TABLEAU CONFERENCE

TABLEAU CONFERENCE

Welcome

#TC18

Building Data Science Applications with TabPy and R

Nathan Mannheimer

Product Manager

Advanced Analytics

TABLEAU CONFERENCE

Who Am I?

- Product Manager for Advanced Analytics
- Lecturer in Data Science and HCDE at University of Washington
- Former high school teacher in Japan and NCAA swimmer

“Hiding within those mounds of data is knowledge that could change the life of a patient, or change the world.

Atul Butte

Stanford

Session Goals

Introduce Tableau’s external analytics integrations

Explore real data science use cases

Learn how to adapt analysis scripts for Tableau

Build self-service interactive dashboards to share insights

Who is this Session For?

TABLEAU CONFERENCE

Data Scientist/Analyst

Where does Tableau fit into a data science and advanced analytics workflow and how can we most effectively share findings with business partners?

Business Data Explorers

How can we increase our cooperation and knowledge share with advanced analytics teams and put data insights into action?

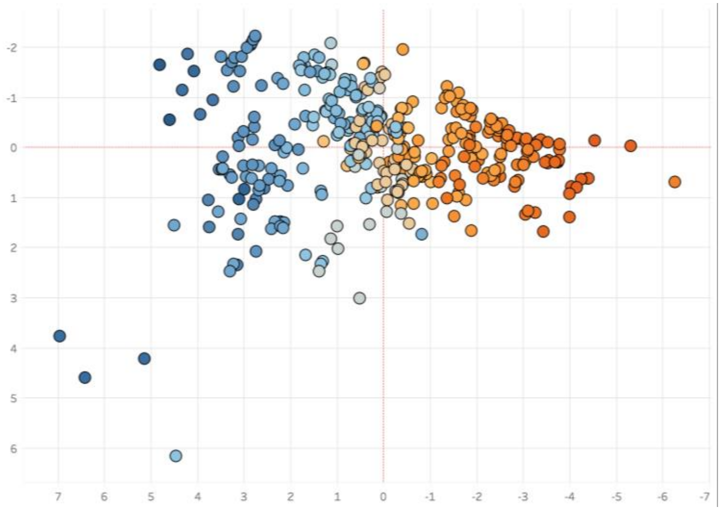
Agenda

Connecting to External Services

1. Sharing Interactive Exploratory Analysis
2. Self-Service Time Series Forecasting
3. Building and Deploying a Credit Classification Application

External Services Workflow

TABLEAU CONFERENCE



PCA Component 1 Cars

Results are computed along Table (across).

```
SCRIPT_REAL("import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

df = pd.DataFrame({'mpg':_arg1,'Cyl':_arg2,'Cost':_arg3,'EngSize':_arg4,'HP':_arg5,'Len':_arg6,'Width':_arg7})
scale = StandardScaler()
dat = scale.fit_transform(df)

n_comp = len(df.columns)
pca = PCA(n_components = n_comp)
comps = pca.fit_transform(dat)

return list(comps[:,_arg8[0]])",
SUM([City MPG]),
SUM([Cyl]),
SUM([Dealer Cost]),
SUM([Engine Size]),
SUM([HP]),
SUM([Len]),
SUM([Width]),
[Selected PCA Component 1])
```

Default Table Calculation

The calculation is valid. 2 Dependencies Apply OK

python

R

Table Calculation

Model Query

Compute Using

Table (across)

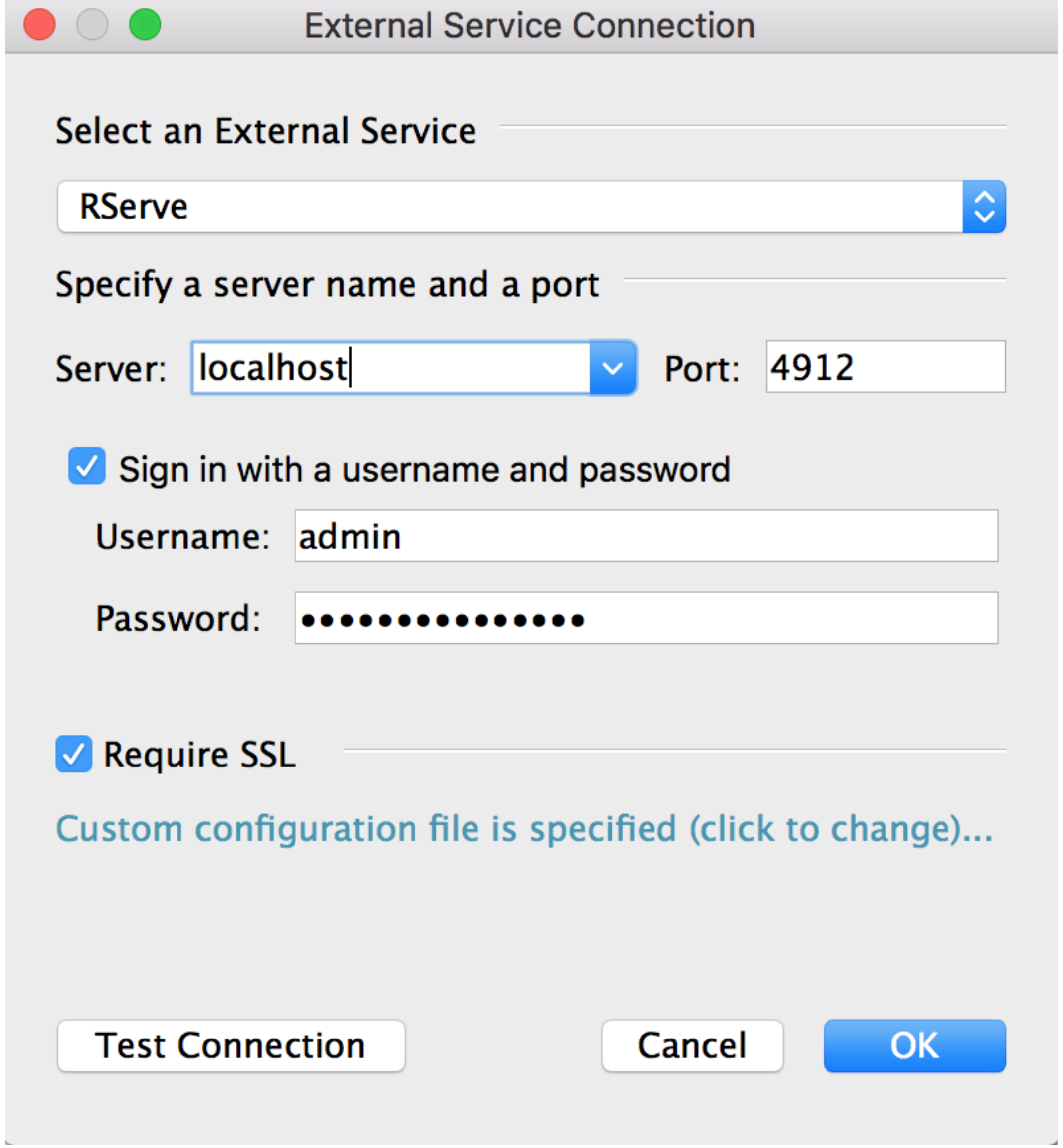
Cell

At the level

Restarting every

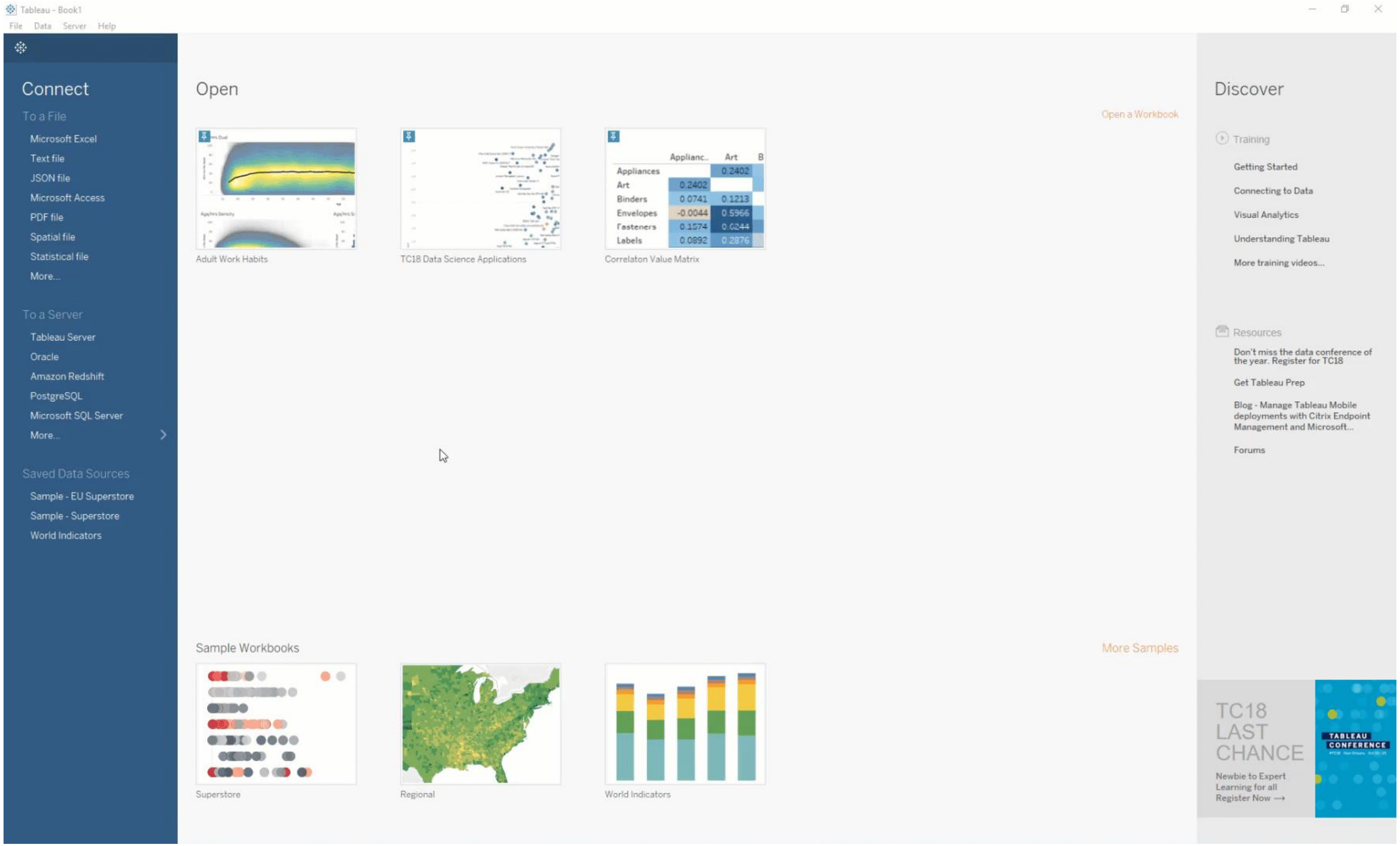
Connecting to R or Python

Connecting to an External Service



- **Supported Connections**
  - Rserve
  - TabPy/MATLAB
- **Connection Information:**
  - Specify Service Type (New!)
  - Choose Host and Port
- **Security:**
  - Authenticate with Username/Password
  - Set up encryption with SSL Cert (New!)

Connecting to an External Service

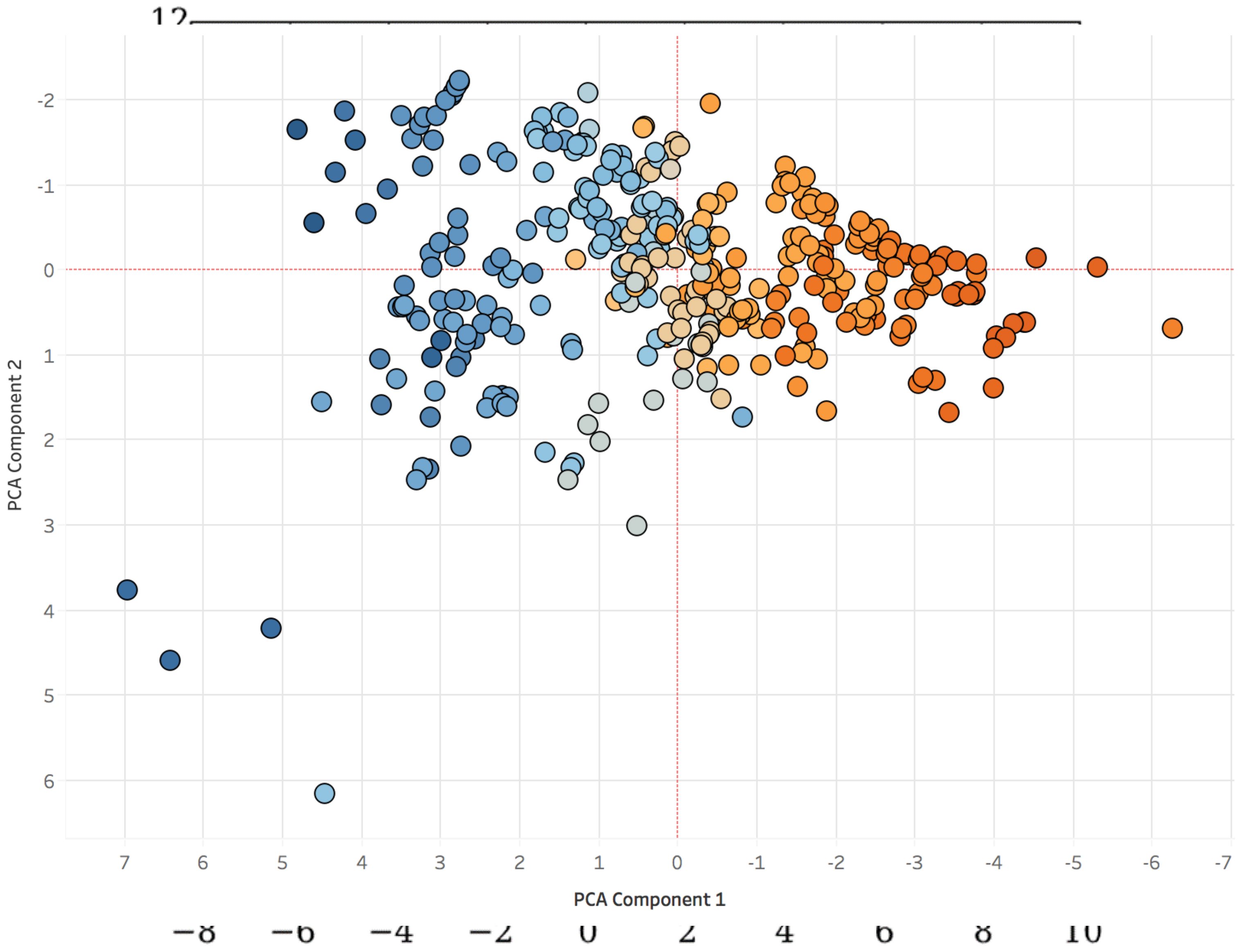


Sharing Interactive Exploratory Analysis

TABLEAU CONFERENCE

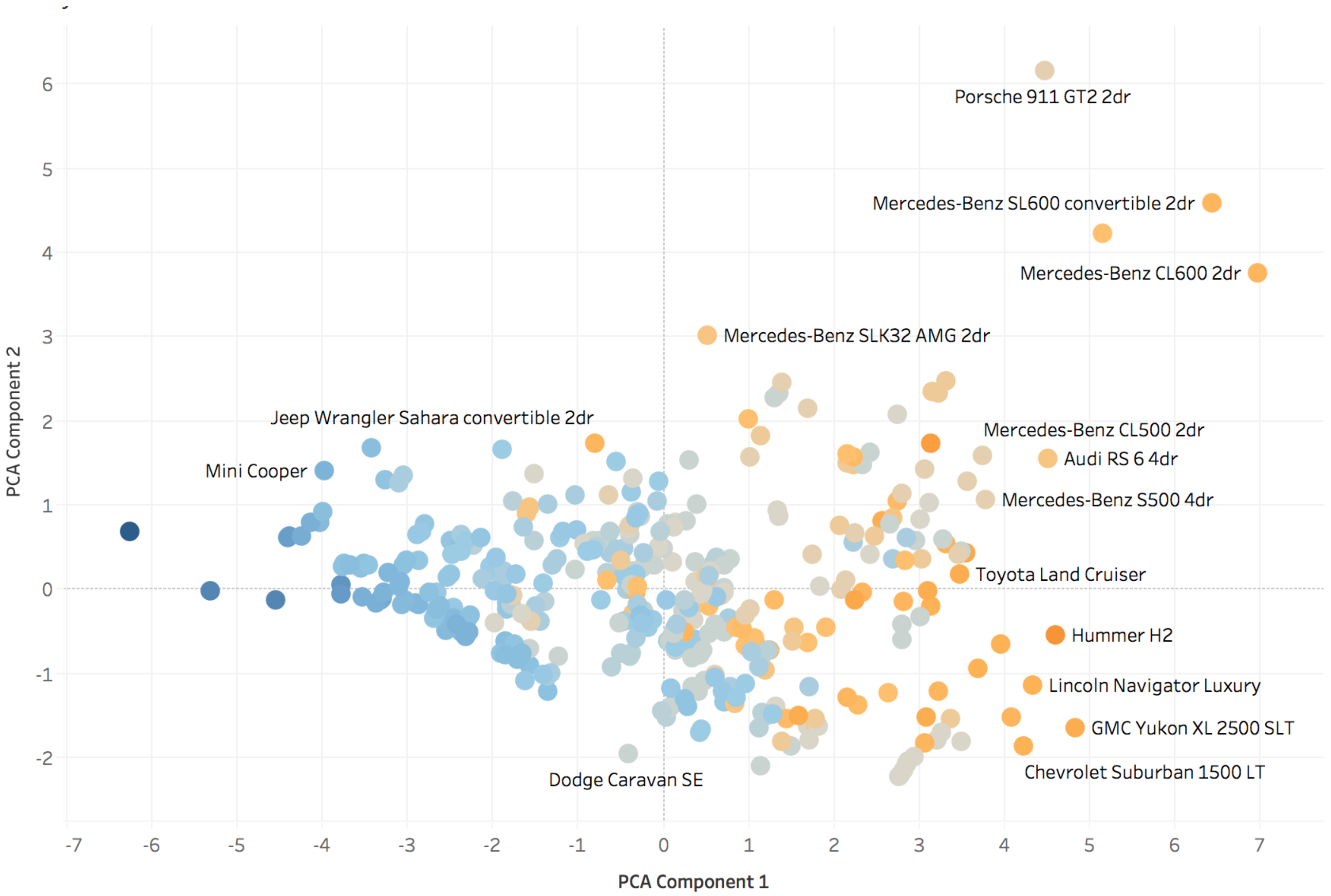
User Story – Dynamic Customer Analysis

- **Question:**
  - What customers have similar attributes across dozens or hundreds of categories?
  - Who stands out from the group?
- **Answer:**
  - Decompose data into a two dimensional visualization.
  - Explore dynamically using parameters and filters.



Answer - Presenting Exploratory Analysis

TABLEAU CONFERENCE



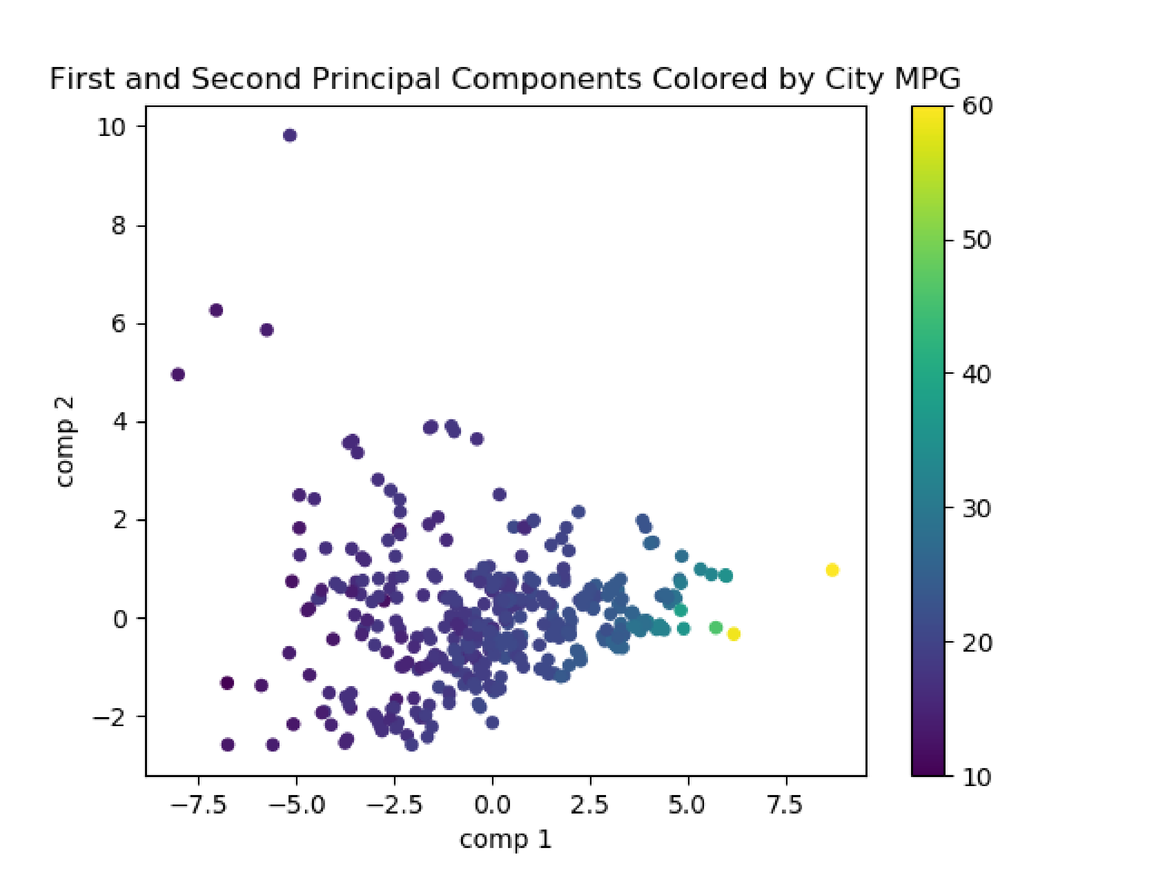
- **Visualizing PCA:**
  - Converting a python script for Tableau
  - Handling data and aggregation
  - Building an interactive dashboard
- **Further Exploration:**
  - Using parameters
  - Using filters

Directly From Python

TABLEAU CONFERENCE

```
import pandas as pd

from sklearn.decomposition import PCA

from sklearn.preprocessing import StandardScaler

df = pd.read_csv('cars.csv')
```

```
scale = StandardScaler()

dat = scale.fit_transform(df)

pca = PCA(n_components = len(df.columns))

comps = pca.fit_transform(dat)

df = pd.DataFrame(comps, columns=["comp 1","comp 2","comp 3"])
```




```
df.plot(x="comp 1",y="comp 2", kind='scatter', c=cars['City_MPG'], colormap='viridis', legend=False, colorbar=True, title='First and Second Principal Components Colored by City MPG')

plt.show()
```

Tableau Calculation

PCA Component 1 | Cars

Results are computed along Table (across).

```
SCRIPT_REAL("import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

df = pd.DataFrame({'mpg':_arg1,'Cyl':_arg2,'Cost':_arg3,'EngSize':_arg4,'HP':_arg5,'Len':_arg6,'Width':_arg7})
scale = StandardScaler()
dat = scale.fit_transform(df)

n_comp = len(df.columns)
pca = PCA(n_components = n_comp)
comps = pca.fit_transform(dat)

return list(comps[:,_arg8[0]])",
SUM([City MPG]),
SUM([Cyl]),
SUM([Dealer Cost]),
SUM([Engine Size]),
SUM([HP]),
SUM([Len]),
SUM([Width]),
[Selected PCA Component 1])
```

Default Table Calculation

The calculation is valid. | 2 Dependencies | Apply | OK

Edit Parameter [Selected PCA Component 1]

Name: Selected PCA Component 1 | Comment >>

Properties

- Data type: Integer
- Current value: 0
- Display format: Automatic
- Allowable values: All / List / Range (Range selected)

Range of values

- Minimum: 0 | Set from Parameter
- Maximum: 6 | Set from Field
- Step size: 1

OK | Cancel

Fully Adapted Code

TABLEAU CONFERENCE

```
SCRIPT_REAL( "import pandas as pd

from sklearn.decomposition import PCA

from sklearn.preprocessing import StandardScaler



df = pd.read_csv(cars.csv)
df = pd.DataFrame({'mpg':_arg1,'Cyl':_arg2,'Cost':_arg3,'EngSize':_arg4,'HP':_arg5,'Len':_arg6,'Width':_arg7})

scale = StandardScaler()

dat = scale.fit_transform(df)

n_comp = len(df.columns)

pca = PCA(n_components = n_comp)



comps = pca.fit_transform(dat)

df.plot(x="comp_1", y="comp_2", kind='scatter', c=cars['City_MPG'], colormap='viridis', legend=False,
colorbar=True, title='First and Second Principal Components Colored by City MPG')
return list(comps[:,_arg8[0]])",

SUM([City MPG]), SUM([Cyl]), SUM([Dealer Cost]), SUM([Engine Size]), SUM([HP]), SUM([Len]), SUM([Width]),
[Selected PCA Component 1])
```

```
SCRIPT_REAL("import pandas as pd

from sklearn.decomposition import PCA

from sklearn.preprocessing import StandardScaler



df = pd.DataFrame({'mpg':_arg1,'Cyl':_arg2,'Cost':_arg3,'EngSize':_arg4,'HP':_arg5,'Len':_arg6,'Width':_arg7})

scale = StandardScaler()

dat = scale.fit_transform(df)

n_comp = len(df.columns)

pca = PCA(n_components = n_comp)



comps = pca.fit_transform(dat)

return list(comps[:,_arg8[0]])",

SUM([City MPG]), SUM([Cyl]), SUM([Dealer Cost]), SUM([Engine Size]), SUM([HP]), SUM([Len]), SUM([Width]),
[Selected PCA Component 1])
```

R PCA Code

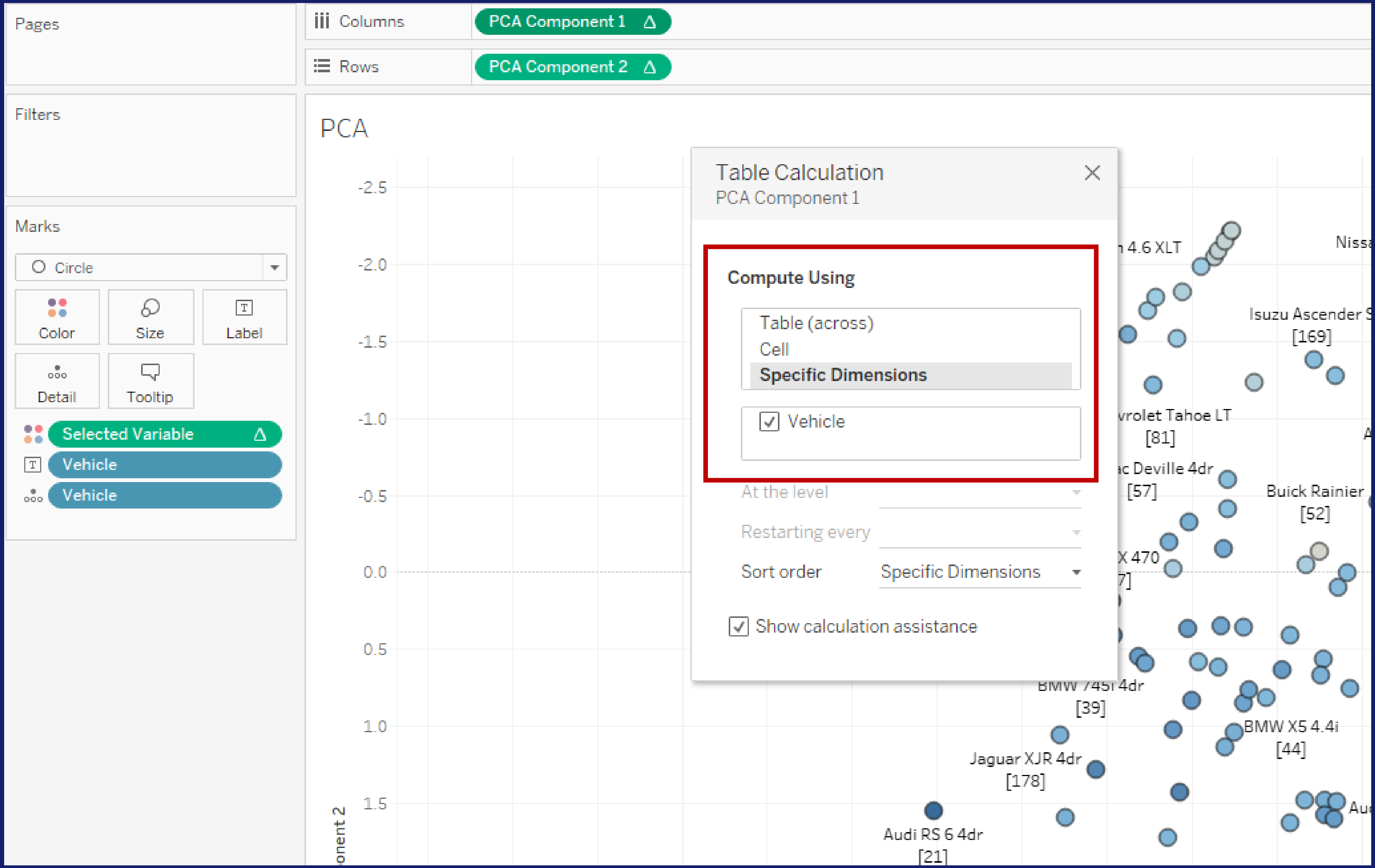
```
SCRIPT_REAL(
"princomp(data.frame(.arg1,.arg2,.arg3,.arg4,.arg5,.arg6,.arg7), cor
= TRUE)$score[,.arg8[1]]",
SUM([City MPG]),
SUM([Cyl]),
SUM([Dealer Cost]),
SUM([Engine Size]),
SUM([HP]),
SUM([Len]),
SUM([Width]),
[Selected PCA Component 1])
```

Let’s Take a Look!

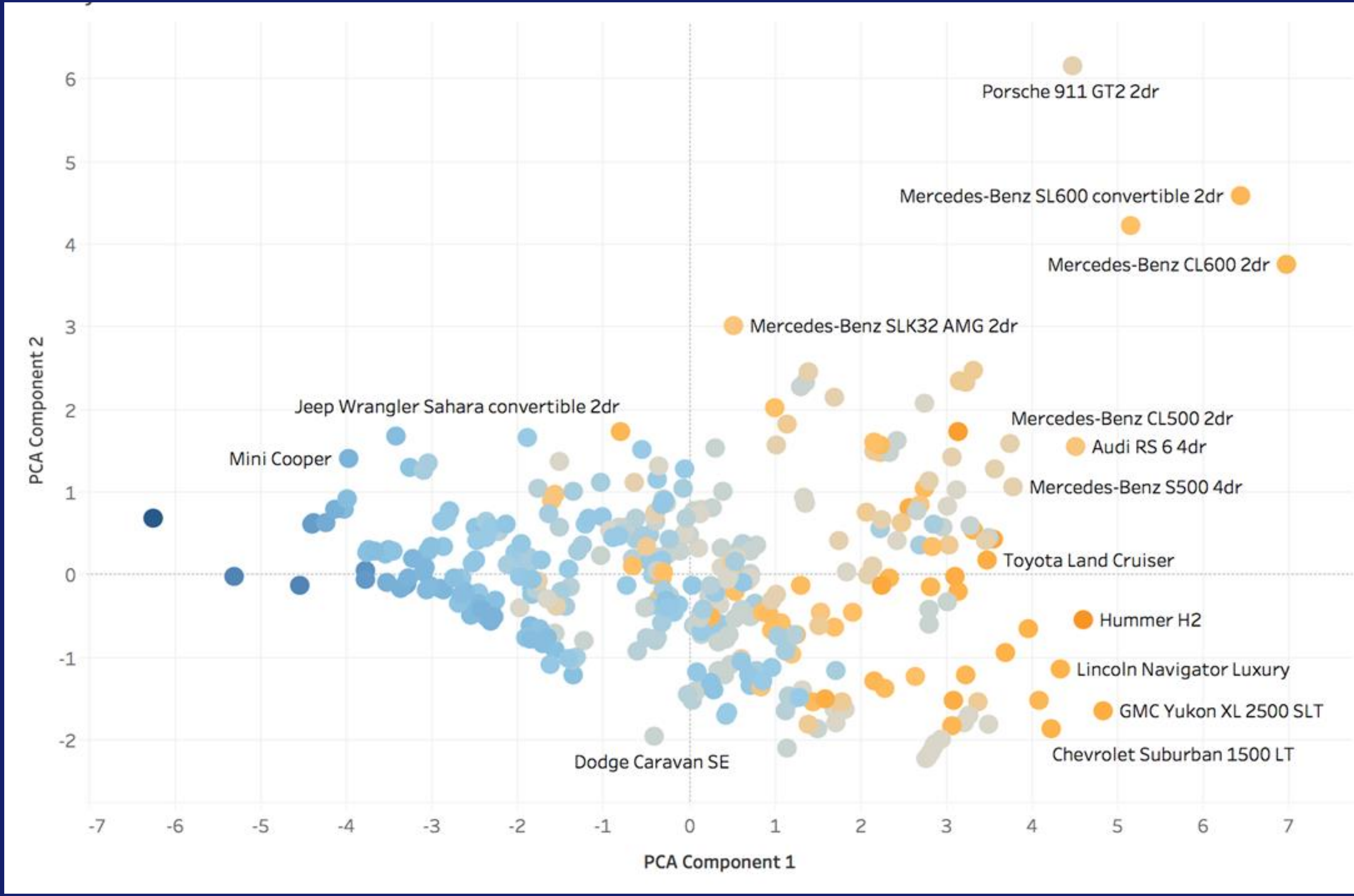
Tech Tip - Setting the Correct Table Calculation

Vehicle	Retail Price	Dealer Cost	Engine Size	Cyl	HP	City MPG	Hwy MPG	Weight	Wheel Base	Len	Width
Acura 3.5 RL 4dr	43,755	39,014	3.50000	6	225	18	24	3,880	115	197	72
Acura 3.5 RL w/Navig...	46,100	41,100	3.50000	6	225	18	24	3,893	115	197	72
Acura MDX	36,945	33,337	3.50000	6	265	17	23	4,451	106	189	77
Acura NSX coupe 2dr ...	89,765	79,978	3.20000	6	290	17	24	3,153	100	174	71
Acura RSX Type S 2dr	23,820	21,761	2.00000	4	200	24	31	2,778	101	172	68
Acura TL 4dr	33,195	30,299	3.20000	6	270	20	28	3,575	108	186	72
Acura TSX 4dr	26,990	24,647	2.40000	4	200	22	29	3,230	105	183	69
Audi A4 1.8T 4dr	25,940	23,508	1.80000	4	170	22	31	3,252	104	179	70
Audi A4 3.0 4dr	31,840	28,846	3.00000	6	220	20	28	3,462	104	179	70
Audi A4 3.0 convertibl...	42,490	38,325	3.00000	6	220	20	27	3,814	105	180	70
Audi A4 3.0 Quattro 4...	34,480	31,388	3.00000	6	220	18	25	3,627	104	179	70
Audi A4 3.0 Quattro 4...	33,430	30,366	3.00000	6	220	17	26	3,583	104	179	70
Audi A4 3.0 Quattro c...	44,240	40,075	3.00000	6	220	18	25	4,013	105	180	70
Audi A41.8T converti...	35,940	32,506	1.80000	4	170	23	30	3,638	105	180	70
Audi A6 2.7 Turbo Qua...	42,840	38,840	2.70000	6	250	18	25	3,836	109	192	71
Audi A6 3.0 4dr	36,640	33,129	3.00000	6	220	20	27	3,561	109	192	71
Audi A6 3.0 Avant Qu...	40,840	37,060	3.00000	6	220	18	25	4,035	109	192	71
Audi A6 3.0 Quattro 4dr	39,640	35,992	3.00000	6	220	18	25	3,880	109	192	71
Audi A6 4.2 Quattro 4dr	49,690	44,936	4.20000	8	300	17	24	4,024	109	193	71
Audi A8 L Quattro 4dr	69,190	64,740	4.20000	8	330	17	24	4,399	121	204	75
Audi RS 6 4dr	84,600	76,417	4.20000	8	450	15	22	4,024	109	191	78
Audi S4 Avant Quattro	49,090	44,446	4.20000	8	340	15	21	3,936	104	179	70
Audi S4 Quattro 4dr	48,040	43,556	4.20000	8	340	14	20	3,825	104	179	70

Tech Tip - Setting the Correct Table Calculation



Tech Tip - Setting the Correct Table Calculation

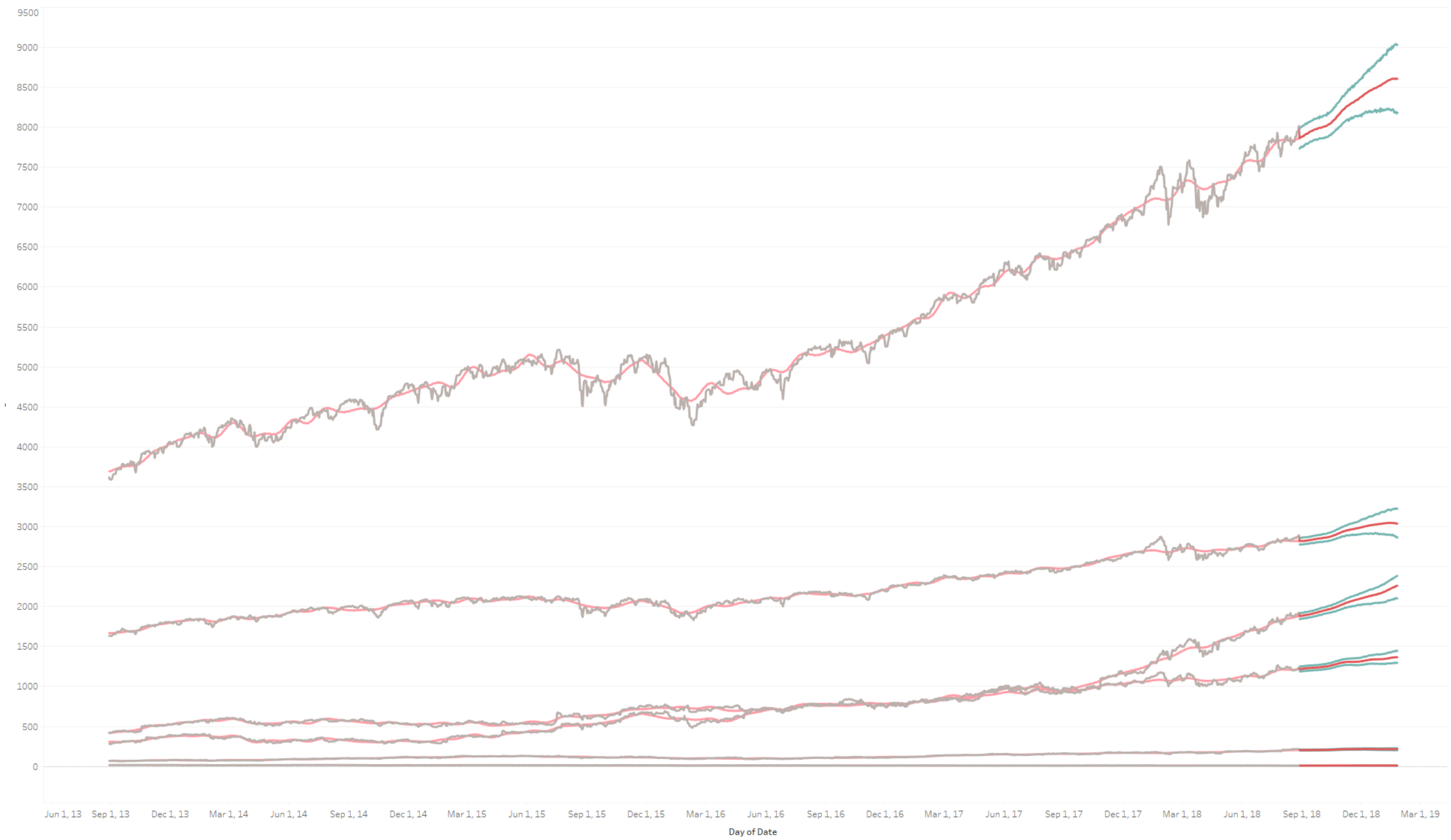


Self-Service Time Series Forecast Application

User Story – Dynamic Forecasting at NetApp

Question:

- Visually exploring forecast results during model evaluation.
- Sharing product utilization forecasts with business managers with current data.



Answer:

- Adapting custom model script for use in Tableau.
- Sharing results in interactive dashboard in Tableau Server.

TABLEAU CONFERENCE

Creating a Self-Service Forecast Application

Converting a Script:

- Understanding how to pass data
- Returning correct results.

Enabling Self-Service:

- Building an interactive forecast dashboard.
- Deploying a Dashboard to Tableau Server for self-service exploration.

Directly From Python

```
import pandas as pd

import numpy as np

from fbprophet import Prophet


df = pd.read_csv('login_history.csv')

periods_to_fcast = 50

m = Prophet()

m.fit(df);


future = m.make_future_dataframe(periods=periods_to_fcast)

forecast = m.predict(future)

m.plot(forecast)
```

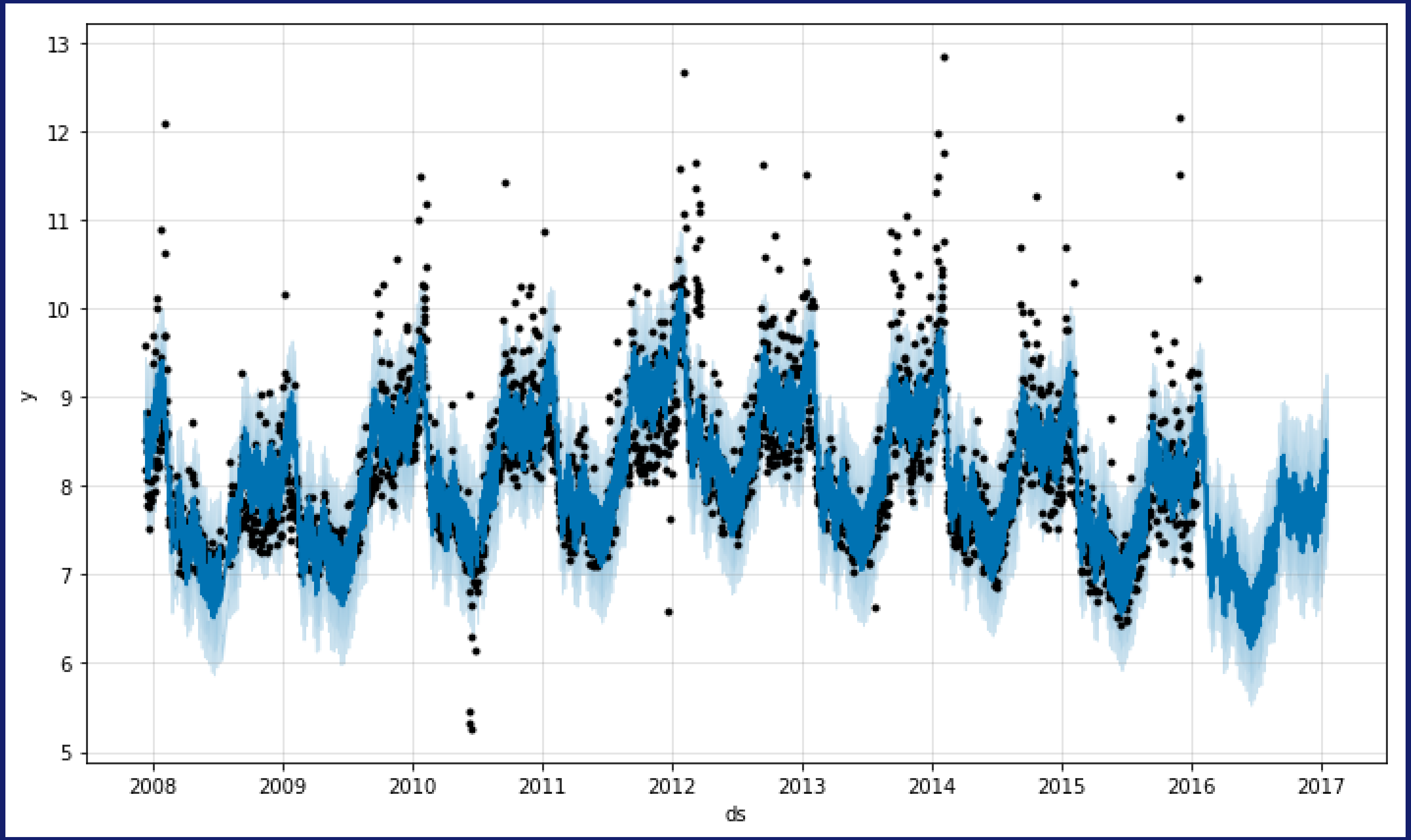


Tableau Calculation

Forecast   Customer Activity

Results are computed along Table (across).

```
SCRIPT_REAL("
import pandas as pd
import numpy as np
from fbprophet import Prophet
period = _arg3[0]+1
df = pd.DataFrame(
    {'ds': _arg1,
     'y': _arg2
    })
print(df.ds)
m = Prophet()
df = df[:-period]
m.fit(df);
future = m.make_future_dataframe(periods=period)
forecast = m.predict(future)
return forecast['yhat'].tolist()
",ATTR([Date]),SUM([Logins]),[Periods to Forecast])
```

Default Table Calculation

The calculation is valid.   3 Dependencies   Apply   OK

Edit Parameter [Periods to Forecast]

Name: Periods to Forecast   Comment >>

Properties

Data type: Integer

Current value: 150

Display format: Automatic

Allowable values: All   List   Range

OK   Cancel

Fully Adapted Code

```
SCRIPT_REAL(" import pandas as pd

import numpy as np

from fbprophet import Prophet


period = _arg3[0]+1

df = pd.DataFrame({'ds': _arg1, 'y': _arg2 })

m = Prophet()

df = df[:-period]

m.fit(df)


future = m.make_future_dataframe(periods=period)

forecast = m.predict(future)

return forecast['yhat'].tolist()

", ATTR([Date]), SUM([Logins]), [Periods to Forecast])
```

TABLEAU CONFERENCE

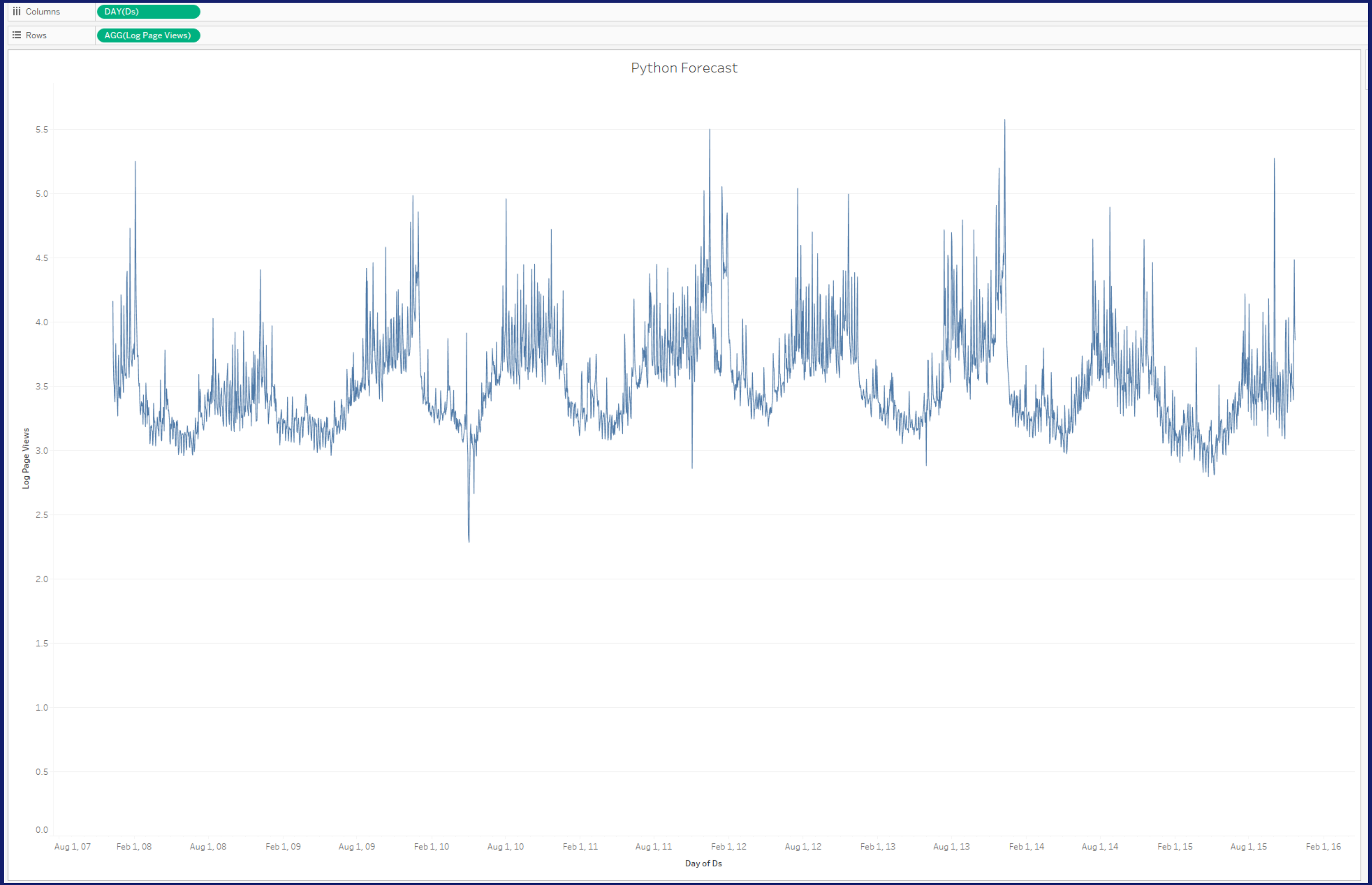
R Forecast Code

```
SCRIPT_REAL(
"library(prophet)
period = .arg3[1]+1
df = data.frame('ds' = .arg1, 'y' = .arg2)
divide = nrow(df)-period
df = df[1:divide,]

m = prophet(df)
future = make_future_dataframe(m, periods=period)
forecast = predict(m, future)

forecast[,'yhat']”,
ATTR([Date]),SUM([Logins]),[Periods to Forecast])
```

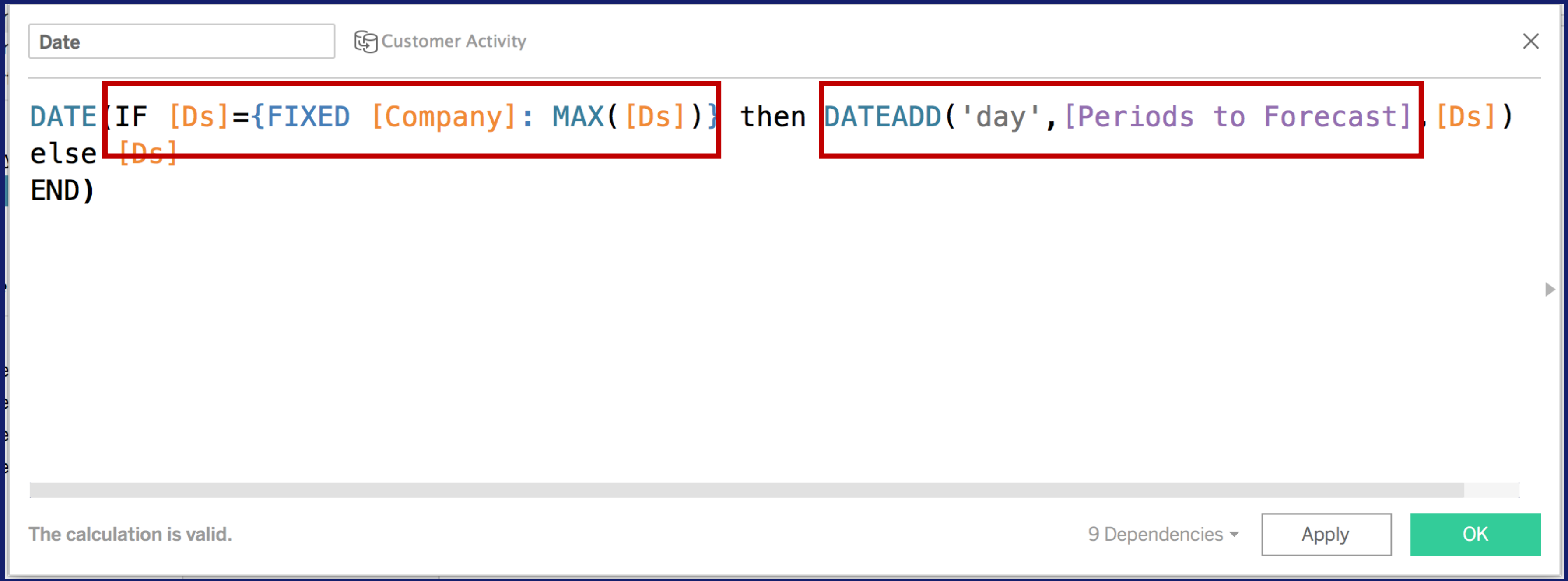
TABLEAU CONFERENCE

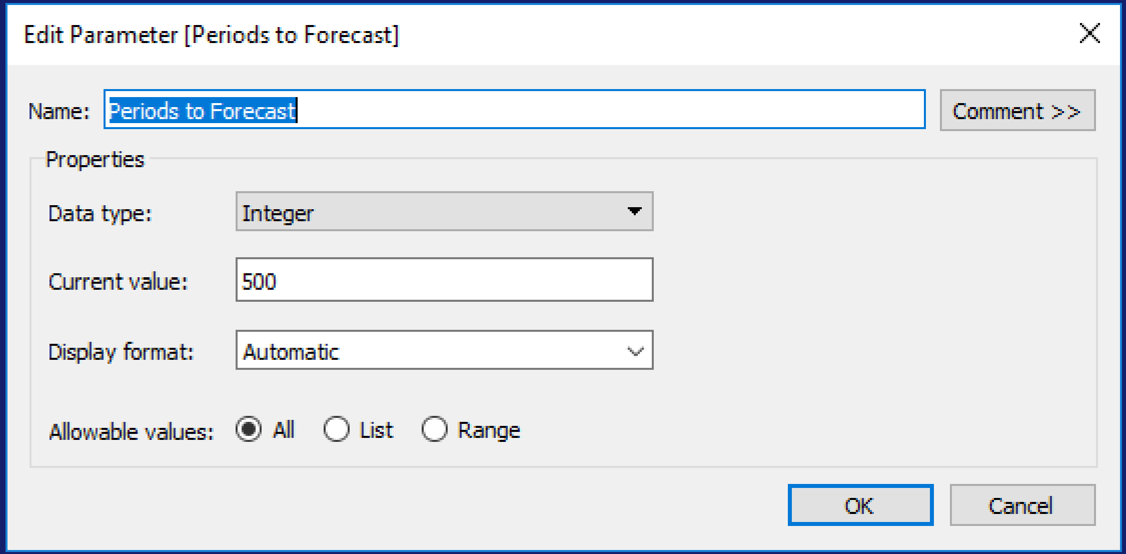
Let’s Take a Look!

Tech Tip - Custom Forecasting in Tableau

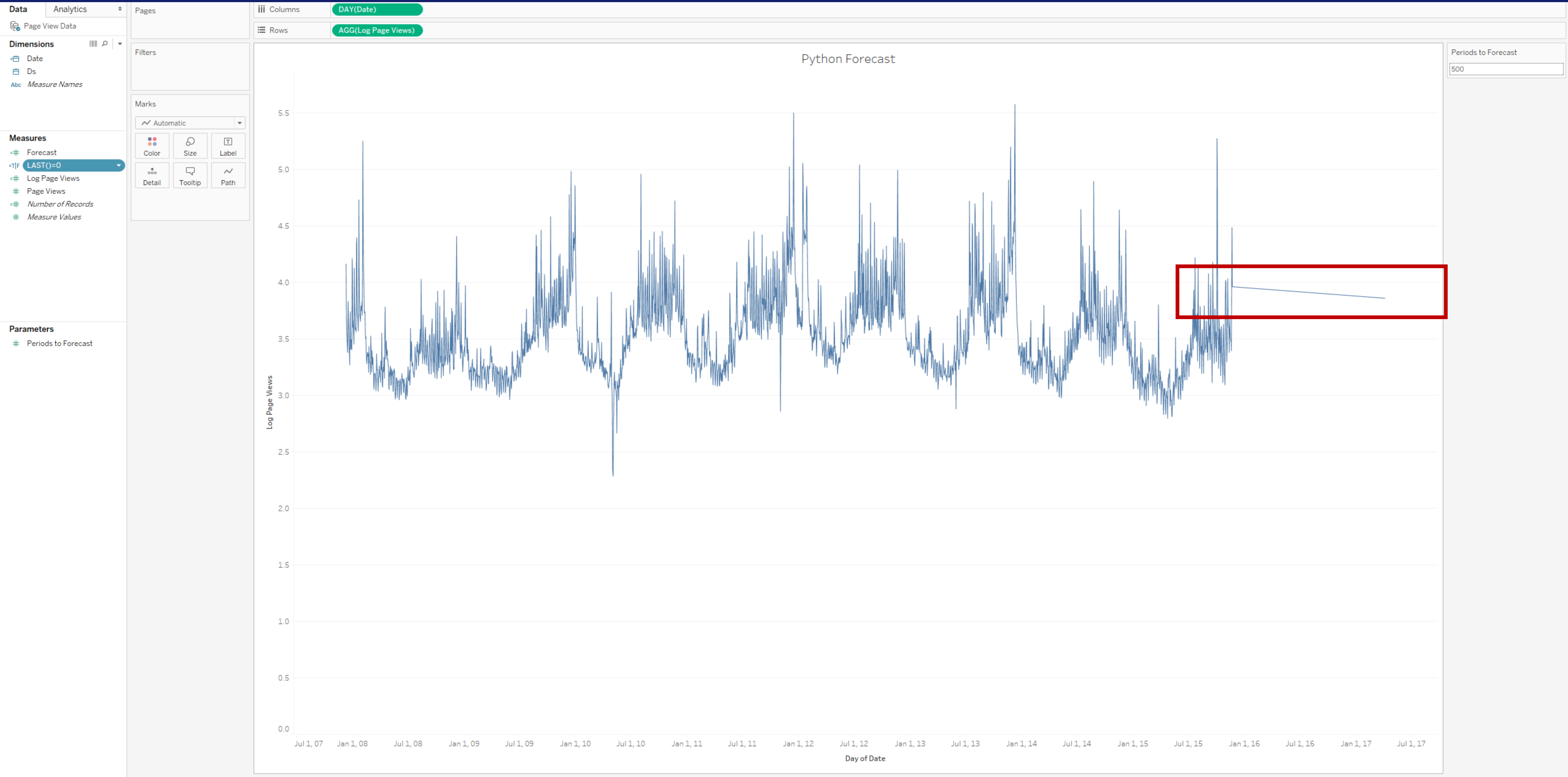


Custom Forecasting in Tableau

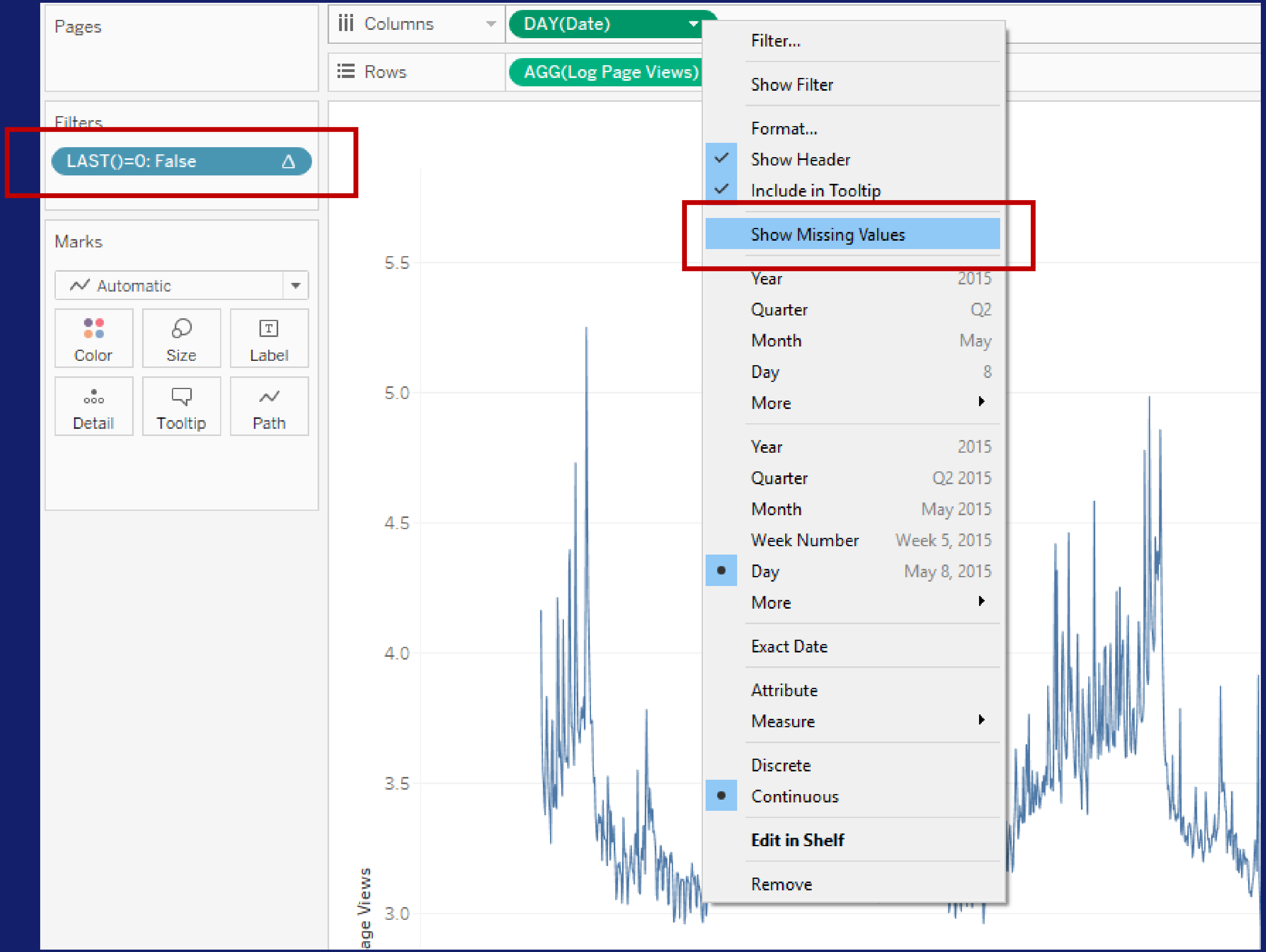




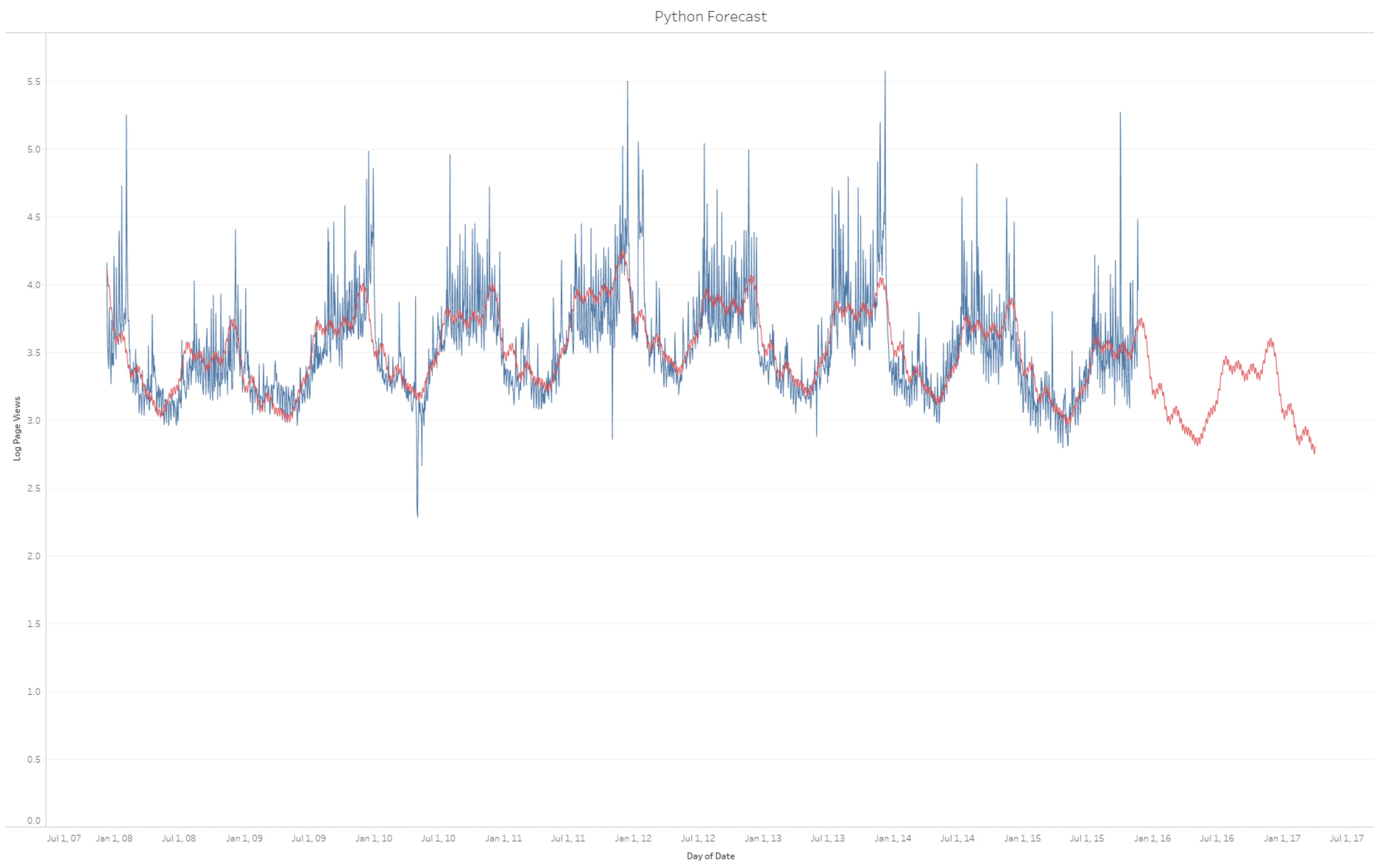
Custom Forecasting in Tableau



Custom Forecasting in Tableau



Custom Forecasting in Tableau



Building and Deploying a Credit Classification Application

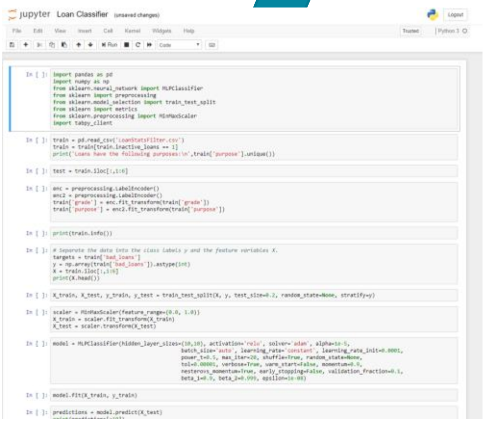
User Story – Self-Service Model Deployment

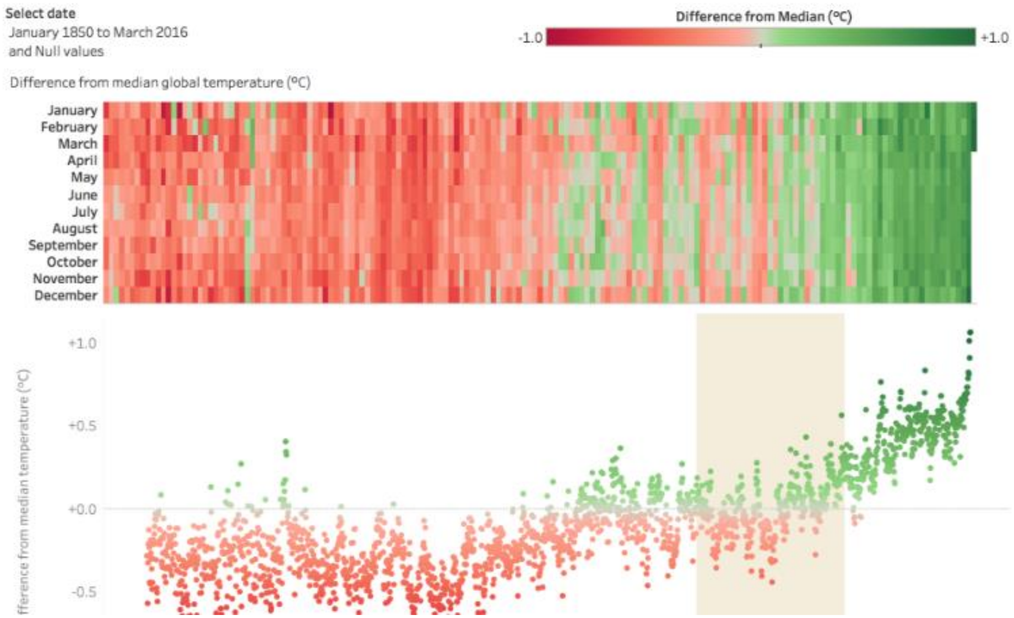
Question:

- Teams have models they want to deploy into production.
- Business users want to explore and iterate on models in real time.

Answer:

- Deploy model in TabPy.
- Make model accessible and interactive in a dashboard application.





Building a Loan Scoring Application

Building a Model:

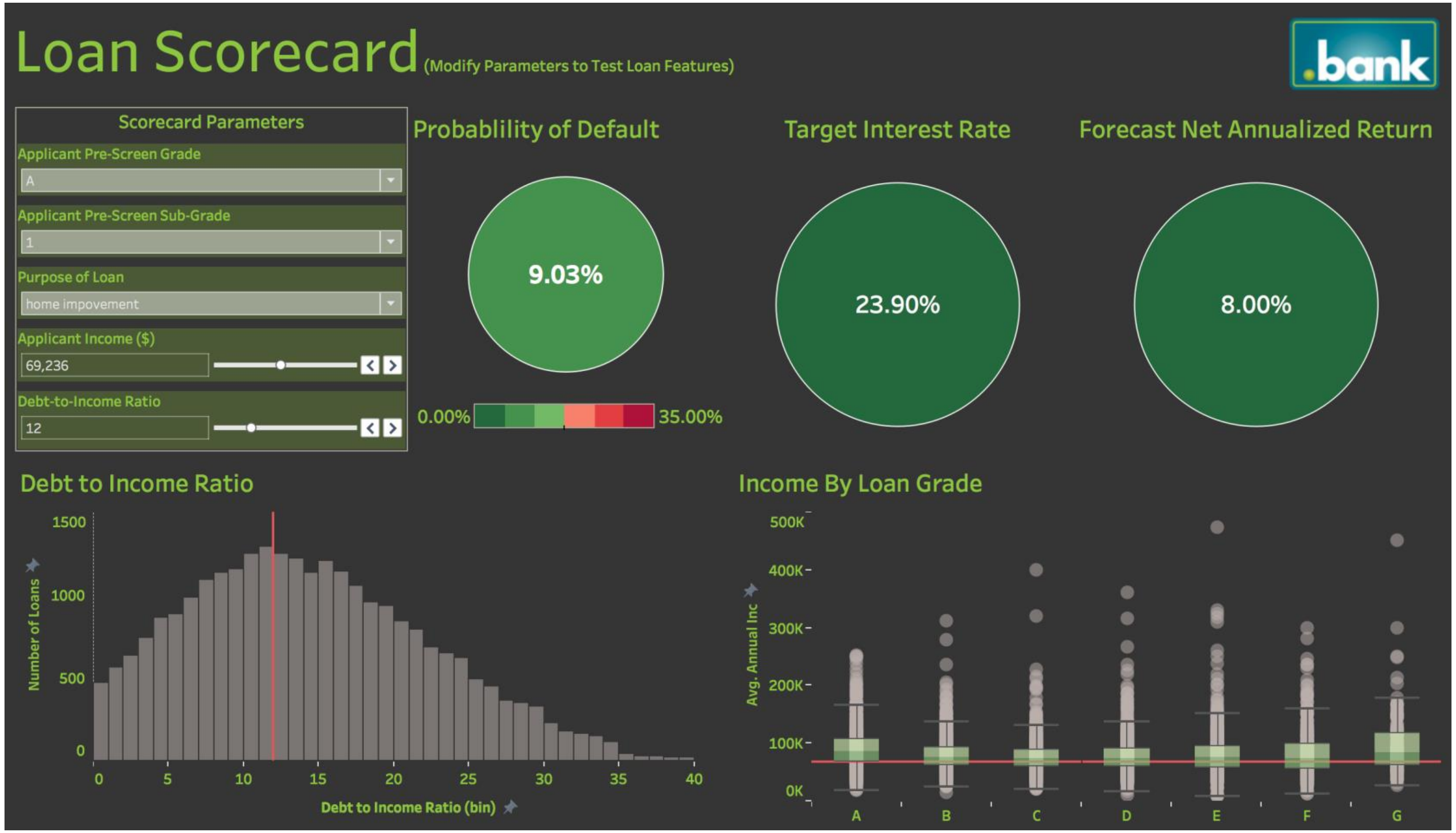
- Training and evaluating
- Adapting for Tableau

Model Simulation:

- Inputting data
- Visualizing results

Deploying at Scale:

- Self-service applications
- Tableau Server



Let’s Take a Look!

Tech Tip – Creating a Model in Jupyter

jupyter Loan Classifier (unsaved changes)

```
In [ ]: import pandas as pd
        import numpy as np
        from sklearn.neural_network import MLPClassifier
        from sklearn import preprocessing
        from sklearn.model_selection import train_test_split
        from sklearn import metrics
        from sklearn.preprocessing import MinMaxScaler
        import tabpy_client

In [ ]: train = pd.read_csv('LoanStatsFilter.csv')
        train = train[train.inactive_loans == 1]
        print('Loans have the following purposes:\n',train['purpose'].unique())

In [ ]: test = train.iloc[:,1:6]

In [ ]: enc = preprocessing.LabelEncoder()
        enc2 = preprocessing.LabelEncoder()
        train['grade'] = enc.fit_transform(train['grade'])
        train['purpose'] = enc2.fit_transform(train['purpose'])

In [ ]: print(train.info())

In [ ]: # Separate the data into the class labels y and the feature variables X.
        targets = train['bad_loans']
        y = np.array(train['bad_loans']).astype(int)
        X = train.iloc[:,1:6]
        print(X.head())

In [ ]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=None, stratify=y)

In [ ]: scaler = MinMaxScaler(feature_range=(0.0, 1.0))
        X_train = scaler.fit_transform(X_train)
        X_test = scaler.transform(X_test)

In [ ]: model = MLPClassifier(hidden_layer_sizes=(10,10), activation='relu', solver='adam', alpha=1e-5,
                                          batch_size='auto', learning_rate='constant', learning_rate_init=0.0001,
                                          power_t=0.5, max_iter=20, shuffle=True, random_state=None,
                                          tol=0.00001, verbose=True, warm_start=False, momentum=0.9,
                                          nesterovs_momentum=True, early_stopping=False, validation_fraction=0.1,
                                          beta_1=0.9, beta_2=0.999, epsilon=1e-08)

In [ ]: model.fit(X_train, y_train)

In [ ]: predictions = model.predict(X_test)
```

Tech Tip – Deploying a Function in TabPy

TABLEAU CONFERENCE

```
In [ ]: metrics.confusion_matrix(y_test, threshold_preds)
```

```
In [ ]: def loanclassifierfull(_arg1, _arg2, _arg3, _arg4, _arg5):
            from pandas import DataFrame

            # Load data from tableau (brought in as lists) into a dictionary
            # Like I mentioned in my email, the columns get sorted alphabetically in this constructor
            # Adding the numbers sorts them correctly
            d = {'1-grade': _arg1, '2-income': _arg2, '3-sub_grade_num': _arg3, '4-purpose': _arg4, '5-dti': _arg5}
            # Convert the dictionary to a Pandas Dataframe
            df = DataFrame(data=d)

            # Transform categorical variables into numerical/continuous features
            df['1-grade'] = enc.transform(df['1-grade'])
            df['4-purpose'] = enc2.transform(df['4-purpose'])
            print(df.head())

            # This is the missing step from my first version
            # We need to scale the inputs to the Model or it will be totally off
            # Hope no one saw this
            # The scaler, since it's saved in the code, should be pickled automatically by TabPy and available for reuse
            # This should also be the case for the feature encoder above
            df = scaler.transform(df)

            # Use the loaded model to develop predictions for the new data from Tableau
            probs = model.predict_proba(df)
            return [loan[1] for loan in probs]
```

```
In [ ]: func_probs =loanclassifierfull(test.iloc[:,0],test.iloc[:,1],test.iloc[:,2],test.iloc[:,3],test.iloc[:,4])
        print('Calc Results Come After This')
        print(func_probs[:10])
```

```
In [ ]: client = tabpy_client.Client('http://localhost:9004')
```

```
In [ ]: client.deploy('loanclassifierfull', loanclassifierfull,
                      'Returns the probability that a loan will result in a bad loan based on its Grade, Income, '
                      'SubGradeNum, Purpose, and DTI', override=True)
```

Let's Take a Look!

Tech Tip – Model Simulation

TABLEAU CONFERENCE

Deployed Test Loan Scenario | Loan Information

Results are computed along Table (across).

```
SCRIPT_REAL("return tabpy.query('loanclassifierfull',_arg1,_arg2,_arg3,_arg4,_arg5)['response']",
([Test Grade]),
([Test Income]),
([Test Sub Grade Num]),
([Test Purpose]),
([Test DTI]))
```

Default Table Calculation

The calculation is valid. 6 Dependencies Apply OK

Edit Parameter [Test Purpose]

Name: Test Purpose | Comment >>

Properties

Data type: String

Current value: home impovement

Display format:

Allowable values: All / List / Range

List of values

Value	Display As
car	car
credit_card	credit card
small_business	small business
other	other
wedding	wedding
debt_consolidation	debt consolidation
home_improvement	home improvement
major_purchase	major purchase

Add from Parameter | Add from Field | Paste from Clipboard | Clear All

OK | Cancel

Edit Parameter [Test DTI]

Name: Test DTI | Comment >>

Properties

Data type: Float

Current value: 7

Display format: Automatic

Allowable values: All / List / Range

Range of values

Minimum: 0

Maximum: 50

Step size: 1

Set from Parameter | Set from Field

OK | Cancel

Edit Parameter [Test Sub Grade Num]

Name: Test Sub Grade Num | Comment >>

Properties

Data type: Float

Current value: 1

Display format: Automatic

Allowable values: All / List / Range

List of values

Value	Display As
0.2	0.2
0.4	0.4
0.6	0.6
0.8	0.8
1	1
Add	

Add from Parameter | Add from Field | Paste from Clipboard | Clear All

OK | Cancel

Conclusion

Data Science:

- Framing business questions
- Building a model
- Adapting code and operationalizing using Tableau

Business Use Cases:

- Exploring complex problems visually
- Scaling with Tableau Server

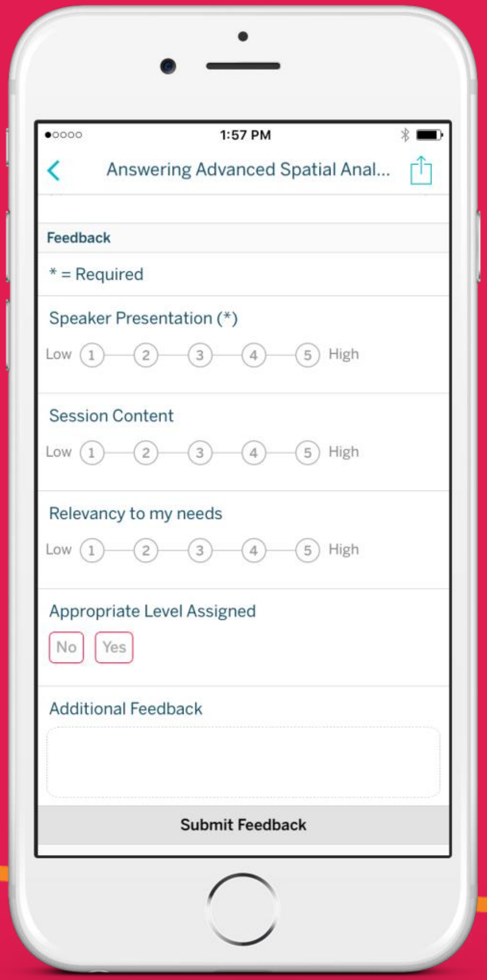
Tableau in Data Science:

- Exploratory data analysis
- Operationalization

Questions?

nmannheimer@tableau.com

TABLEAU CONFERENCE



Please complete the session survey from the Session Details screen in your TC18 app

TABLEAU CONFERENCE

RELATED SESSIONS

Advanced analytics at scale | Deploying machine learning in the enterprise

Today | 12:30 – 1:30 | MCCNO - L3 - 346

Embedding Tableau for self-service data science

Today | 2:15 – 3:15 | MCCNO - L2 - R02

Thank you!

Contact me at nmannheimer@tableau.com

TABLEAU
CONFERENCE