In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

from prophet import Prophet
```

In [2]:
```
data = "Rev/ecom_data.csv"
```

In [3]:
```
df = pd.read_csv("ecom_data1.csv", parse_dates=["InvoiceDay"])
df
```

Out[3]:

	SalesOrder	SKU	Description	UnitPrice	CustomerID	Channel	State	InvoiceDay	Sales	Quantity
0	580636	22474	SPACEBOY TV DINNER TRAY	1.95	16746	Mailing	IL	2011-12-05	31.20	16
1	581426	70006	LOVE HEART POCKET WARMER	0.79	17757	Organic Social	WA	2011-12-08	2.37	3
2	575063	22697	GREEN REGENCY TEACUP AND SAUCER	2.95	16764	Display	TX	2011-11-08	8.85	3
3	544065	20726	LUNCH BAG WOODLAND	1.65	14346	Organic Social	TX	2011-02-15	13.20	8
4	568896	85049E	SCANDINAVIAN REDS RIBBONS	1.25	16361	Store	NY	2011-09-29	52.50	42
...	...	...	...	...	...	...	...	...	...	...
406824	573774	85099F	JUMBO BAG STRAWBERRY	1.74	16029	Store	NM	2011-11-01	1485.96	854
406825	573361	23263	SET OF 3 WOODEN HEART DECORATIONS	1.25	14456	Email	OH	2011-10-30	7.50	6
406826	565748	22739	RIBBON REEL CHRISTMAS SOCK BAUBLE	1.65	14156	Email	GA	2011-09-06	33.00	20
406827	577504	21733	RED HANGING HEART T-LIGHT HOLDER	2.95	14159	SEO	NY	2011-11-20	88.50	30
406828	554085	23208	LUNCH BAG VINTAGE LEAF DESIGN	1.65	16945	Organic Social	NC	2011-05-22	13.20	8

406829 rows × 10 columns

In [4]:
```
df = df.drop(columns=['SKU', 'Description','State', 'Quantity', 'UnitPrice'])
df
```

Out[4]:

	SalesOrder	CustomerID	Channel	InvoiceDay	Sales
0	580636	16746	Mailing	2011-12-05	31.20
1	581426	17757	Organic Social	2011-12-08	2.37
2	575063	16764	Display	2011-11-08	8.85
3	544065	14346	Organic Social	2011-02-15	13.20
4	568896	16361	Store	2011-09-29	52.50
...	...	...	...	...	...
406824	573774	16029	Store	2011-11-01	1485.96
406825	573361	14456	Email	2011-10-30	7.50
406826	565748	14156	Email	2011-09-06	33.00
406827	577504	14159	SEO	2011-11-20	88.50
406828	554085	16945	Organic Social	2011-05-22	13.20

406829 rows × 5 columns

In [5]:
```
df=df.astype({"CustomerID": "category"})
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 406829 entries, 0 to 406828
Data columns (total 5 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   SalesOrder  406829 non-null  object
 1   CustomerID  406829 non-null  category
 2   Channel     406829 non-null  object
 3   InvoiceDay  406829 non-null  datetime64[ns]
 4   Sales       406829 non-null  float64
dtypes: category(1), datetime64[ns](1), float64(1), object(2)
memory usage: 13.4+ MB
```

In [6]:
```
df=df.sort_values(["InvoiceDay"])
```

In [7]:
```
df= df[df.Sales > 0.00]
```

In [8]:
```
df=df.rename(columns={"InvoiceDay": "ds", "Sales": "y"})
df
```

Out[8]:

	SalesOrder	CustomerID	Channel	ds	y
274668	536500	17377	SEO	2010-12-01	53.55
265079	536520	14729	Store	2010-12-01	6.96
2498	536542	16456	Organic Social	2010-12-01	24.75
200022	536488	17897	Organic Social	2010-12-01	1.65
38935	536366	17850	Organic Social	2010-12-01	9.25
...	...	...	...	...	...
313261	581501	12985	Organic Social	2011-12-09	17.55
143963	581578	12713	Store	2011-12-09	106.25
73537	581496	16558	Store	2011-12-09	106.25
61943	581476	12433	Email	2011-12-09	56.88
216935	581501	12985	SEO	2011-12-09	61.25

397854 rows × 5 columns

In [9]:
```
m = Prophet()
m.fit(df)
```

```
INFO:prophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True to override this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
```

Out[9]:
```
<prophet.forecaster.Prophet at 0x20157b16588>
```

In [10]:
```
future = m.make_future_dataframe(periods=90)
future.tail()
```

Out[10]:

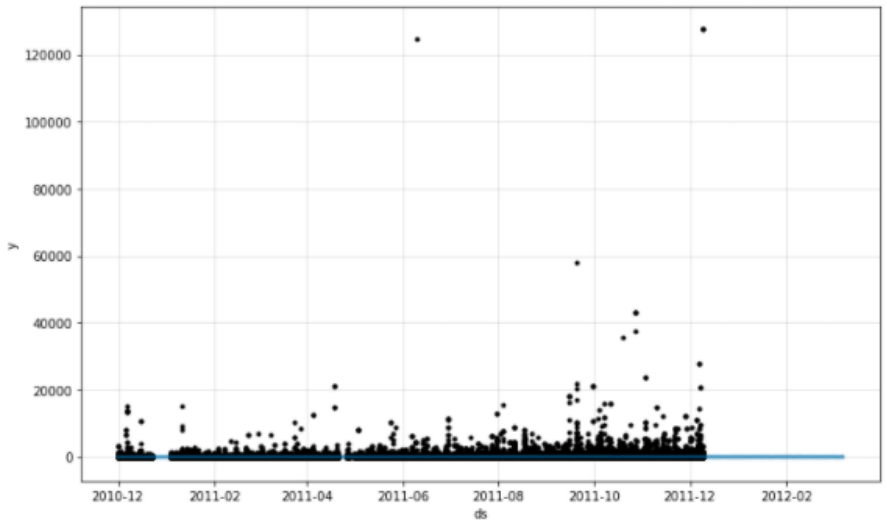
	ds
390	2012-03-04
391	2012-03-05
392	2012-03-06
393	2012-03-07
394	2012-03-08

In [11]:
```
forecast = m.predict(future)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

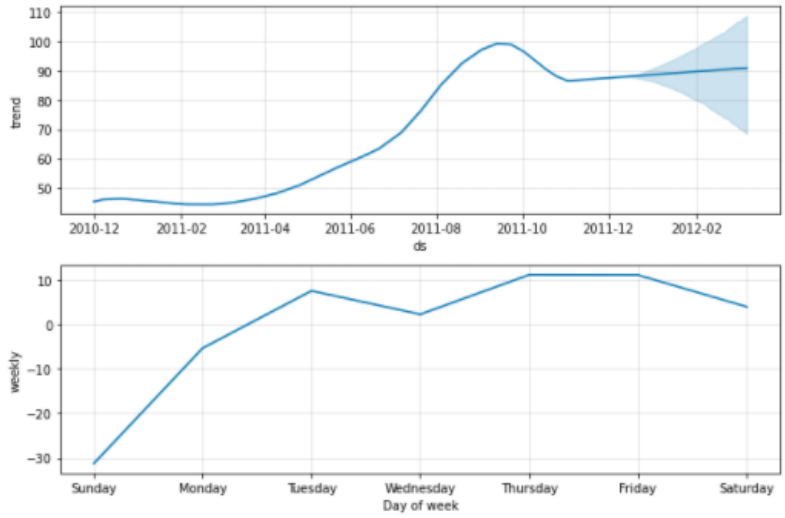
Out[11]:

	ds	yhat	yhat_lower	yhat_upper
390	2012-03-04	59.576619	-647.880598	686.983589
391	2012-03-05	85.608468	-604.002043	717.972534
392	2012-03-06	98.553392	-546.892453	734.932877
393	2012-03-07	93.305035	-590.986188	816.393416
394	2012-03-08	102.230663	-541.184046	757.025455

In [12]:
```
fig1 = m.plot(forecast)
```



In [13]:
```
fig2 = m.plot_components(forecast)
```



In [14]:
```
from prophet.plot import plot_plotly, plot_components_plotly

plot_plotly(m, forecast)
```