**predict**

**Description**

The `predict` command forecasts values for one or more sets of time-series data. The command can also fill in missing data in a time-series and provide predictions for the next several time steps.

The `predict` command provides confidence intervals for all of its estimates. The command adds a predicted value and an upper and lower 95th percentile range to each event in the time-series. See the **Usage** section in this topic.

**Syntax**

```
predict <field-list> [AS <newfield>] [<predict_options>]
```

**Required arguments**

```
<field-list>
```

*Syntax:* `<field>...`

*Description:* The names of the fields for the variable that you want to predict. You can specify one or more fields.

**Optional arguments**

```
<newfield>
```

*Syntax:* `<string>`

*Description:* Renames the fields that are specified in the `<field-list>`. You do not need to rename every field that you specify in the `<field-list>`. However, for each field that you want to rename, you must specify a separate `AS <newfield>` clause.

```
<predict_options>
```

*Syntax:* `algorithm=<algorithm_name> | correlate_field=<field> | future_timespan=<number> | holdback=<number> | period=<number> | suppress=<bool> | lowerXX=<field> | upperYY=<field>`

*Description:* Options you can specify to control the predictions. You can specify one or more options, in any order. Each of these options is described in the **Predict options** section.

**Predict options**

```
algorithm
```

*Syntax:* `algorithm= LL | LLT | LLP | LLP5 | LLB | BiLL`

*Description:* Specify the name of the forecasting algorithm to apply. LL, LLT, LLP, and LLP5 are univariate algorithms. LLB and BiLL are bivariate algorithms. All the algorithms are variations based on the Kalman filter. Each algorithm expects a minimum number of data points. If not enough effective data points are supplied, an error message is displayed. For instance, the field itself might have more than enough data points, but the number of effective data points might be small if the `holdback` value that you specify is large.

*Default:* LLP5

<table>
<thead>
<tr>
<th>Algorithm option</th>
<th>Algorithm type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL</td>
<td>Local level</td>
<td>A univariate model with no trends and no seasonality. Requires a minimum of 2 data points. The LL algorithm is the simplest algorithm and computes the levels of the time series. For example, each new state equals the previous state, plus</td>
</tr>
</tbody>
</table>
the Gaussian noise.

<table>
<thead>
<tr>
<th>Algorithm option</th>
<th>Algorithm type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLT</td>
<td>Local level trend</td>
<td>A univariate model with trend, but no seasonality. Requires a minimum of 3 data points.</td>
</tr>
<tr>
<td>LLP</td>
<td>Seasonal local level</td>
<td>A univariate model with seasonality. The number of data points must be at least twice the number of periods, using the period attribute. The LLP algorithm takes into account the cyclical regularity of the data, if it exists. If you know the number of periods, specify the period argument. If you do not set the period, this algorithm tries to calculate it. LLP returns an error message if the data is not periodic.</td>
</tr>
<tr>
<td>LLP5</td>
<td>Combines LLT and LLP models for its prediction.</td>
<td>If the time series is periodic, LLP5 computes two predictions, one using LLT and the other using LLP. The algorithm then takes a weighted average of the two values and outputs that as the prediction. The confidence interval is also based on a weighted average of the variances of LLT and LLP.</td>
</tr>
<tr>
<td>LLB</td>
<td>Bivariate local level</td>
<td>A bivariate model with no trends and no seasonality. Requires a minimum of 2 data points. LLB uses one set of data to make predictions for another. For example, assume it uses dataset Y to make predictions for dataset X. If holdback=10, LLB takes the last 10 data points of Y to make predictions for the last 10 data points of X.</td>
</tr>
<tr>
<td>BiLL</td>
<td>Bivariate local level</td>
<td>A bivariate model that predicts both time series simultaneously. The covariance of the two series is taken into account.</td>
</tr>
</tbody>
</table>

**correlate**

*Syntax:* correlate=<field>*

*Description:* Specifies the time series that the LLB algorithm uses to predict the other time series. Required when you specify the LLB algorithm. Not used for any other algorithm.

*Default:* None

**future_timespan**

*Syntax:* future_timespan=<num>*

*Description:* Specifies how many future predictions the predict command will compute. This number must be a non-negative number. You would not use the future_timespan option if algorithm=LLB.

*Default:* 5

**holdback**

*Syntax:* holdback=<num>*

*Description:* Specifies the number of data points from the end that are not to be used by the predict command. Use in conjunction with the future_timespan argument. For example, 'holdback=10 future_timespan=10' computes the predicted values for the last 10 values in the data set. You can then judge how accurate the predictions are by checking whether the actual data point values fall into the predicted confidence intervals.

*Default:* 0

**lowerXX**

*Syntax:* lower<int>=<field>*

*Description:* Specifies a percentage for the confidence interval and a field name to use for the lower confidence interval curve. The int value is a percentage that specifies the confidence level. The integer must be a number between 0 and 100. The field value is the field name.

*Default:* The default confidence interval is 95%. The default field name is 'lower95(prediction(X))' where X is the name of the field to be predicted.

**period**

*Syntax:* period=<num>*
**Description:** Specifies the length of the time period, or recurring cycle, in the time series data. The number must be at least 2. The LLP and LLP5 algorithms attempt to compute the length of time period if no value is specified. If you specify the `span` argument with the `timechart` command, the unit that you specify for `span` is the unit used for `period`. For example, if your search is `...|timechart span=1d foo2| predict foo2 period=3`. The spans are 1 day and the period for the predict is 3 days. Otherwise, the unit for the time period is a data point. For example, if there are a thousand events, then each event is a unit. If you specify `period=7`, that means the data recycles after every 7 data points, or events.

**Default:** None

**suppress**

**Syntax:** suppress=<field>

**Description:** Used with the multivariate algorithms. Specifies one of the predicted fields to hide from the output. Use `suppress` when it is difficult to see all of the predicted visualizations at the same time.

**Default:** None

**upperYY**

**Syntax:** upper<int>=<field>

**Description:** Specifies a percentage for the confidence interval and a field name to use for the upper confidence interval curve. The `<int>` value is a percentage that specifies the confidence level. This must be a number between 0 and 100. The `<field>` value is the field name.

**Default:** The default confidence interval is 95%. The default field name is 'upper95(prediction(X))' where X is the name of the field to be predicted.

**Confidence intervals**

The lower and upper confidence interval parameters default to `lower95` and `upper95`. These values specify a confidence interval where 95% of the predictions are expected fall.

It is typical for some of the predictions to fall outside the confidence interval.

- The confidence interval does not cover 100% of the predictions.
- The confidence interval is about a probabilistic expectation and results do not match the expectation exactly.

**Usage**

**Command sequence requirement**

The `predict` command must be preceded by the `timechart` command. The `predict` command requires time series data. See the `Examples` section for more details.

**How it works**

The `predict` command models the data by stipulating that there is an unobserved entity which progresses through time in different states.

To predict a value, the command calculates the best estimate of the state by considering all of the data in the past. To compute estimates of the states, the command hypothesizes that the states follow specific linear equations with Gaussian noise components.

Under this hypothesis, the least-squares estimate of the states are calculated efficiently. This calculation is called the Kalman filter, or Kalman-Bucy filter. For each state estimate, a confidence interval is obtained. The estimate is not a point
estimate. The estimate is a range of values that contain the observed, or predicted, values.

The measurements might capture only some aspect of the state, but not necessarily the whole state.

**Missing values**

The `predict` command can work with data that has missing values. The command calculates the best estimates of the missing values.

Do not remove events with missing values. Removing the events might distort the periodicity of the data. Do not specify `cont=false` with the `timechart` command. Specifying `cont=false` removes events with missing values.

**Specifying span**

The unit for the `span` specified with the `timechart` command must be seconds or higher. The `predict` command cannot accept subseconds as an input when it calculates the period.

**Examples**

1. **Predict future access**

This example uses the sample data from the Search Tutorial but should work with any format of Apache web access log. To try this example on your own Splunk instance, you must download the sample data and follow the instructions to get the tutorial data into Splunk. Use the time range **All time** when you run the search.

Predict future access based on the previous access numbers that are stored in Apache web access log files. Count the number of access attempts using a span of 1 day.

```
sourcetype=access_combined_* | timechart span=1d count(file) as count | predict count
```

The results appear on the Statistics tab. Click the Visualization tab. If necessary change the chart type to a Line Chart.
2. Predict future purchases for a product

This example uses the sample data from the Search Tutorial. To try this example on your own Splunk instance, you must download the sample data and follow the instructions to get the tutorial data into Splunk. Use the time range All time when you run the search.

Chart the number of purchases made daily for a specific product.

```
sourcetype=access_* action=purchase arcade | timechart span=1d count
```

- This example searches for all purchases events, defined by the action=purchase for the arc, and pipes those results into the timechart command.
- The span=1day argument buckets the count of purchases into daily chunks.

The results appear on the Statistics tab and look something like this:

<table>
<thead>
<tr>
<th>_time</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018-06-11</td>
<td>17</td>
</tr>
<tr>
<td>2018-06-12</td>
<td>63</td>
</tr>
<tr>
<td>2018-06-13</td>
<td>94</td>
</tr>
<tr>
<td>2018-06-14</td>
<td>82</td>
</tr>
<tr>
<td>2018-06-15</td>
<td>63</td>
</tr>
<tr>
<td>2018-06-16</td>
<td>76</td>
</tr>
<tr>
<td>2018-06-17</td>
<td>70</td>
</tr>
<tr>
<td>2018-06-18</td>
<td>72</td>
</tr>
</tbody>
</table>

Add the predict command to the search to calculate the prediction for the number of purchases of the Arcade games that might be sold in the near future.

```
sourcetype=access_* action=purchase arcade | timechart span=1d count | predict count
```

The results appear on the Statistics tab. Click the Visualization tab. If necessary change the chart type to a Bar Chart.
3. **Predict the values using the default algorithm**

Predict the values of foo using the default LLP5 algorithm, an algorithm that combines the LLP and LLT algorithms.

```plaintext
... | timechart span="1m" count AS foo | predict foo
```

4. **Predict multiple fields using the same algorithm**

Predict multiple fields using the same algorithm. The default algorithm in this example.

```plaintext
... | timechart ... | predict foo1 foo2 foo3
```

5. **Specify different upper and lower confidence intervals**

When specifying confidence intervals, the upper and lower confidence interval values do not need to match. This example predicts 10 values for a field using the LL algorithm, holding back the last 20 values in the data set.

```plaintext
... | timechart span="1m" count AS foo | predict foo AS foobar algorithm=LL upper90=high lower97=low future_timespan=10 holdback=20
```

6. **Predict the values using the LLB algorithm**

This example illustrates the LLB algorithm. The foo3 field is predicted by correlating it with the foo2 field.

```plaintext
... | timechart span="1m" count(x) AS foo2 count(y) AS foo3 | predict foo3 AS foobar algorithm=LLB correlate=foo2 holdback=100
```

7. **Omit the last 5 data points and predict 5 future values**

In this example, the search abstains from using the last 5 data points and makes 5 future predictions. The predictions correspond to the last 5 values in the data. You can judge how accurate the predictions are by checking whether the observed values fall into the predicted confidence intervals.

```plaintext
... | timechart ... | predict foo holdback=5 future_timespan=5
```

8. **Predict multiple fields using the same algorithm and the same future_timespan and holdback**

Predict multiple fields using the same algorithm and same future_timespan and holdback.

```plaintext
... | timechart ... | predict foo1 foo2 foo3 algorithm=LLT future_timespan=15 holdback=5
```

9. **Specify aliases for fields**

Use aliases for the fields by specifying the AS keyword for each field.

```plaintext
... | timechart ... | predict foo1 AS foobar1 foo2 AS foobar2 foo3 AS foobar3 algorithm=LLT future_timespan=15 holdback=5
```
10. *Predict multiple fields using different algorithms and options*

Predict multiple fields using different algorithms and different options for each field.

```bash
... | timechart ... | predict foo1 algorithm=LL future_timespan=15 foo2 algorithm=LLP period=7 future_timespan=7
```

11. *Predict multiple fields using the BiLL algorithm*

Predict values for foo1 and foo2 together using the bivariate algorithm BiLL.

```bash
... | timechart ... | predict foo1 foo2 algorithm=BiLL future_timespan=10
```

**See also**

trendline, x11

**Answers**

Have questions? Visit Splunk Answers and see what questions and answers the Splunk community has about using the predict command.