## OneClassSVM

The OneClassSVM algorithm uses the scikit-learn OneClassSVM to fit a model from a set of features or fields for detecting anomalies and outliers, where features are expected to contain numerical values. OneClassSVM is an unsupervised outlier detection method.

For further information, see the sci-kit learn documentation:
http://scikit-learn.org/stable/modules/svm.html#kernel-functions

### Parameters

- The `kernel` parameter specifies the kernel type for using in the algorithm, where the default value is kernel is `rbf`.
    - Kernel types include: linear, rbf, poly, and sigmoid.
- You can specify the upper bound on the fraction of training error as well as the lower bound of the fraction of support vectors using the `nu` parameter, where the default value is 0.5.
- The `degree` parameter is ignored by all kernels except the polynomial kernel, where the default value is 3.
- `gamma` is the kernel co-efficient that specifies how much influence a single data instance has, where the default value is `1/ number of features`.
- The independent term of `coef0` in the kernel function is only significant if you have polynomial or sigmoid function.
- The term `tol` is the tolerance for stopping criteria.
- The `shrinking` parameter determines whether to use the shrinking heuristic.

### Syntax

```
fit OneClassSVM <fields> [into <model name>]
[kernel=<str>] [nu=<float>] [coef0=<float>]
[gamma=<float>] [tol=<float>] [degree=<int>] [shrinking=<true|false>]
```

- You can save OneClassSVM models using the `into` keyword.
- You can apply the saved model later to new data with the `apply` command.

### Syntax constraints

- After running the `fit` or `apply` command, a new field named `isNormal` is generated. This field defines whether a particular record (row) is normal (`isNormal=1`) or anomalous (`isNormal=-1`).
- You cannot inspect the model learned by OneClassSVM with the `summary` command.

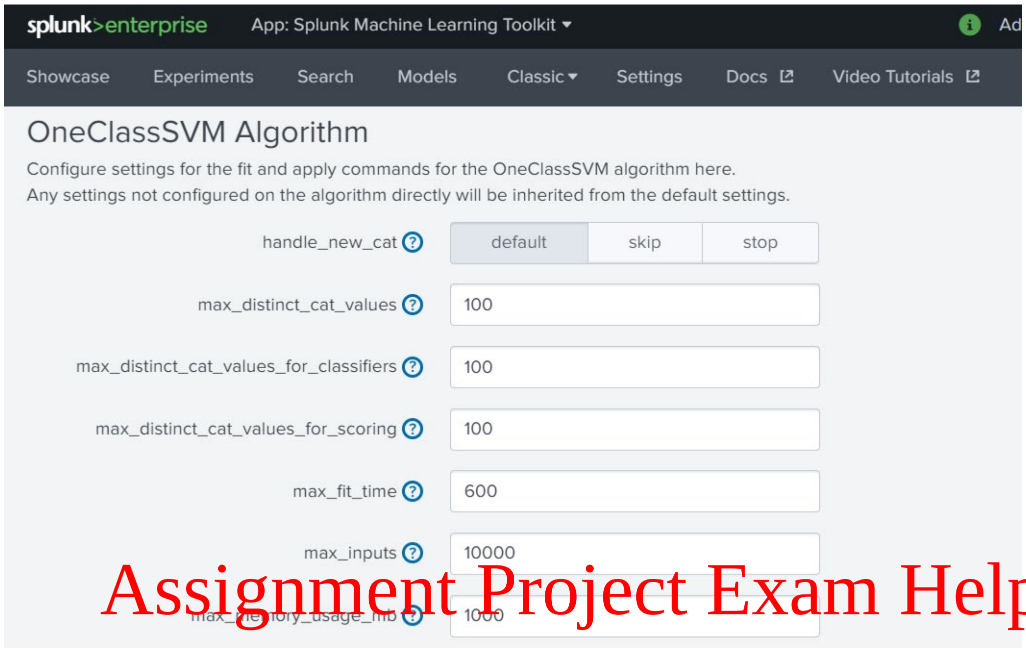
### Example

The following example uses OneClassSVM on a test set.

```
... | fit OneClassSVM * kernel="poly" nu=0.5 coef0=0.5 gamma=0.5 tol=1
degree=3 shrinking=f into
TESTMODEL_OneClassSVM
```

**Advanced settings (keep default unless you have special requirements)**

App: **Splunk Machine Learning Toolkit** -> **Settings** -> **OneClassSVM**



OneClassSVM Algorithm

Configure settings for the fit and apply commands for the OneClassSVM algorithm here.
Any settings not configured on the algorithm directly will be inherited from the default settings.

| | |
|---|---|
| handle_new_cat | [✓ default] skip stop |
| max_distinct_cat_values | 100 |
| max_distinct_cat_values_for_classifiers | 100 |
| max_distinct_cat_values_for_scoring | 100 |
| max_fit_time | 600 |
| max_inputs | 10000 |
| max_memory_usage_mb | 1000 |

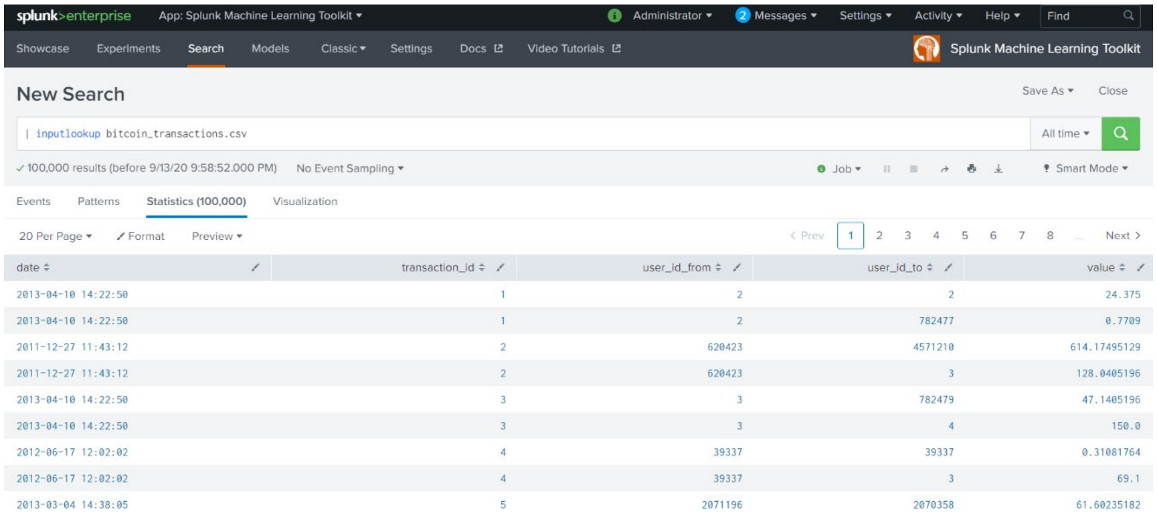
**Where to put my data (CSV) files?**

```
>> $SPLUNK_HOME\etc\apps\Splunk_ML_Toolkit\lookups
```

**Example: Bitcoin Transaction**

First, let's inspect the <bitcoin_transactions.csv> using the following command:

```
| inputlookup bitcoin_transactions.csv
```



```
| inputlookup bitcoin_transactions.csv
```

✓ 100,000 results (before 9/13/20 9:58:52.000 PM)   No Event Sampling

| date | transaction_id | user_id_from | user_id_to | value |
|---|---|---|---|---|
| 2013-04-10 14:22:50 | 1 | 2 | 2 | 24.375 |
| 2013-04-10 14:22:50 | 1 | 2 | 782477 | 0.7709 |
| 2011-12-27 11:43:12 | 2 | 620423 | 4571210 | 614.17495129 |
| 2011-12-27 11:43:12 | 2 | 620423 | 3 | 128.0405196 |
| 2013-04-10 14:22:50 | 3 | 3 | 782479 | 47.1405196 |
| 2013-04-10 14:22:50 | 3 | 3 | 4 | 150.0 |
| 2012-06-17 12:02:02 | 4 | 39337 | 39337 | 0.31081764 |
| 2012-06-17 12:02:02 | 4 | 39337 | 3 | 69.1 |
| 2013-03-04 14:38:05 | 5 | 2071196 | 2070358 | 61.60235182 |

In this dataset, we have the following 5 fields (aka. features):

`date` – Date and time of the transaction, **not** interested in this anomaly detection task

`transaction_id` – **nominal** variable, **not** interested in this anomaly detection task

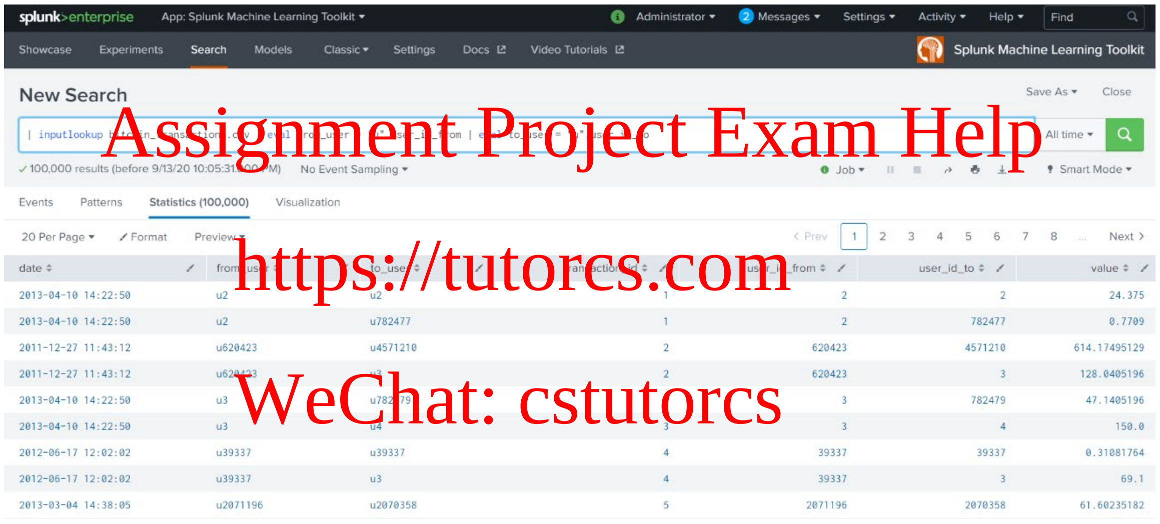
`user_id_from` – **nominal** variable, **interested** in this anomaly detection task

`user_id_to` – **nominal** variable, **interested** in this anomaly detection task

`value` – **continuous** variable, **interested** in this anomaly detection task

For all nominal variables, we need to pre-process that so that Splunk will recognize them as <nominal variables> using the following command:

```
| inputlookup bitcoin_transactions.csv
| eval from_user = "u".user_id_from | eval to_user = "u".user_id_to
```



```
| inputlookup bitcoin_transactions.csv | eval from_user = "u".user_id_from | eval to_user = "u".user_id_to
```

✓ 100,000 results (before 9/13/20 10:05:31.000 PM)   No Event Sampling

| date | from_user | to_user | transaction_id | user_id_from | user_id_to | value |
|---|---|---|---|---|---|---|
| 2013-04-10 14:22:50 | u2 | u2 | 1 | 2 | 2 | 24.375 |
| 2013-04-10 14:22:50 | u2 | u782477 | 1 | 2 | 782477 | 0.7709 |
| 2011-12-27 11:43:12 | u620423 | u4571210 | 2 | 620423 | 4571210 | 614.17495129 |
| 2011-12-27 11:43:12 | u620423 | u3 | 2 | 620423 | 3 | 128.0405196 |
| 2013-04-10 14:22:50 | u3 | u782479 | 3 | 3 | 782479 | 47.1405196 |
| 2013-04-10 14:22:50 | u3 | u4 | 3 | 3 | 4 | 150.0 |
| 2012-06-17 12:02:02 | u39337 | u39337 | 4 | 39337 | 39337 | 0.31081764 |
| 2012-06-17 12:02:02 | u39337 | u3 | 4 | 39337 | 3 | 69.1 |
| 2013-03-04 14:38:05 | u2071196 | u2070358 | 5 | 2071196 | 2070358 | 61.60235182 |

Now, we have the field `from_user` and `to_user` which have “u” as the prefix. Thus, Splunk will treat them as <nominal variables>.

Next, we can fit the OneClassSVM algorithm using the following commands:

```
| inputlookup bitcoin_transactions.csv
| eval from_user = "u".user_id_from | eval to_user = "u".user_id_to
| fit OneClassSVM "from_user" "to_user" "value" kernel="rbf" gamma=0.33
nu=0.02 shrinking=True into "OneSVM_Bitcoin"
```

Where the hyper-parameters are set as: `kernel="rbf" gamma=0.33 nu=0.02`

After the model is trained, Splunk will show you 10,000 lines of data which is evaluated by the trained model, which means the **<u>displayed data is not the full dataset</u>**.

Therefore, to get the full processed dataset, we need to apply our trained model using:

```
| inputlookup bitcoin_transactions.csv | eval from_user = "u".user_id_from |
eval to_user = "u".user_id_to
```

```
| apply "OneSVM_Bitcoin" | outputlookup bitcoin_svm.csv
```

Optionally, we can save the processed full dataset as <bitcoin_svm.csv> to the following path:

```
>> $SPLUNK_HOME\etc\apps\Splunk_ML_Toolkit\lookups
```

Now, we can use the following command to load the processed data, and see some statistics of the anomaly detection:

```
| inputlookup bitcoin_svm.csv | chart count by isNormal
```

The result shows 61749 transactions are abnormal (`isNormal=-1`) and 38251 transactions are normal (`isNormal=1`), which seems to be too sensitive.

Now, we retrain our model using different parameters: `kernel="rbf" gamma=0.33 nu=0.001`

The full commands will be as below:

*Step 1:*

```
| inputlookup bitcoin_transactions.csv
```

```
| eval from_user = "u".user_id_from | eval to_user = "u".user_id_to
```

```
| fit OneClassSVM "from_user" "to_user" "value" kernel="rbf" gamma=0.33
nu=0.001 shrinking=True into "OneSVM_Bitcoin_2"
```

*Step 2:*

```
| inputlookup bitcoin_transactions.csv | eval from_user = "u".user_id_from |
eval to_user = "u".user_id_to
```

```
| apply "OneSVM_Bitcoin_2" | outputlookup bitcoin_svm_2.csv
```

*Step 3:*

```
| inputlookup bitcoin_svm_2.csv | chart count by isNormal
```

The result shows 2488 transactions are abnormal (`isNormal=-1`) and 97512 transactions are normal (`isNormal=1`), which seems to be acceptable.