Experiment Results

Results for Miami

In [2]:
```
import warnings
warnings.filterwarnings('ignore')

%run helper_functions.py
%matplotlib inline
```

Lets review the top 10 results for Miami, ordered by Mean Square Error (ascending)

In [125]:
```
city='Miami'
results_top_10 = get_results(city=city, top_hm_results=10)
results_top_10
```

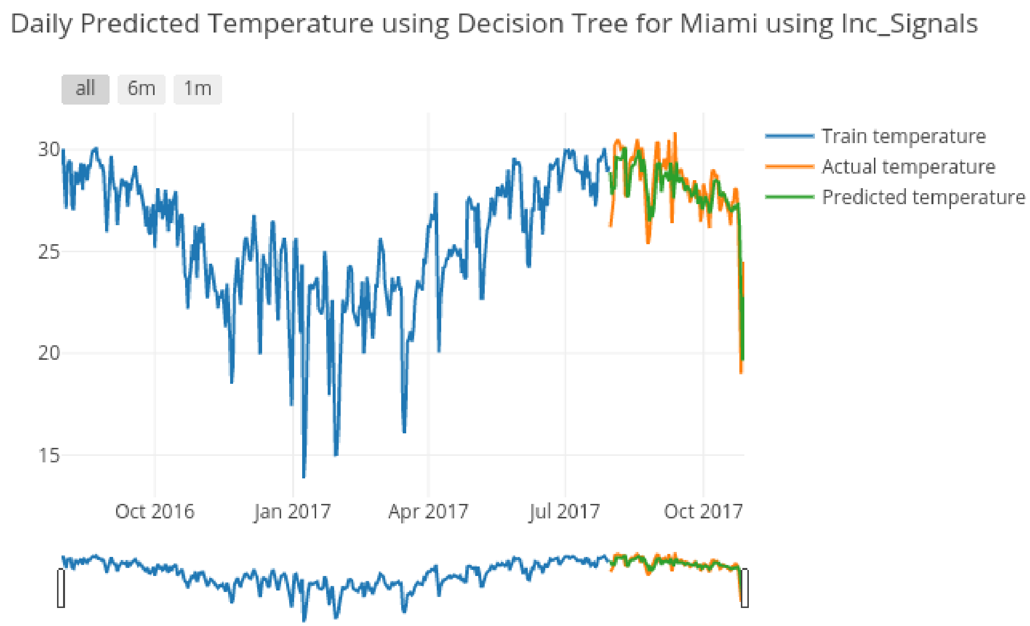
Out[125]:

	RUN_ID	DATETIME	MODEL_NAME	CITY	FEATURE_TYPE	HOST_MACHINE	MODEL_PARAMETERS	MODEL_RESULTS	MEAN_SQUARED_ERROR
1	30	2018-08-14 19:56:00.644376	RANDOM FOREST	Miami	Inc_Signals	DESKTOP-KN40C32	{'max_depth': 8, 'n_estimators': 50, 'Info': {...	{'features': ['temperature_lag1', 'temperature...	1.168790
2	36	2018-08-14 19:56:51.638468	RANDOM FOREST	Miami	Enhanced_Signals	DESKTOP-KN40C32	{'max_depth': 8, 'n_estimators': 50, 'Info': {...	{'features': ['temperature_lag1', 'temperature...	1.168790
3	24	2018-08-14 19:55:11.164461	RANDOM FOREST	Miami	Basic	DESKTOP-KN40C32	{'max_depth': 8, 'n_estimators': 50, 'Info': {...	{'features': ['temperature_lag1', 'temperature...	1.206302
4	17	2018-08-14 19:44:46.805353	DECISION TREE	Miami	Enhanced_Signals	DESKTOP-KN40C32	{'max_depth': 8, 'Info': {'feature_set_type': ...	{'features': ['temperature_lag1', 'temperature...	1.556905
5	12	2018-08-14 19:43:48.176029	DECISION TREE	Miami	Inc_Signals	DESKTOP-KN40C32	{'max_depth': 8, 'Info': {'feature_set_type': ...	{'features': ['temperature_lag1', 'temperature...	1.577366
6	6	2018-08-14 19:42:49.124681	DECISION TREE	Miami	Basic	DESKTOP-KN40C32	{'max_depth': 8, 'Info': {'feature_set_type': ...	{'features': ['temperature_lag1', 'temperature...	1.833555
7	54	2018-08-14 20:40:54.336628	RNN	Miami	Enhanced_Signals	DESKTOP-KN40C32	{'epochs': 50, 'Info': {'feature_set_type': 'E...	{'features': ['temperature_lag1', 'temperature...	2.012860
8	42	2018-08-14 20:35:28.587513	RNN	Miami	Basic	DESKTOP-KN40C32	{'epochs': 50, 'Info': {'feature_set_type': 'B...	{'features': ['temperature_lag1', 'temperature...	2.016882
9	48	2018-08-14 20:38:20.610680	RNN	Miami	Inc_Signals	DESKTOP-KN40C32	{'epochs': 50, 'Info': {'feature_set_type': 'I...	{'features': ['temperature_lag1', 'temperature...	2.093093

Random forest appears to be the clear winner, lets take a look at that chart.

In [73]:
```
display_results(results_top_10.head(1), chart_type='predict')
```

Experiment #30 - run by DESKTOP-KN40C32 on 2018-08-14 19:56:00.644376



We can see the forecast follows the trend but does not have the predictive power to track some of the more volatile swings in the signals.

Lets examine the top 10 features across the model runs for this location in a pivot table form.

In [138]:

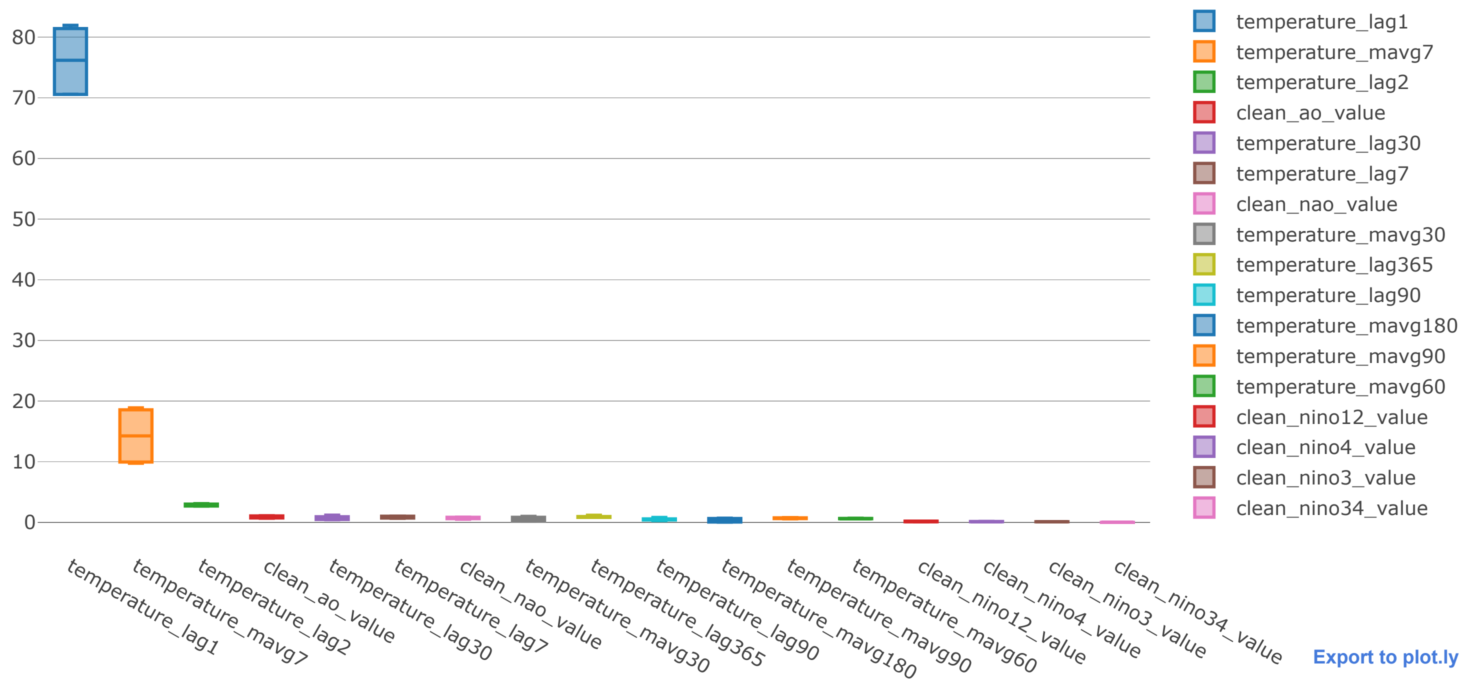
```
features_df = get_feature_importances(results_top_10)
features_df.pivot_table(index=['FEATURE'], columns=['MODEL_NAME','FEATURE_TYPE'], values="IMPORTANCE")
```

Out[138]:

MODEL_NAME	DECISION TREE			RANDOM FOREST		
FEATURE_TYPE	Basic	Enhanced_Signals	Inc_Signals	Basic	Enhanced_Signals	Inc_Signals
FEATURE						
clean_ao_value	NaN	0.762803	0.697029	NaN	1.069394	1.069394
clean_nao_value	NaN	0.705866	0.518667	NaN	0.844053	0.844053
clean_nino12_value	NaN	0.205030	0.101543	NaN	NaN	NaN
clean_nino34_value	NaN	0.002508	0.000472	NaN	NaN	NaN
clean_nino3_value	NaN	0.113445	0.100821	NaN	NaN	NaN
clean_nino4_value	NaN	0.149838	0.105239	NaN	NaN	NaN
temperature_lag1	81.951355	81.333226	81.412405	71.035461	70.556534	70.556534
temperature_lag2	2.944449	2.956676	3.089420	3.020370	2.698790	2.698790
temperature_lag30	0.612916	0.475372	0.456209	1.198484	0.934467	0.934467
temperature_lag365	0.842632	0.837027	0.969628	1.184014	0.823453	0.823453
temperature_lag7	1.021827	0.703784	0.710618	1.039751	0.860722	0.860722
temperature_lag90	0.399428	0.296508	0.412470	0.818218	0.580242	0.580242
temperature_mavg180	0.607840	0.055693	0.061311	0.679769	NaN	NaN
temperature_mavg30	0.235605	0.239946	0.236853	0.995305	0.842836	0.842836
temperature_mavg60	0.630974	0.594589	0.659230	NaN	NaN	NaN
temperature_mavg7	9.960203	9.951108	9.751084	18.888060	18.563162	18.563162
temperature_mavg90	0.792771	0.616582	0.717001	0.631938	NaN	NaN

Lets look a boxplot of this data to see how the distributions look across our experiments for this location.

In [123]:
```
traces = create_boxplot_traces_for_features(features_df)
iplot(traces)
```



Export to plot.ly »

Lets review the means for the features

In [153]:
```
features_df = get_feature_importances(results_top_10)
pivot_df = features_df.pivot_table(index=['FEATURE'], columns=[], values="IMPORTANCE", aggfunc= [np.mean])
pivot_df
```

Out[153]:

	mean
	IMPORTANCE
FEATURE	
clean_ao_value	0.899655
clean_nao_value	0.728160
clean_nino12_value	0.153286
clean_nino34_value	0.001490
clean_nino3_value	0.107133
clean_nino4_value	0.127538
temperature_lag1	76.140919
temperature_lag2	2.901416
temperature_lag30	0.768653
temperature_lag365	0.913368
temperature_lag7	0.866237
temperature_lag90	0.514518
temperature_mavg180	0.351153
temperature_mavg30	0.565563
temperature_mavg60	0.628264
temperature_mavg7	14.279463
temperature_mavg90	0.689573

Now lets review the top 5 only

In [155]:
```
pivot_df.columns = pivot_df.columns.get_level_values(0)
pivot_df.sort_values(['mean'], ascending=False).head(5)
```

Out[155]:

	mean
FEATURE	
temperature_lag1	76.140919
temperature_mavg7	14.279463
temperature_lag2	2.901416
temperature_lag365	0.913368
clean_ao_value	0.899655

In [ ]: