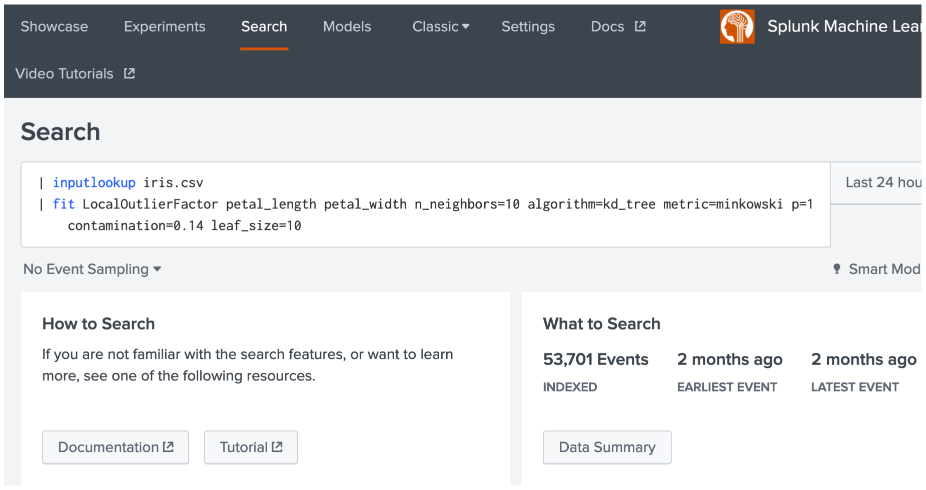
# 1 ML Tutorial 1

Figure 1 outlines the access to Splunk for machline learning. For this select the link below and enter your username and password:

http://asecuritysite.com:8000



```
| inputlookup iris.csv
| fit LocalOutlierFactor petal_length petal_width n_neighbors=10 algorithm=kd_tree metric=minkowski p=1
    contamination=0.14 leaf_size=10
```

How to Search

If you are not familiar with the search features, or want to learn more, see one of the following resources.

What to Search

53,701 Events INDEXED | 2 months ago EARLIEST EVENT | 2 months ago LATEST EVENT

Figure 1: Machine Learning

## 1.1 Anomaly Detection

In this section we will analyse some of the models used to detect anomalies. Figure 2 outlines the mapping of the Iris dataset. The dataset contains four types of flower, and which have different dimensions for the petal length, petal width, septal length and septal width.

1. Select the Search tab, and in the search facility, enter the following:

```
| inputlookup iris.csv
| fit LocalOutlierFactor petal_length petal_width n_neighbors=10
    algorithm=kd_tree metric=minkowski p=1 contamination=0.14 leaf_size
    =10
```

How many records are there?

What are the fields within the dataset:

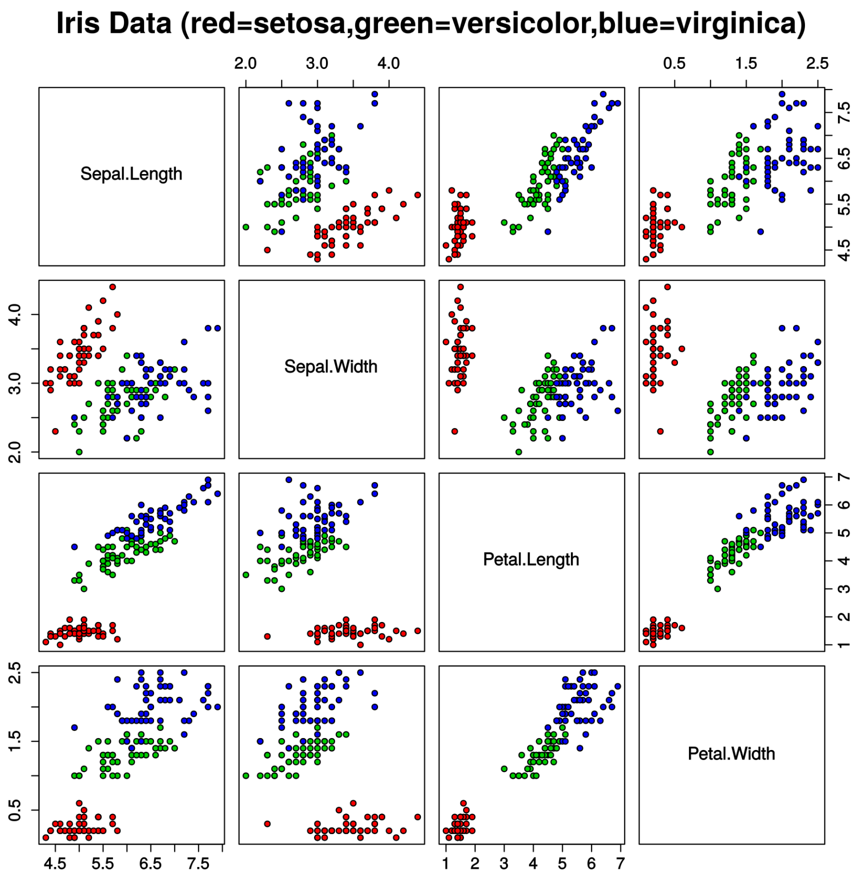


Figure 2: Iris dataset
[1]

What is the value assigned for outliers:

What is the value assigned for inliers:

2. Next we will run a One Class SVM model for anomaly detection:

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

How many records are there?

Using petal_length and petal_width for the identification of an anomaly, determine the details of one flower with an anomaly:

Using petal_length and petal_width for the identification of an anomaly, determine the details of one flower with an anomaly:

3. We now will use a new data setup (call_center.csv) and run the Density Function method:

```
| inputlookup call_center.csv
| fit DensityFunction count by "source" into mymodel Link.
```

How many records are there?

What is the format of the stored data fields:

Outline one possible anomaly:

## 1.2 Prediction (Categories)

4. We will go back to our Iris dataset, and run AutoPrediction:

```
| inputlookup iris.csv
| fit AutoPrediction random_state=42 petal_length from * max_features=0.1
    into auto_classify_model test_split_ratio=0.3 random_state=42
```

Can you now train for sepal_length using sepal_width and petal_length. Outline one prediction for sepal_length (and the error from the actual value):

5. Now apply the BernoulliNB prediction model:

```
| inputlookup iris.csv
| fit BernoulliNB petal_length from * into TESTMODEL_BernoulliNB alpha
   =0.5 binarize=0 fit_prior=f
```

Can you now train for petal_length using sepal_width and species. Outline one prediction for petal_length (and the error from the actual value):

6. Now apply the DecisionTreeClassifier prediction model:

```
| inputlookup iris.csv
| fit DecisionTreeClassifier petal_length from * into sla_ MOD
```

Can you now train for petal_length using petal_width and species. Outline one prediction for petal_length (and the error from the actual value):

7. Now apply the GaussianNB:

```
| inputlookup iris.csv
| fit GaussianNB petal_length from * into MOD.
```

Can you now train for petal_length using petal_width and species. Outline one prediction for petal_length (and the error from the actual value):

8. Now apply the GradientBoostingClassifier:

```
| inputlookup iris.csv
| fit GradientBoostingClassifier petal_length from * into MOD
```

Can you now train for petal_length using petal_width and species. Outline one prediction for petal_length (and the error from the actual value):

### 1.3 Prediction (Numeric

9. In the following we will use the track_day.csv data source. Perform an AutoPrediction model on batteryVoltage using longitudeGForce, speed and verticalGForce:

```
| inputlookup track_day.csv
| fit AutoPrediction batteryVoltage target_type=numeric test_split_ratio
    =0.7 from * into PM
```

Outline some of the features within the track_day data set:

Outline one prediction for batteryVoltage (and the error from the actual value):

10. Perform an DecisionTreeRegressor model on batteryVoltage using longitudeGForce, speed and verticalGForce:

```
| inputlookup track_day.csv
| fit DecisionTreeRegressor batteryVoltage from * into PM
```

Outline one prediction for batteryVoltage (and the error from the actual value):

11. Perform an ElasticNet model on batteryVoltage using longitudeGForce, speed and verticalGForce:

```
| inputlookup track_day_missing.csv
| fit ElasticNet batteryVoltage from * into EN.
```

Outline one prediction for batteryVoltage (and the error from the actual value):

12. Perform an GradientBoostingRegressor model on batteryVoltage using longitudeGForce, speed and verticalGForce:

```
| inputlookup track_day_missing.csv
| fit GradientBoostingRegressor batteryVoltage from * into GB
```

Outline one prediction for batteryVoltage (and the error from the actual value):

## 1.4 Clustering

13. In the following we will use the iris.csv data source. Perform Birch clustering model on petal_length for three clusters:

```
| inputlookup iris.csv
| fit Birch petal_length k=3 partial_fit=true into MOD
```

Outline one value from each cluster:

14. In the following we will use the iris.csv data source. Perform DBSCAN clustering model on petal_length for three clusters:

```
| inputlookup iris.csv
| fit DBSCAN petal_length min_samples=4
```

How many clusters have been created:

Now train on species, how many clusters are created, and how is the clustering setup:

15. In the following we will use the iris.csv data source. Perform GMeans clustering model on petal_length for three clusters:

```
| inputlookup iris.csv
| fit GMeans petal_length random_state=42 into MOD3
```

How many clusters have been created:

16. In the following we will use the iris.csv data source. Perform GMeans clustering model on petal_length for three clusters:

```
| inputlookup iris.csv
| fit KMeans petal_length k=3 into MOD4
```

How many clusters have been created:

## 1.5 Feature Extraction

17. In the following we will use the FieldSelector method to determine the best feature selector for batteryVoltage given engineCoolantTemperature, engineSpeed, lateralGForce,longitudeGForce, speed:

```
| inputlookup track_day.csv
| fit FieldSelector batteryVoltage from engineCoolantTemperature,
    engineSpeed, lateralGForce,longitudeGForce, speed type=numeric
```

Best feature:

18. Now we find the best feature for batteryVoltage for engineSpeed, lateralGForce and longitudeGForce:

```
| inputlookup track_day.csv
| fit FieldSelector batteryVoltage from engineSpeed, lateralGForce,
    longitudeGForce, speed type=numeric
```

Best feature:

19. Now for vechicleType (and which is a category), determine its best match for engineCoolantTemperature, engineSpeed, lateralGForce,longitudeGForce, and speed:

```
| inputlookup track_day.csv
| fit FieldSelector vehicleType from engineCoolantTemperature,
    engineSpeed, lateralGForce,longitudeGForce, speed type=categorical
```

Best feature:

20. Within the best feature, we can use the HashingVectorization to analyse the difference between strings (using N-grams). Run the following search:

```
| inputlookup passwords.csv
| fit HashingVectorizer Passwords ngram_range=1-2 k=10
```

What is the data used in the dataset

For the tokens tried, which ones have been successful in finding password matches:

21. The ICA (Independent component analysis) method is used to reduce the number of dimensions in the data. Run the following search:

```
| inputlookup track_day.csv
| fit ICA batteryVoltage, engineSpeed, engineCoolantTemperature
    n_components=2 as IC
```

What do you observe from the output:

What happens when you change *n_components* to 1:

22. We can do the same reduction with PCA. Run the following search:

```
| inputlookup track_day.csv
| fit KernelPCA batteryVoltage, engineSpeed, engineCoolantTemperature k=2
     gamma=0.001 as PCA
```

What do you observe from the output:

What happens when you change the *k* to 1:

23. NPR (Normalized Perlich Ratio) is useful in converting categorical fields into numeric values:

```
| inputlookup track_day.csv
| fit NPR vehicleType from engineSpeed as npr01
```

What do you observe from the output:

24. Principal Component Analysis (PCA) reduce the number of fields in the data. Run the following search:

```
| inputlookup track_day.csv
| fit PCA engineCoolantTemperature, engineSpeed, lateralGForce,speed k=2
    as pca01
```

What do you observe from the output:

25. TF-IDF (Term Frequency-Inverse Document Frequency) converts raw text data into a matrix, and can be used to find words within documents. Run the following search:

```
| inputlookup track_day.csv
| fit TFIDF vehicleType ngram_range=1-2 max_df=0.6 min_df=0.2 stop_words=
    english as tf01
```

What do you observe from the output:

## 1.6 Feature Extraction

26. We can Imputer to replace data that is missing. Enter the following search:

```
| inputlookup track_day_missing.csv
| fit Imputer batteryVoltage
```

What data has been replaced:

We can also add a strategy or mean, median and most_frequent, what are the results when you add each of these:

27. RobustScaler scales to median and interquartile range to 0 and 1. It thus reduces the bias caused by large data ranges swamping smaller values.

```
| inputlookup track_day_missing.csv
| fit RobustScaler *
```

Outline the result:

28. Now try the ScandardScalar:

```
| inputlookup track_day_missing.csv
| fit StandardScaler *
```

Outline the result:

## 1.7 Forecasting

29. A common method we have is to forecast over time. The StateSpaceForecast method is based on Kalman filters. Run the following query:

```
| inputlookup app_usage.csv
| fit StateSpaceForecast CRM ERP Expenses holdback=12 into SF
```

Outline the result:

30. Finally, we can use the ARIMA model for forecasting. Perform the following search:

```
| inputlookup logins.csv
| fit ARIMA _time logins holdback=0 conf_interval=95 order=0-0-0
    forecast_k=5 as AR
```

Outline the result:

## 1.8 References

[1] https://en.wikipedia.org/wiki/Iris_flower_data_set