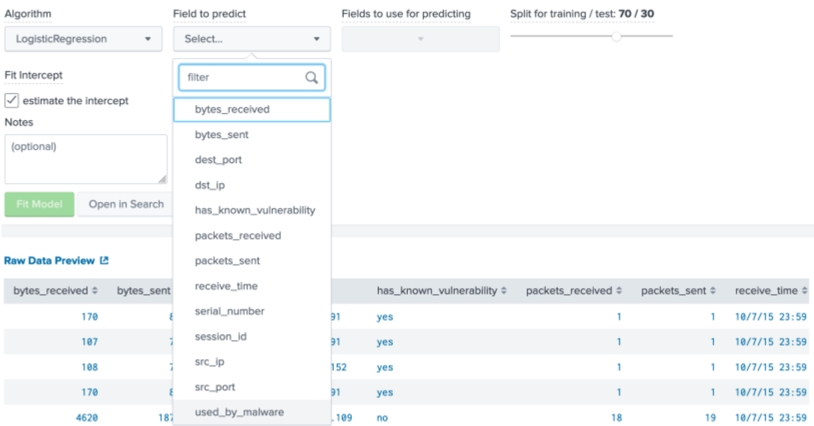


Figure 6: Predicting for "used_by_malware"

Next we shall train against all the other parameters (Figure 7). Finally we are using a 70/30 split, and 70% of training and 30% for testing the model created.

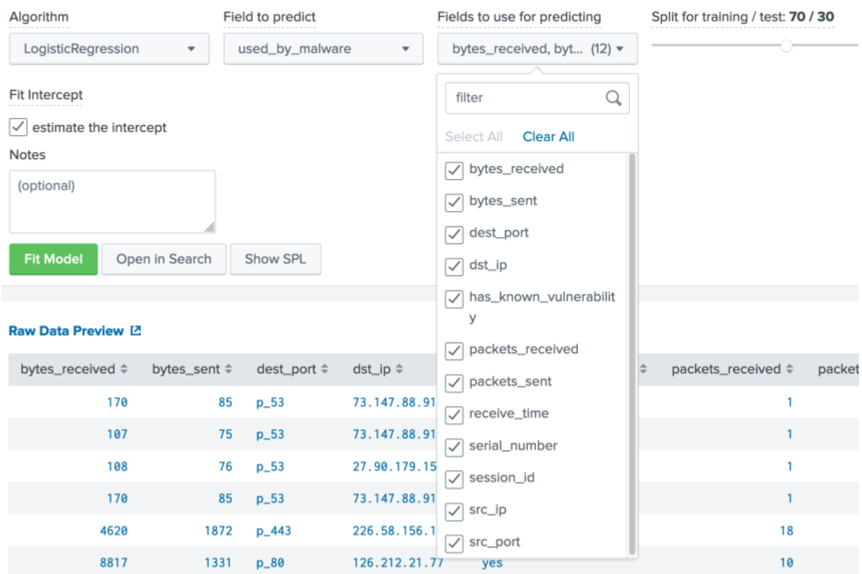


Figure 7: Fields used to predict

Finally, we select the "Fitting Model.." button, and waiting until the model is built. When complete we should see prediction data (Figure 8).

Outline the destination IP addresses for two false positives:
Outline the destination IP addresses for two true positives:

Now outline the following:

Precision: Recall: Accuracy:

F1:

Outline the confusion matrix:

used_by_malware	predicted(used_by_malware)	bytes_received	bytes_sent	dest_port	dst_ip	has_known_vulnerability
no	no	4620	1872	p_443	226.58.156.109	no
yes	no	4160	1243	p_443	47.242.134.132	yes
yes	no	507	976	p_443	204.243.248.73	yes
yes	yes	3950	2447	p_443	84.216.108.116	yes
yes	yes	3950	2447	p_443	84.216.108.116	yes
yes	yes	98	86	p_53	73.147.88.91	yes
yes	yes	98	86	p_53	73.147.88.91	yes
yes	yes	3950	2447	p_443	84.216.108.116	yes
yes	yes	456	669	p_80	72.8.163.120	yes
yes	no	572	1018	p_443	32.246.18.81	yes

Figure 8: Predictions

Precision	Recall	Accuracy	F1
0.83	0.82	0.82	0.83

Classification Results (Confusion Matrix)

Predicted actual	Predicted no	Predicted yes
no	4818 (80.7%)	1152 (19.3%)
yes	1486 (16.4%)	7580 (83.6%)

Figure 9: Confusion Matric

New Search

Save As Close

```
| inputlookup firewall_traffic.csv | head 50000 | apply "_exp_draft_0e467230935543b98e7882eebdfce34d" | `confusionmatrix
("used_by_malware","predicted(used_by_malware)")`
```

All time

✓ 2 results (before 28/05/2020 18:30:29.000) No Event Sampling Job Smart Mode

Events Patterns Statistics (2) Visualization

20 Per Page Format Preview

Predicted actual	Predicted no	Predicted yes
no	16033	3672
yes	4961	25334

Figure 10: Confusion Matric

Now click on “Open Search” in the button beside “Fit Model”:

```
| inputlookup firewall_traffic.csv | head 50000 | fit LogisticRegression fit_intercept=true "used_by_malware" from
    "bytes_received" "bytes_sent" "dest_port" "dst_ip" "has_known_vulnerability" "packets_received" "packets_sent" "receive_time"
    "serial_number" "session_id" "src_ip" "src_port" into "_exp_draft_0e467230935543b98e7882eebdfce34d"
```

Next press SHIFT-ENTER, and force the “| fit …” to move to the next line:

```
| inputlookup firewall_traffic.csv | head 50000
| fit LogisticRegression fit_intercept=true "used_by_malware" from "bytes_received" "bytes_sent" "dest_port" "dst_ip"
   "has_known_vulnerability" "packets_received" "packets_sent" "receive_time" "serial_number" "session_id" "src_ip" "src_port"
   into "_exp_draft_0e467230935543b98e7882eebdfce34d"
```

Now add:

| inputlookup firewall_traffic.csv | head 50000 | apply "_exp_draft_0e467230935543b98e7882eebdfce34d"
| multireport
[score precision_recall_fscore_support "used_by_malware" against "predicted(used_by_malware)" average=weighted
| rename fbeta_score as f1
| eval f1 = round(f1, 2)
| eval precision = round(precision, 2)
| eval recall = round(recall, 2)
| fields f1 precision recall]

[score accuracy_score "used_by_malware" against "predicted(used_by_malware)"
| eval accuracy = round(accuracy_score, 2)]

| table accuracy f1 precision recall
| stats first(*) as *

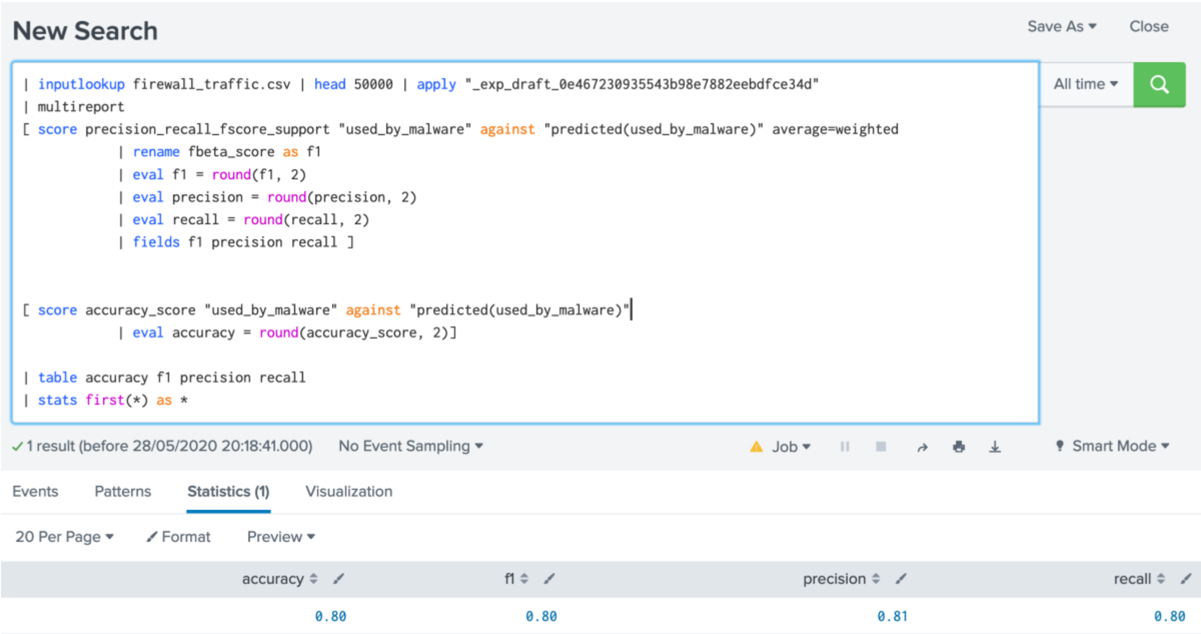
Run the result and check the output

```
| inputlookup firewall_traffic.csv | head 50000
| fit LogisticRegression fit_intercept=true "used_by_malware" from "bytes_received" "bytes_sent" "dest_port" "dst_ip"
   "has_known_vulnerability" "packets_received" "packets_sent" "receive_time" "serial_number" "session_id" "src_ip" "src_port"
   into "_exp_draft_0e467230935543b98e7882eebdfce34d"
| multireport
[ score precision_recall_fscore_support "used_by_malware" against "predicted(used_by_malware)" average=weighted
          | rename fbeta_score as f1
          | eval f1 = round(f1, 2)
          | eval precision = round(precision, 2)
          | eval recall = round(recall, 2)
          | fields f1 precision recall ]


[ score accuracy_score "used_by_malware" against "predicted(used_by_malware)"
          | eval accuracy = round(accuracy_score, 2)]

| table accuracy f1 precision recall
| stats first(*) as *
```

What are the results:



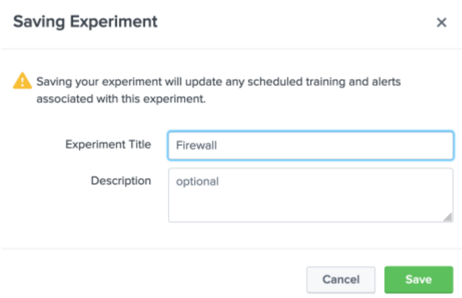
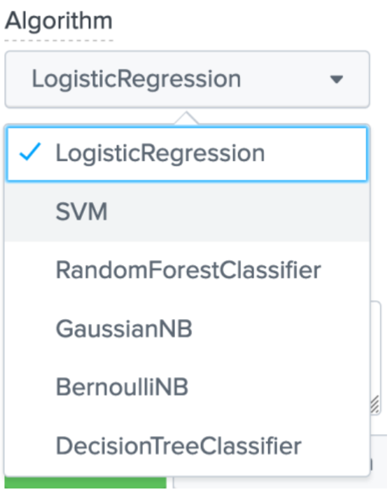


Figure 9: Saving experiment

SVM

We will now use an SVM (Support Vector Machine) model and which is a supervised learning technique. Overall, it is used to create two categories, and will try to allocate each of the training values into one category or the other. Basically, we have points in a multidimensional space, and try to create a clear gap between the categories. New values are then placed within one of the two categories. In this case we will train with SVM, and rerun the model. Now determine the following:

Precision: Recall: Accuracy: F1:



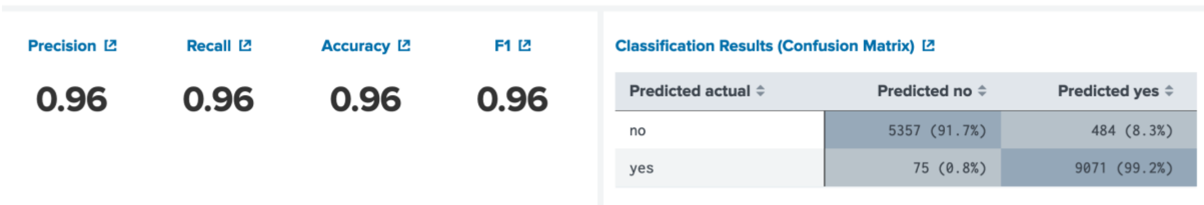
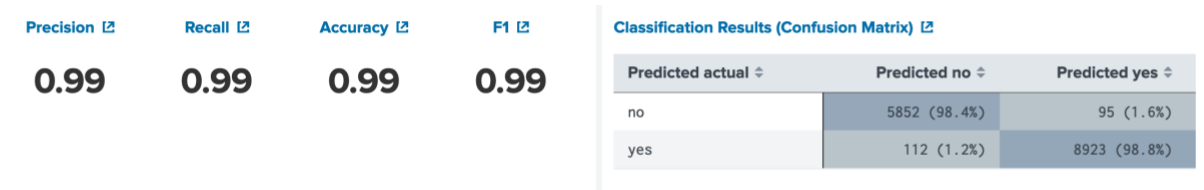


Figure 9: Saving experiment



Appendix

```
| inputlookup firewall_traffic.csv | head 50000 | apply
"_exp_draft_0e467230935543b98e7882eebdfce34d"
| table "used_by_malware", "predicted(used_by_malware)", "bytes_received" "bytes_sent"
"dest_port" "dst_ip" "has_known_vulnerability" "packets_received" "packets_sent"
"receive_time" "serial_number" "session_id" "src_ip" "src_port"
```

```
| inputlookup firewall_traffic.csv | head 50000
| fit SVM  "used_by_malware" from "bytes_received" "bytes_sent" "dest_port" "dst_ip"
"has_known_vulnerability" "packets_received" "packets_sent" "receive_time"
```

```
"serial_number" "session_id" "src_ip" "src_port" into
"_exp_draft_0e467230935543b98e7882eebdfce34d"

| inputlookup firewall_traffic.csv | head 50000
| fit RandomForestClassifier  "used_by_malware" from "bytes_received" "bytes_sent"
"dest_port" "dst_ip" "has_known_vulnerability" "packets_received" "packets_sent"
"receive_time" "serial_number" "session_id" "src_ip" "src_port" into
"_exp_draft_0e467230935543b98e7882eebdfce34d"

| inputlookup firewall_traffic.csv | head 50000
| fit GaussianNB  "used_by_malware" from "bytes_received" "bytes_sent" "dest_port"
"dst_ip" "has_known_vulnerability" "packets_received" "packets_sent" "receive_time"
"serial_number" "session_id" "src_ip" "src_port" into
"_exp_draft_0e467230935543b98e7882eebdfce34d"
```