TABLEAU CONFERENCE
Welcome
Building Data Science Applications with TabPy and R

Nathan Mannheimer
Product Manager
Advanced Analytics
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?

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
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
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

• 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
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()
SCRIPT_REAL('import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
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[:,8])','
SUM([City MPG]),
SUM([Cyl]),
SUM([Dealer Cost]),
SUM([Engine Size]),
SUM([HP]),
SUM([Len]),
SUM([Width]),
[Selected PCA Component 1])

The calculation is valid.
2 Dependencies → Apply → OK
Fully Adapted Code
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]])
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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Acura 3.5 RL 4dr</td>
<td>43,755</td>
<td>39,014</td>
<td>3.50000</td>
<td>6</td>
<td>225</td>
<td>19</td>
<td>24</td>
<td>3.880</td>
<td>115</td>
</tr>
<tr>
<td>Acura 3.5 RL w/Navig..</td>
<td>46,100</td>
<td>41,100</td>
<td>3.50000</td>
<td>6</td>
<td>225</td>
<td>19</td>
<td>24</td>
<td>3.893</td>
<td>115</td>
</tr>
<tr>
<td>Acura MDX</td>
<td>36,945</td>
<td>33,337</td>
<td>3.50000</td>
<td>6</td>
<td>265</td>
<td>17</td>
<td>23</td>
<td>4.451</td>
<td>106</td>
</tr>
<tr>
<td>Acura NSX coupe 2dr..</td>
<td>89,765</td>
<td>79,978</td>
<td>3.20000</td>
<td>6</td>
<td>290</td>
<td>17</td>
<td>24</td>
<td>3.153</td>
<td>100</td>
</tr>
<tr>
<td>Acura RSX Type S 2dr..</td>
<td>28,820</td>
<td>21,781</td>
<td>2.00000</td>
<td>4</td>
<td>200</td>
<td>31</td>
<td></td>
<td>2.778</td>
<td>101</td>
</tr>
<tr>
<td>Acura TL 4dr</td>
<td>32,195</td>
<td>30,299</td>
<td>3.20000</td>
<td>6</td>
<td>270</td>
<td>20</td>
<td>28</td>
<td>3.575</td>
<td>108</td>
</tr>
<tr>
<td>Acura TSX 4dr</td>
<td>26,990</td>
<td>24,647</td>
<td>2.40000</td>
<td>4</td>
<td>200</td>
<td>22</td>
<td>29</td>
<td>3.230</td>
<td>105</td>
</tr>
<tr>
<td>Audi A4 1.8T 4dr</td>
<td>25,940</td>
<td>23,508</td>
<td>1.80000</td>
<td>4</td>
<td>170</td>
<td>22</td>
<td>31</td>
<td>3.252</td>
<td>104</td>
</tr>
<tr>
<td>Audi A4 3.0 4dr</td>
<td>31,840</td>
<td>28,846</td>
<td>3.00000</td>
<td>6</td>
<td>220</td>
<td>20</td>
<td>28</td>
<td>3.462</td>
<td>104</td>
</tr>
<tr>
<td>Audi A4 3.0 convertib..</td>
<td>42,490</td>
<td>38,325</td>
<td>3.00000</td>
<td>6</td>
<td>220</td>
<td>20</td>
<td>27</td>
<td>3.014</td>
<td>105</td>
</tr>
<tr>
<td>Audi A4 3.0 Quattro 4..</td>
<td>34,480</td>
<td>31,388</td>
<td>3.00000</td>
<td>6</td>
<td>220</td>
<td>18</td>
<td>25</td>
<td>3.627</td>
<td>104</td>
</tr>
<tr>
<td>Audi A4 2.0 Quattro 4..</td>
<td>33,420</td>
<td>30,266</td>
<td>3.00000</td>
<td>6</td>
<td>220</td>
<td>17</td>
<td>25</td>
<td>3.583</td>
<td>104</td>
</tr>
<tr>
<td>Audi A4 3.0 Quattro c..</td>
<td>44,240</td>
<td>40,075</td>
<td>3.00000</td>
<td>6</td>
<td>220</td>
<td>18</td>
<td>25</td>
<td>4.013</td>
<td>105</td>
</tr>
<tr>
<td>Audi A4 1.8T converti..</td>
<td>35,940</td>
<td>32,505</td>
<td>1.80000</td>
<td>4</td>
<td>170</td>
<td>23</td>
<td>30</td>
<td>3.638</td>
<td>105</td>
</tr>
<tr>
<td>Audi A4 2.7 Turbo Qu..</td>
<td>42,840</td>
<td>39,840</td>
<td>2.70000</td>
<td>6</td>
<td>250</td>
<td>18</td>
<td>25</td>
<td>3.836</td>
<td>109</td>
</tr>
<tr>
<td>Audi A6 2.0 4dr</td>
<td>36,640</td>
<td>33,129</td>
<td>3.00000</td>
<td>6</td>
<td>220</td>
<td>20</td>
<td>27</td>
<td>3.561</td>
<td>109</td>
</tr>
<tr>
<td>Audi A6 3.0 Avant Qu..</td>
<td>40,840</td>
<td>37,060</td>
<td>3.00000</td>
<td>6</td>
<td>220</td>
<td>18</td>
<td>25</td>
<td>4.055</td>
<td>109</td>
</tr>
<tr>
<td>Audi A6 3.0 Quattro 4dr</td>
<td>39,640</td>
<td>35,992</td>
<td>3.00000</td>
<td>6</td>
<td>220</td>
<td>18</td>
<td>25</td>
<td>3.880</td>
<td>109</td>
</tr>
<tr>
<td>Audi A6 4.2 Quattro 4dr</td>
<td>40,690</td>
<td>44,936</td>
<td>4.20000</td>
<td>8</td>
<td>300</td>
<td>17</td>
<td>24</td>
<td>4.024</td>
<td>109</td>
</tr>
<tr>
<td>Audi A6 L Quattro 4dr</td>
<td>69,180</td>
<td>64,740</td>
<td>4.20000</td>
<td>8</td>
<td>330</td>
<td>17</td>
<td>24</td>
<td>4.399</td>
<td>121</td>
</tr>
<tr>
<td>Audi RS 6.4dr</td>
<td>84,600</td>
<td>76,417</td>
<td>4.20000</td>
<td>8</td>
<td>450</td>
<td>15</td>
<td>22</td>
<td>4.024</td>
<td>109</td>
</tr>
<tr>
<td>Audi S4 Avant Qu..</td>
<td>49,090</td>
<td>44,446</td>
<td>4.20000</td>
<td>8</td>
<td>340</td>
<td>15</td>
<td>21</td>
<td>3.936</td>
<td>104</td>
</tr>
<tr>
<td>Audi S4 Quattro 4dr</td>
<td>48,040</td>
<td>43,556</td>
<td>4.20000</td>
<td>8</td>
<td>340</td>
<td>14</td>
<td>20</td>
<td>3.825</td>
<td>104</td>
</tr>
</tbody>
</table>
Tech Tip - Setting the Correct Table Calculation

In Tableau, when setting up a calculation for PCA (Principal Component Analysis), it's important to choose the correct Compute Using option. For instance, selecting 'Specific Dimensions' with 'Vehicle' as the dimension ensures that the calculation is performed across specific vehicle categories, rather than across all data points. This is crucial for accurate data representation and analysis.
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.
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

```python
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])"
```
Fully Adapted Code
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()
R Forecast Code
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])
Let’s Take a Look!
Tech Tip - Custom Forecasting in Tableau

![Python Forecast Chart](chart.png)
Custom Forecasting in Tableau

```plaintext
DATEIF ([Ds]={[FIXED [Company]: MAX([Ds])}
then
DATEADD('day',[Periods to Forecast],[Ds])
else
[Ds]
END)
```

The calculation is valid.

Edit Parameter [Periods to Forecast]

- **Name:** Periods to Forecast
- **Data type:** Integer
- **Current value:** 500
- **Display format:** Automatic
- **Allowable values:** All

OK
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
Tech Tip – Deploying a Function in TabPy

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(dict(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

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

The calculation is valid.

---

**Edit Parameter [Test Purpose]**

- **Name**: Test Purpose
- **Data type**: String
- **Current value**: home improvement
- **Display format**: None
- **Allowable values**: All
- **List**: (library, credit_card, college, student, vehicle, small_business, other, wedding, debt consolidation, home improvement, home improvement, major_purchase, major_purchase)

**Edit Parameter [Test DTI]**

- **Name**: Test DTI
- **Data type**: Float
- **Current value**: 0
- **Display format**: Automatic
- **Allowable values**: All
- **List**: (0.2, 0.4, 0.6, 0.8, 1.0)

**Edit Parameter [Test Sub Grade Num]**

- **Name**: Test Sub Grade Num
- **Data type**: Float
- **Current value**: 1
- **Display format**: Automatic
- **Allowable values**: All
- **List**: (0.2, 0.4, 0.6, 0.8, 1.0)
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
Please complete the session survey from the Session Details screen in your TC18 app.
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