

Level 3 – Advanced Analytics

Nathan Mannheimer

Product Manager

DATA
DAY OUT



Who Am I?

- Product Manager for Advanced Analytics
- Lecturer in Data Science and Human Centered Engineering at University of Washington
- 私は茨城県に住んでいました!



Advanced Analysis in Tableau

Drag and Drop Statistics

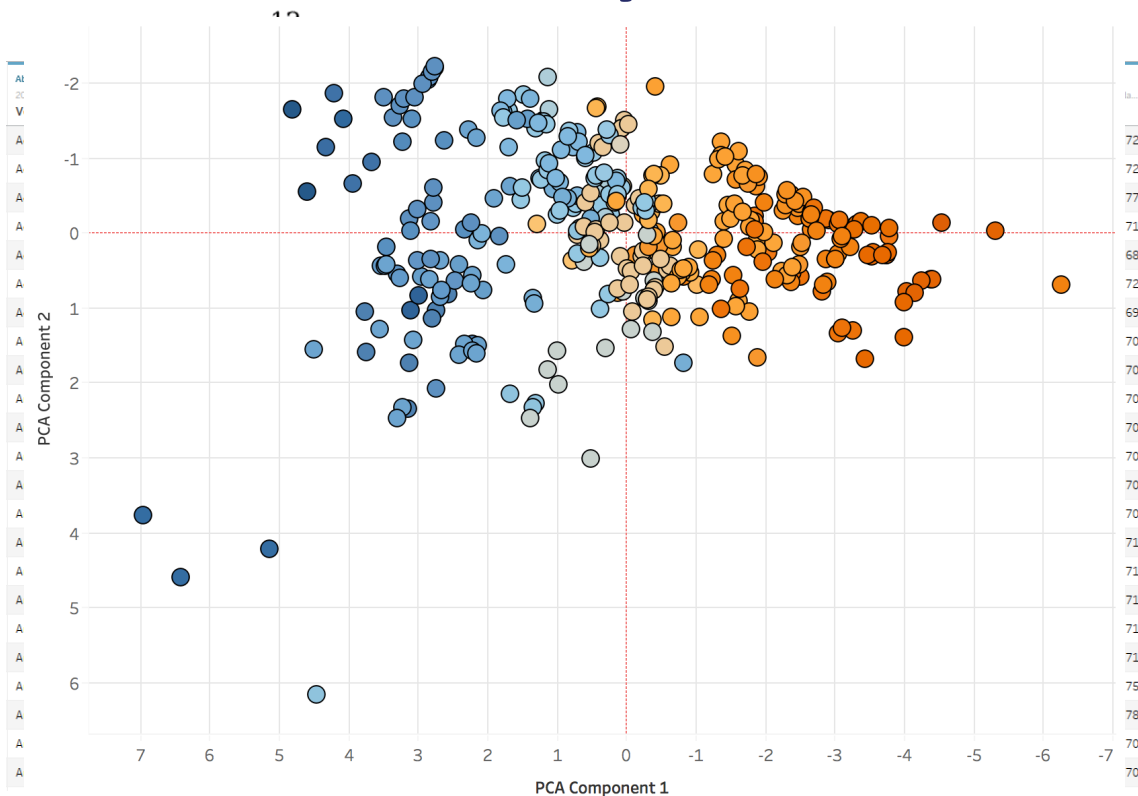
Advanced Calculations

Extensions for Data Science

Extensible Analytics R and Python

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.



User Story – Dynamic Forecasting at

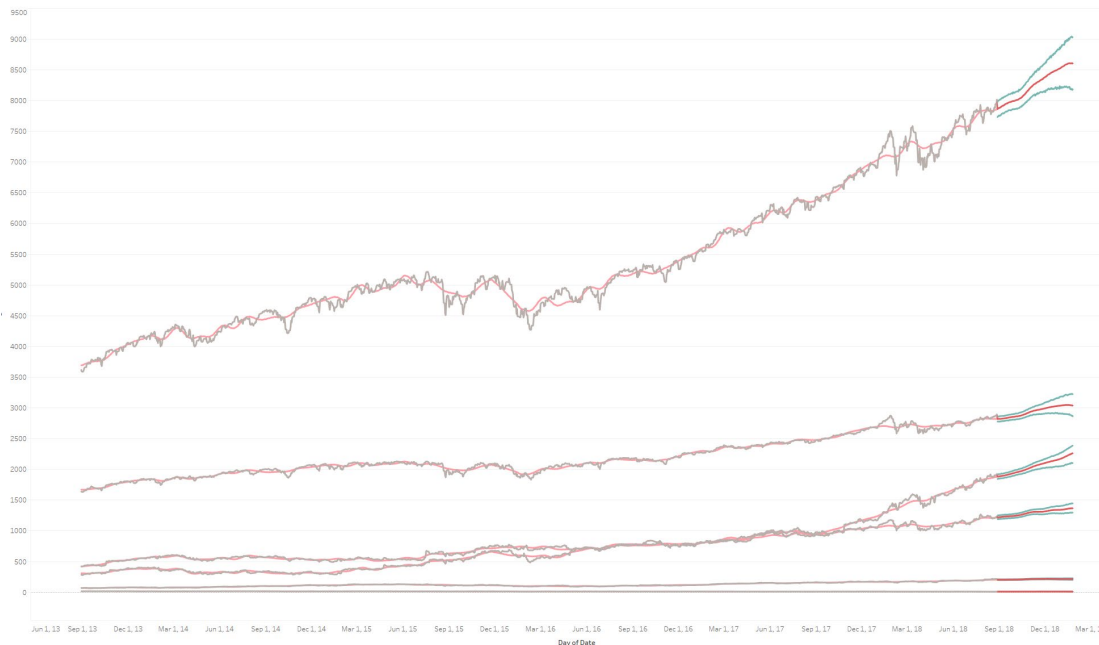


Question:

- 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.



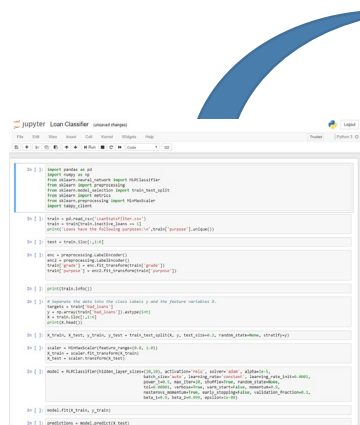
User Story – Self-Service Model Deployment

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.



```
%%writefile loan_classifier.py
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

def train_model(X_train, y_train):
    model = LogisticRegression()
    model.fit(X_train, y_train)
    return model

def predict(model, X_test):
    y_pred = model.predict(X_test)
    return y_pred

# Load data
data = pd.read_csv('loan_data.csv')

# Split data
X_train, X_test, y_train, y_test = train_test_split(data[['L1', 'L2', 'L3', 'L4', 'L5', 'L6', 'L7', 'L8', 'L9', 'L10', 'L11', 'L12', 'L13', 'L14', 'L15', 'L16', 'L17', 'L18', 'L19', 'L20', 'L21', 'L22', 'L23', 'L24', 'L25', 'L26', 'L27', 'L28', 'L29', 'L30', 'L31', 'L32', 'L33', 'L34', 'L35', 'L36', 'L37', 'L38', 'L39', 'L40', 'L41', 'L42', 'L43', 'L44', 'L45', 'L46', 'L47', 'L48', 'L49', 'L50', 'L51', 'L52', 'L53', 'L54', 'L55', 'L56', 'L57', 'L58', 'L59', 'L60', 'L61', 'L62', 'L63', 'L64', 'L65', 'L66', 'L67', 'L68', 'L69', 'L70', 'L71', 'L72', 'L73', 'L74', 'L75', 'L76', 'L77', 'L78', 'L79', 'L80', 'L81', 'L82', 'L83', 'L84', 'L85', 'L86', 'L87', 'L88', 'L89', 'L90', 'L91', 'L92', 'L93', 'L94', 'L95', 'L96', 'L97', 'L98', 'L99', 'L100']], data[['L101', 'L102', 'L103', 'L104', 'L105', 'L106', 'L107', 'L108', 'L109', 'L110', 'L111', 'L112', 'L113', 'L114', 'L115', 'L116', 'L117', 'L118', 'L119', 'L120', 'L121', 'L122', 'L123', 'L124', 'L125', 'L126', 'L127', 'L128', 'L129', 'L130', 'L131', 'L132', 'L133', 'L134', 'L135', 'L136', 'L137', 'L138', 'L139', 'L140', 'L141', 'L142', 'L143', 'L144', 'L145', 'L146', 'L147', 'L148', 'L149', 'L150', 'L151', 'L152', 'L153', 'L154', 'L155', 'L156', 'L157', 'L158', 'L159', 'L160', 'L161', 'L162', 'L163', 'L164', 'L165', 'L166', 'L167', 'L168', 'L169', 'L170', 'L171', 'L172', 'L173', 'L174', 'L175', 'L176', 'L177', 'L178', 'L179', 'L180', 'L181', 'L182', 'L183', 'L184', 'L185', 'L186', 'L187', 'L188', 'L189', 'L190', 'L191', 'L192', 'L193', 'L194', 'L195', 'L196', 'L197', 'L198', 'L199', 'L200']], test_size=0.2, random_state=42)

# Train model
model = train_model(X_train, y_train)

# Predict on test set
y_pred = predict(model, X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy}')
```

Select date
January 1850 to March 2016
and Null values

Difference from median global temperature (°C)

