R, Python, and Tableau

A Love Triangle

James Blair
Product Manager, Posit PBC

Photo by Arindam Mahanta on Unsplash

Photo by Geert Pieters on Unsplash

Photo by Samuel Sianipar on Unsplash

Photo by Geert Pieters on Unsplash

Photo by Samuel Sianipar on Unsplash

Photo by Geert Pieters on Unsplash

Photo by Samuel Sianipar on Unsplash

Tableau - Music Dashboard

Entire View

Show Me

Pages

Columns: Attribute One, X axis

Rows: Attribute Two, Y axis

Data

Analytics

Filters

Music Genre

Marks

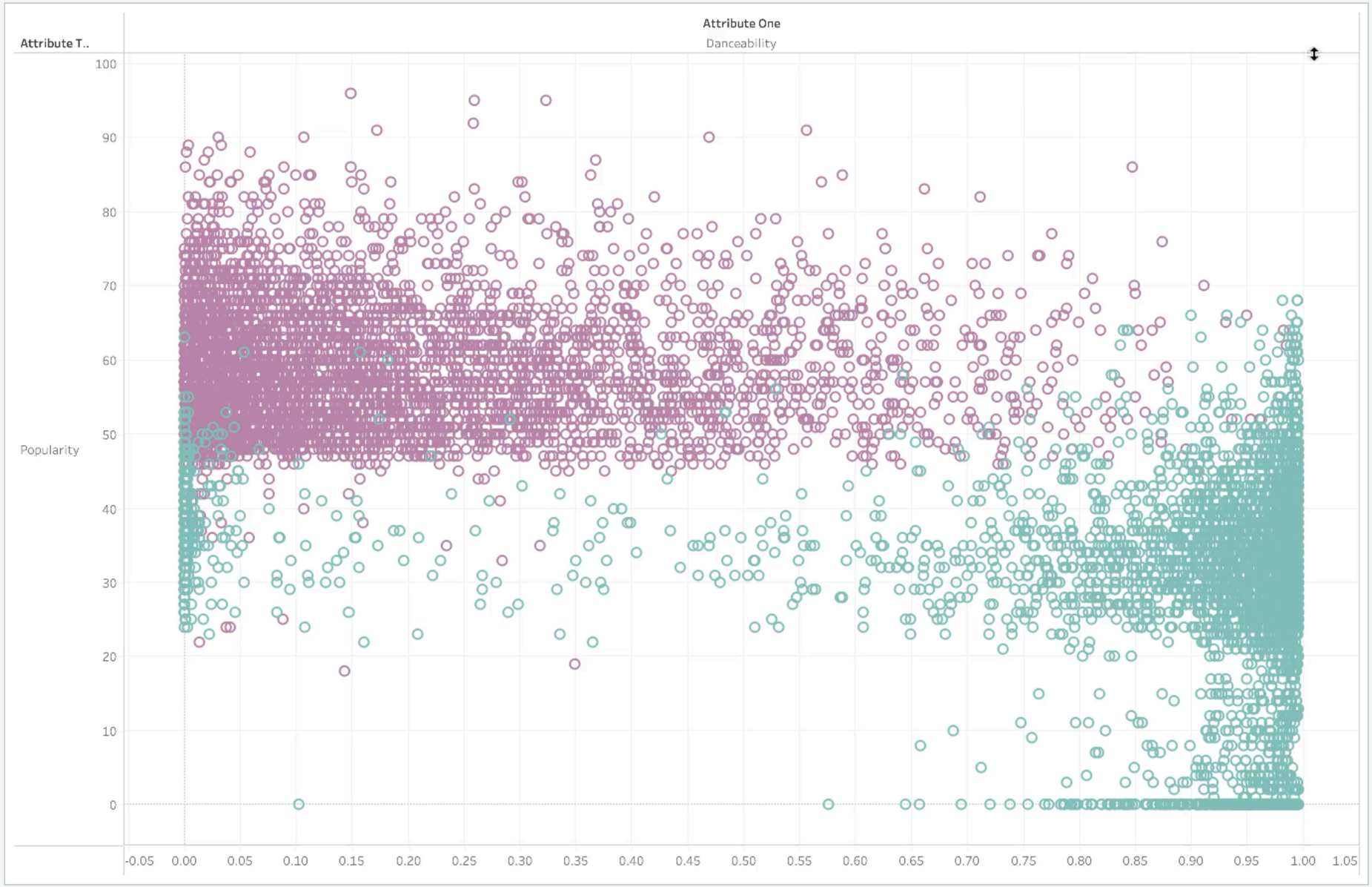
Automatic

Color | Size | Label

Detail | Tooltip | Shape

Music Genre

Track Name



Music Genre

Classical

Hip-Hop

Music Genre

(All)

Alternative

Anime

Blues

Classical

Country

Electronic

Hip-Hop

Jazz

Rap

Rock

Highlight Music Genre

Highlight Music Genre

Attribute One

Danceability

Attribute Two

Popularity

Data Source | Scatter | R RF | Python RF | Music Explorer

10000 marks | 1 row by 1 column | SUM(X axis): 5,119.6657

James Blair

A random Forest

A random Forest

```r
library(ranger)

rf_fit <- function(x, y) {
  train_data <- data.frame(x = x, y = y)
  fit <- ranger(y ~ x, data = train_data)
  predictions(fit)
}
```

plumber

```r
library(plumber)
library(ranger)

#* @apiTitle Tableau Random Forest
#* @apiDescription Fit simple random forest model to input data

#* Fit random forest to input data
#* @post /rf
rf_fit <- function(x, y) {
  train_data <- data.frame(x = x, y = y)
  fit <- ranger(y ~ x, data = train_data)
  predictions(fit)
}
```

plumbertableau

```r
library(plumber)
library(plumbertableau)
library(ranger)

#* @apiTitle Tableau Random Forest
#* @apiDescription Fit simple random forest model to input data

#* Fit random forest to input data
#* @tableauArg x:numeric independent variable
#* @tableauArg y:any dependent variable
#* @tableauReturn numeric Predicted output from fit model
#* @post /rf
rf_fit <- function(x, y) {
  train_data <- data.frame(x = x, y = y)
  fit <- ranger(y ~ x, data = train_data)
  predictions(fit)
}

#* @plumber
tableau_extension
```

A random Forest

```python
from typing import List

from sklearn.ensemble import RandomForestRegressor
import numpy as np

def rf(x: List[float], y: List[float]) -> List[float]:
    X = np.array(x).reshape(-1, 1)
    regr = RandomForestRegressor()
    regr.fit(X, y)
    preds = regr.predict(X)
    return preds.tolist()
```

A random Forest

```python
from typing import List

from sklearn.ensemble import RandomForestRegressor
import numpy as np

def rf(x: List[float], y: List[float]) -> List[float]:
    X = np.array(x).reshape(-1, 1)
    regr = RandomForestRegressor()
    regr.fit(X, y)
    preds = regr.predict(X)
    return preds.tolist()
```

fastapi

```python
from typing import List

from fastapi import FastAPI
from sklearn.ensemble import RandomForestRegressor
import numpy as np

app = FastAPI()

@app.post("/rf")
def rf(x: List[float], y: List[float]) -> List[float]:
    X = np.array(x).reshape(-1, 1)
    regr = RandomForestRegressor()
    regr.fit(X, y)
    preds = regr.predict(X)
    return preds.tolist()
```

fastapitableau

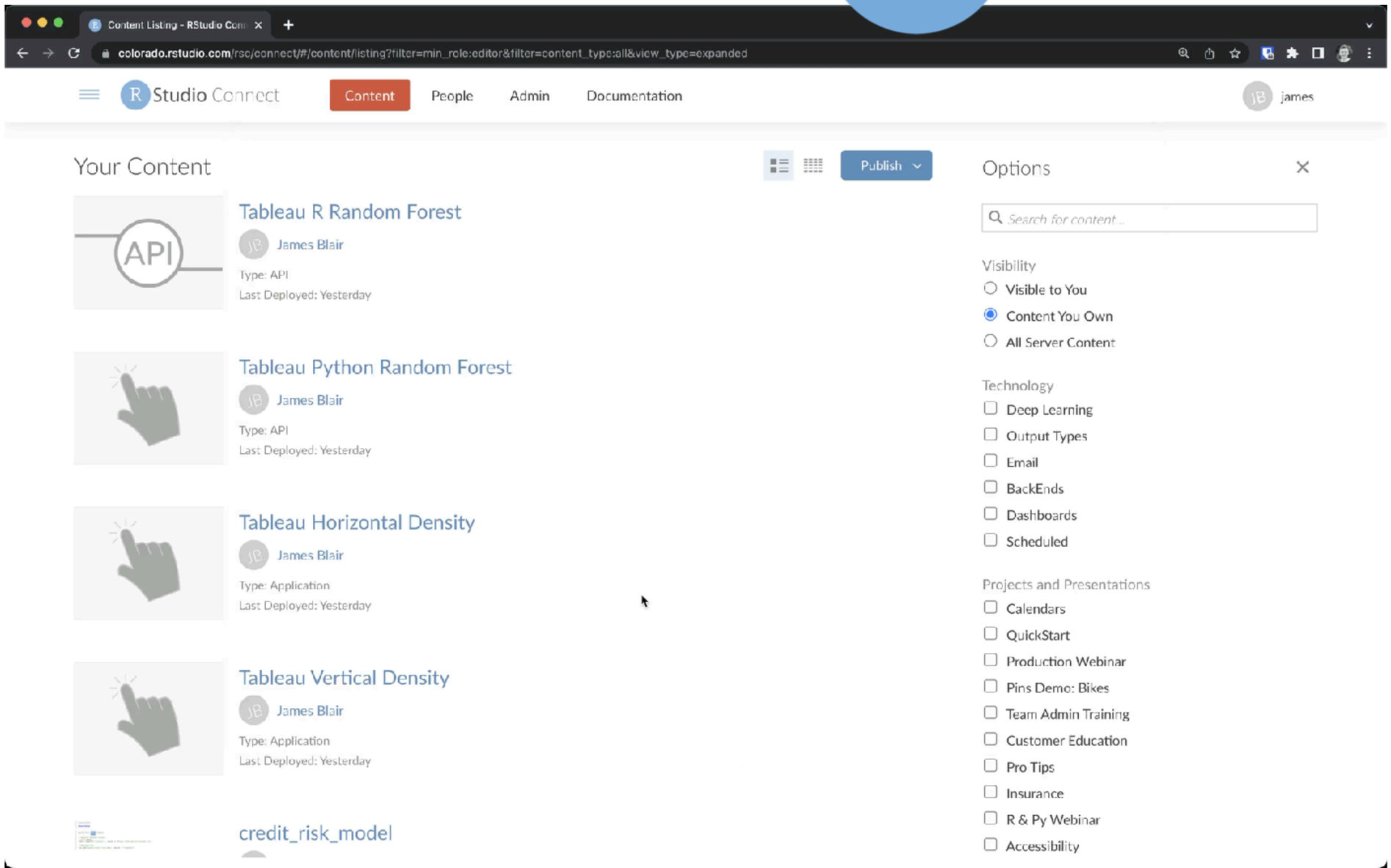
```python
from typing import List

from fastapitableau import FastAPITableau
from sklearn.ensemble import RandomForestRegressor
import numpy as np

app = FastAPITableau(
    title="Random Forest Regression",
    description="Fit a random forest to the input data",
    version="0.1.0"
)

@app.post("/rf")
def rf(x: List[float], y: List[float]) -> List[float]:
    X = np.array(x).reshape(-1, 1)
    regr = RandomForestRegressor()
    regr.fit(X, y)
    preds = regr.predict(X)
    return preds.tolist()
```

RStudio Connect



Tableau

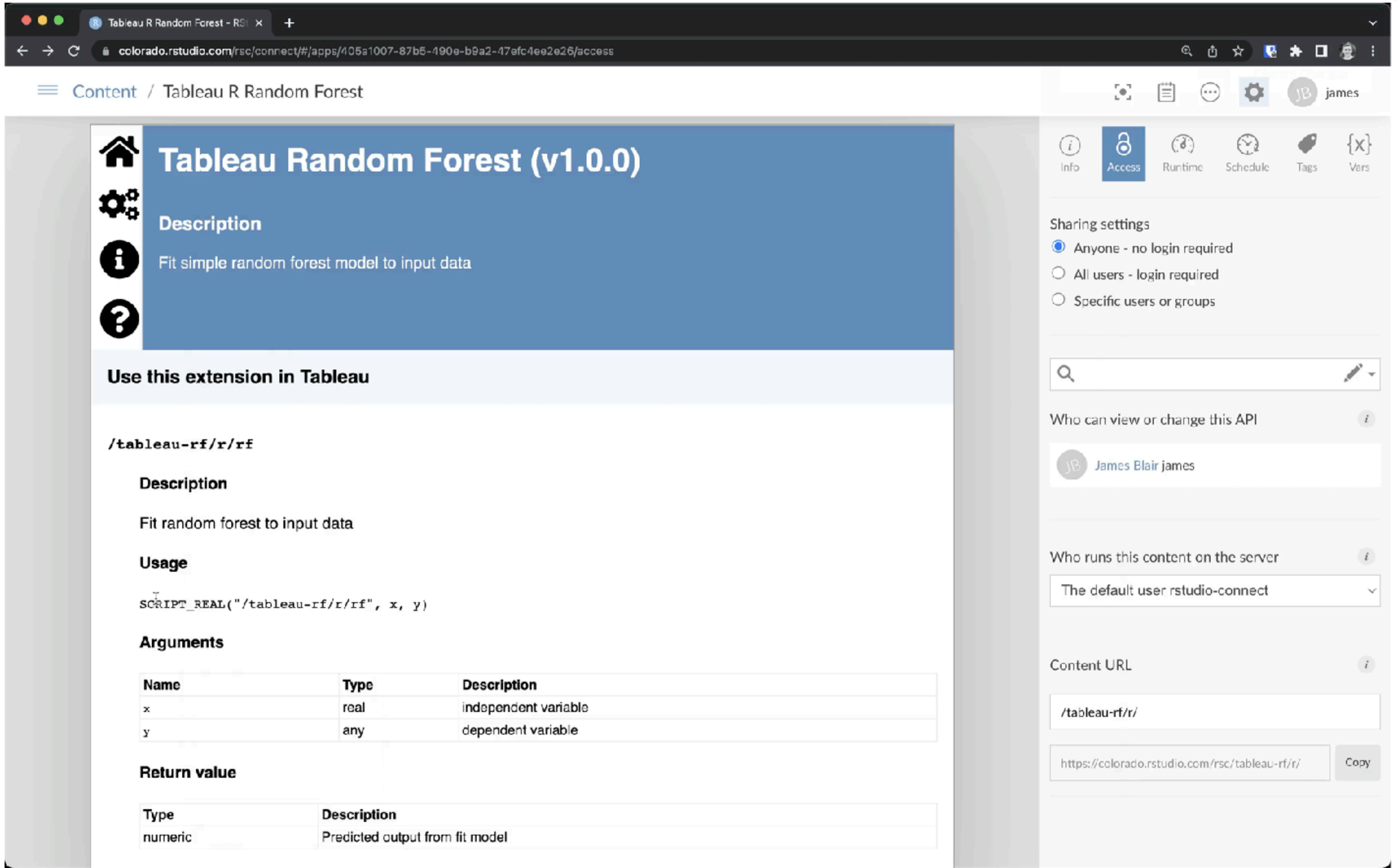


Tableau R Random Forest - RSI
colorado.rstudio.com/rsc/connect/#/apps/405a1007-87b5-490e-b9a2-47afc4ee2e26/access
Content / Tableau R Random Forest
james
Tableau Random Forest (v1.0.0)
Description
Fit simple random forest model to input data
Use this extension in Tableau
/tableau-rf/r/rf
Description
Fit random forest to input data
Usage
SCRIPT_REAL("/tableau-rf/r/rf", x, y)
Arguments
Name | Type | Description
x | real | independent variable
y | any | dependent variable
Return value
Type | Description
numeric | Predicted output from fit model
Info
Access
Runtime
Schedule
Tags
Vers
Sharing settings
Anyone - no login required
All users - login required
Specific users or groups
Who can view or change this API
James Blair james
Who runs this content on the server
The default user rstudio-connect
Content URL
/tableau-rf/r/
https://colorado.rstudio.com/rsc/tableau-rf/r/
Copy

All together now

Photo by Manuel Nägeli on Unsplash

Tableau - Music Dashboard

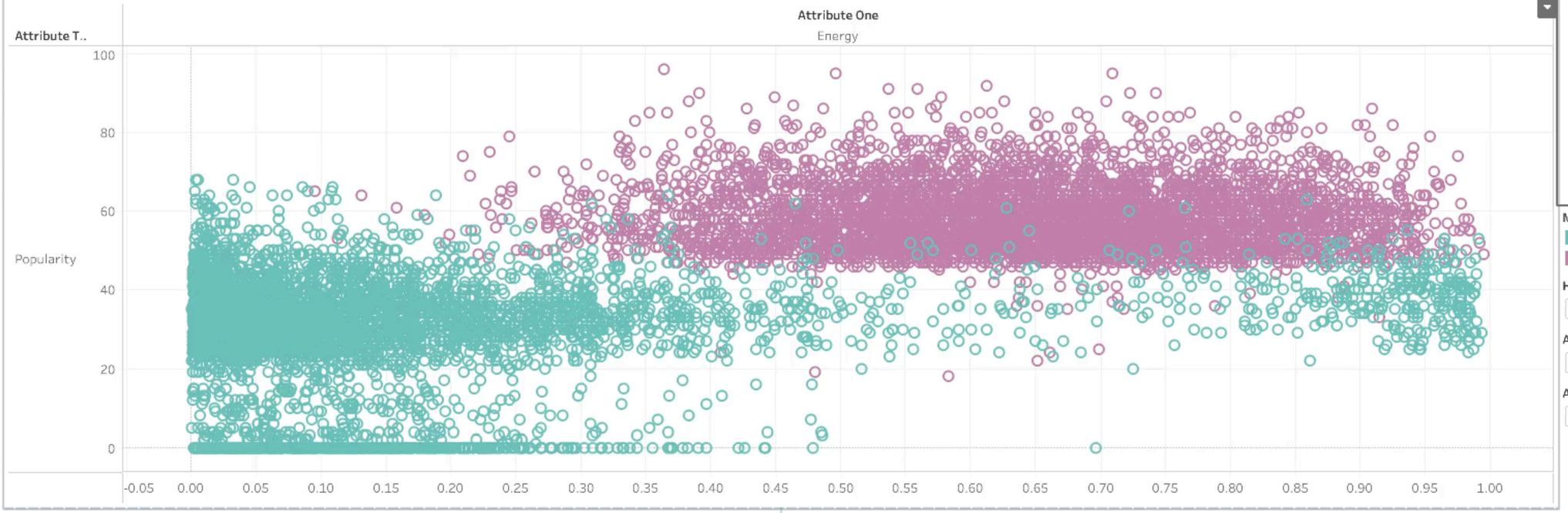
Entire View

Show Me

Dashboard

Layout

Music Attribute Explorer



Music Genre

- (All)
- Alternative
- Anime
- Blues
- Classical
- Country
- Electronic
- Hip-Hop
- Jazz
- Rap
- Rock

Music Genre

- Classical
- Hip-Hop

Highlight Music Genre

Highlight Music Genre

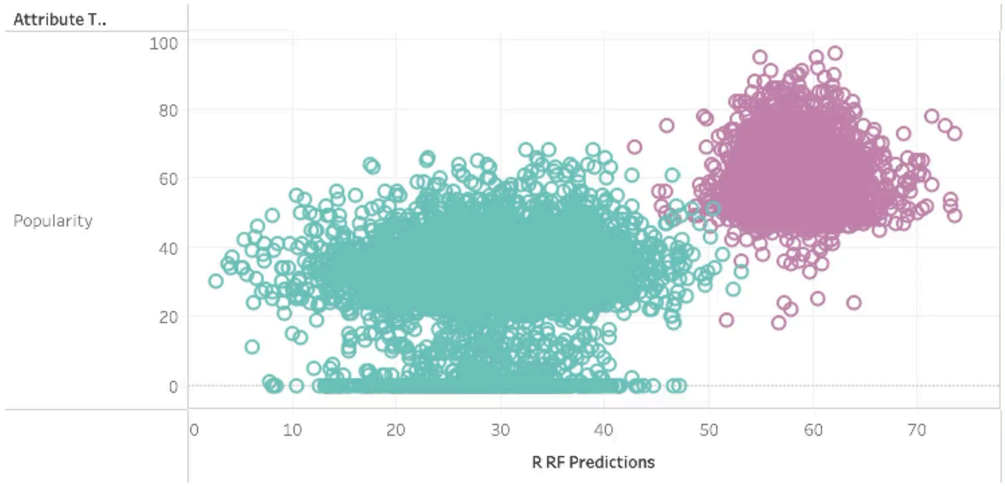
Attribute One

Energy

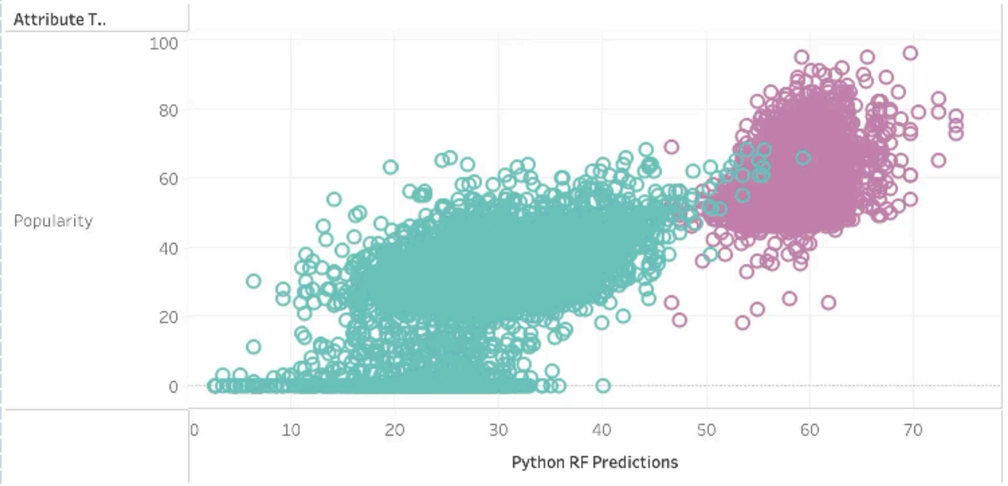
Attribute Two

Popularity

R RF Model



Python RF Model



Data Source | Scatter | R RF | Python RF | Music Explorer

10000 marks | 1 row by 1 column | SUM(X axis): 4,104.3423

James Blair

Tableau - Music Dashboard

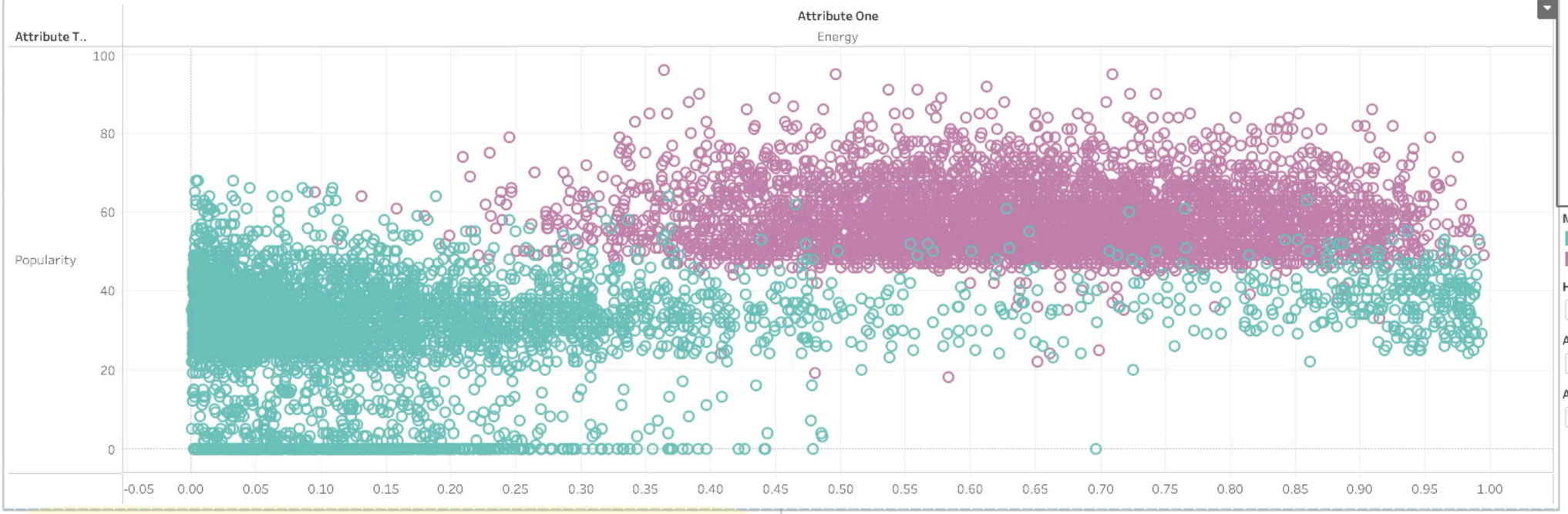
Entire View

Show Me

Dashboard

Layout

Music Attribute Explorer



Music Genre

- (All)
- Alternative
- Anime
- Blues
- Classical
- Country
- Electronic
- Hip-Hop
- Jazz
- Rap
- Rock

Music Genre

- Classical
- Hip-Hop

Highlight Music Genre

Highlight Music Genre

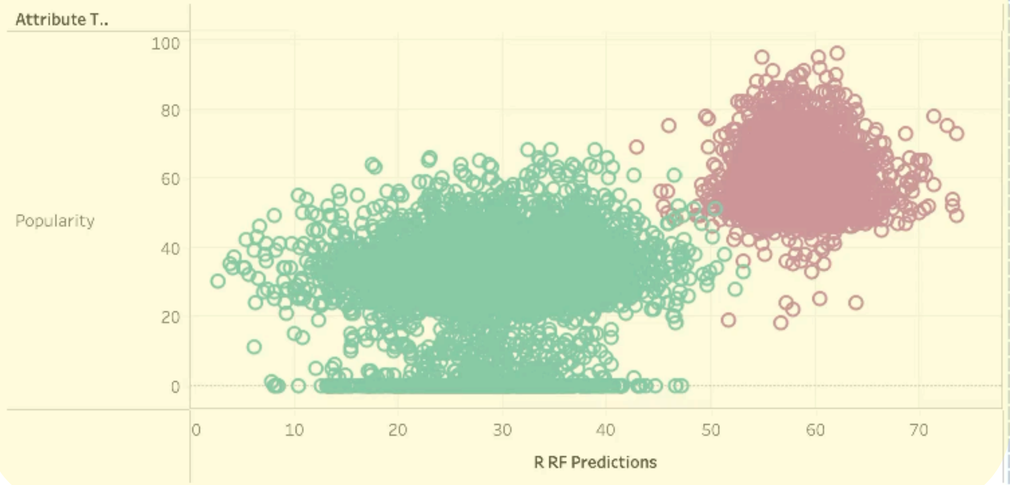
Attribute One

Energy

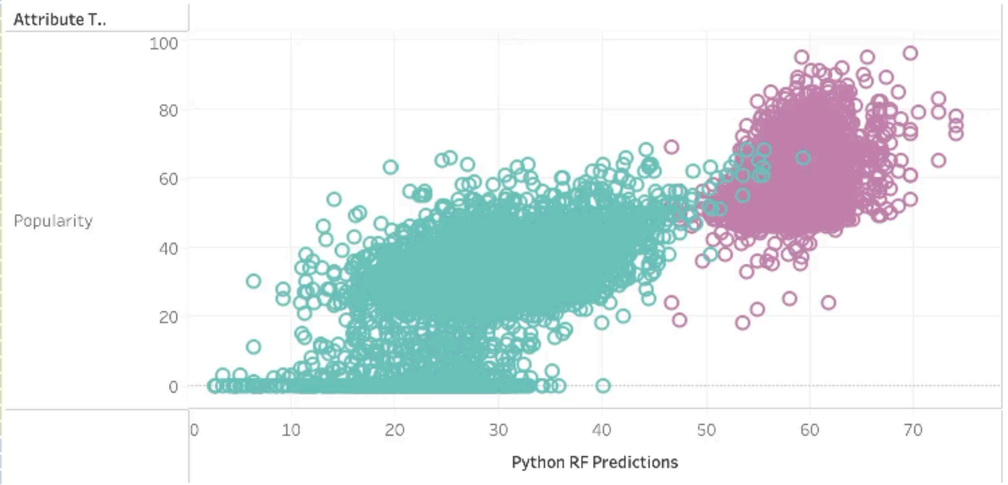
Attribute Two

Popularity

R RF Model



Python RF Model



Data Source | Scatter | R RF | Python RF | Music Explorer

10000 marks | 1 row by 1 column | SUM(X axis): 4,104.3423

James Blair

Tableau - Music Dashboard

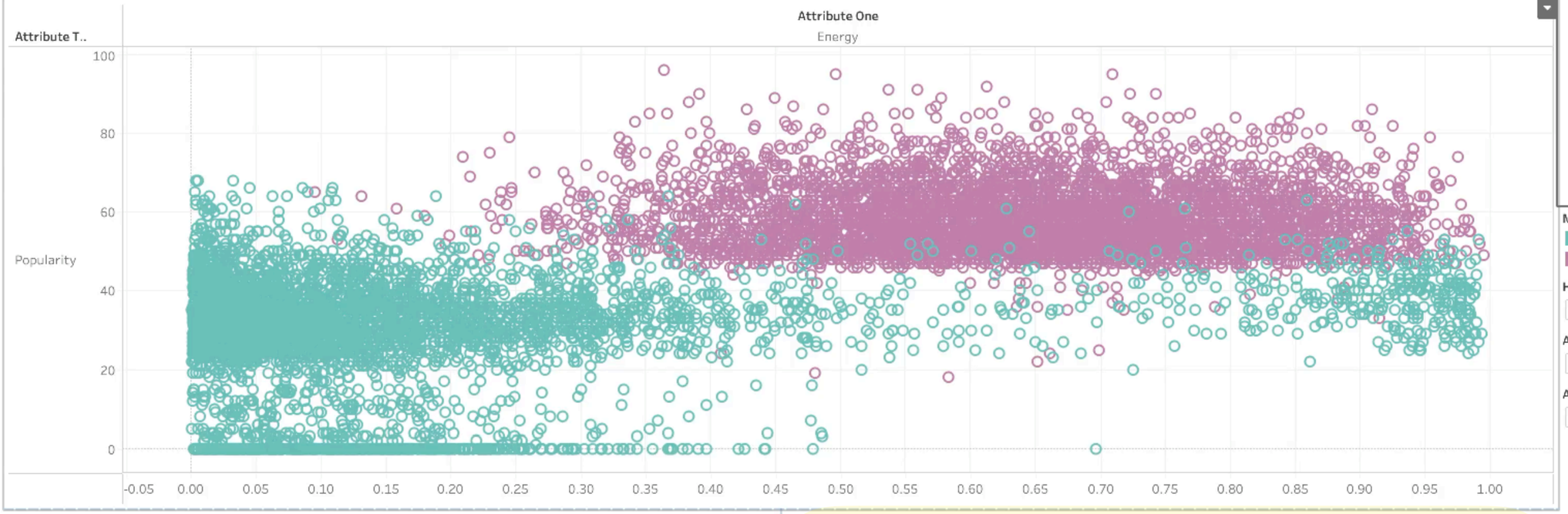
Entire View

Show Me

Dashboard

Layout

Music Attribute Explorer



Music Genre

- (All)
- Alternative
- Anime
- Blues
- Classical
- Country
- Electronic
- Hip-Hop
- Jazz
- Rap
- Rock

Music Genre

- Classical
- Hip-Hop

Highlight Music Genre

Highlight Music Genre

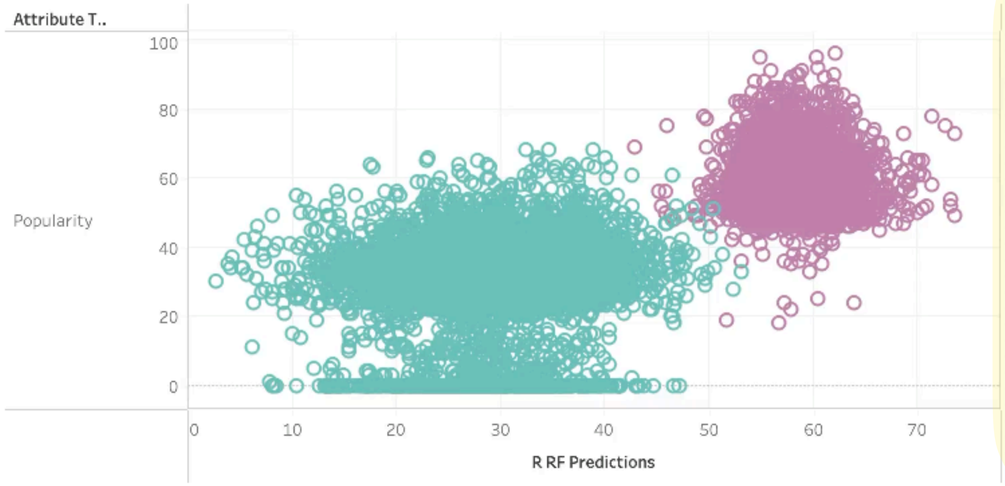
Attribute One

Energy

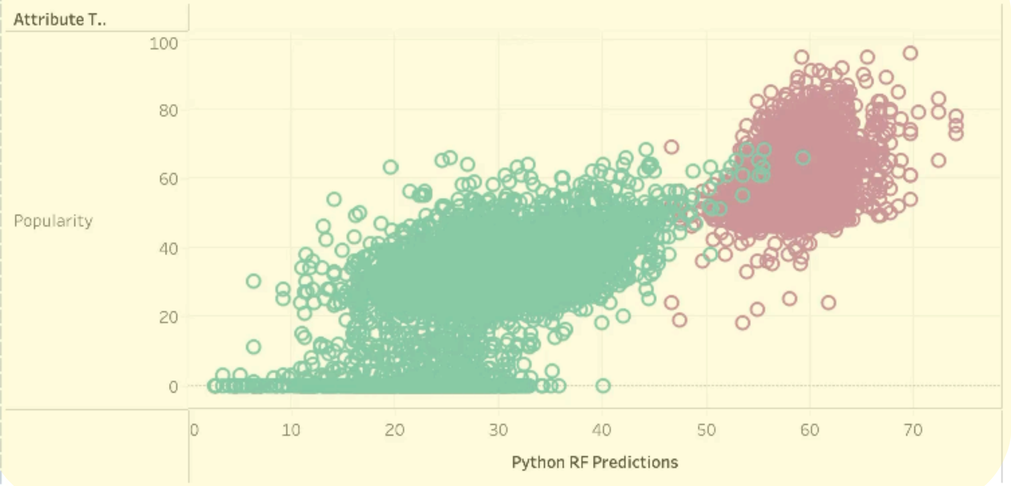
Attribute Two

Popularity

R RF Model



Python RF Model



Data Source | Scatter | R RF | Python RF | Music Explorer

10000 marks | 1 row by 1 column | SUM(X axis): 4,104.3423

James Blair

shinytableau

```yaml
extension_id: com.rstudio.extensions.horizontal-density-plot
extension_version: 0.1.0
name: Horizontal Density Plot
description: Density Plot for Tableau
extended_description: |
  An extension for creating a horizontal density plot for use within Tablaeu
  workbooks.
author_name: James Blair
author_email: james@rstudio.com
author_organization: RStudio
website: https://www.rstudio.com/

icon:
  file: default_icon.png
  package: shinytableau

permissions: full data

min_api_version: '1.4'
```

Once more, from the top

Photo by Kazuo ota on Unsplash

Tableau - Music Dashboard

Show Me

Dashboard | Layout

Default
Phone

Device Preview

Size

Automatic

Sheets

- Scatter
- R RF
- Python RF

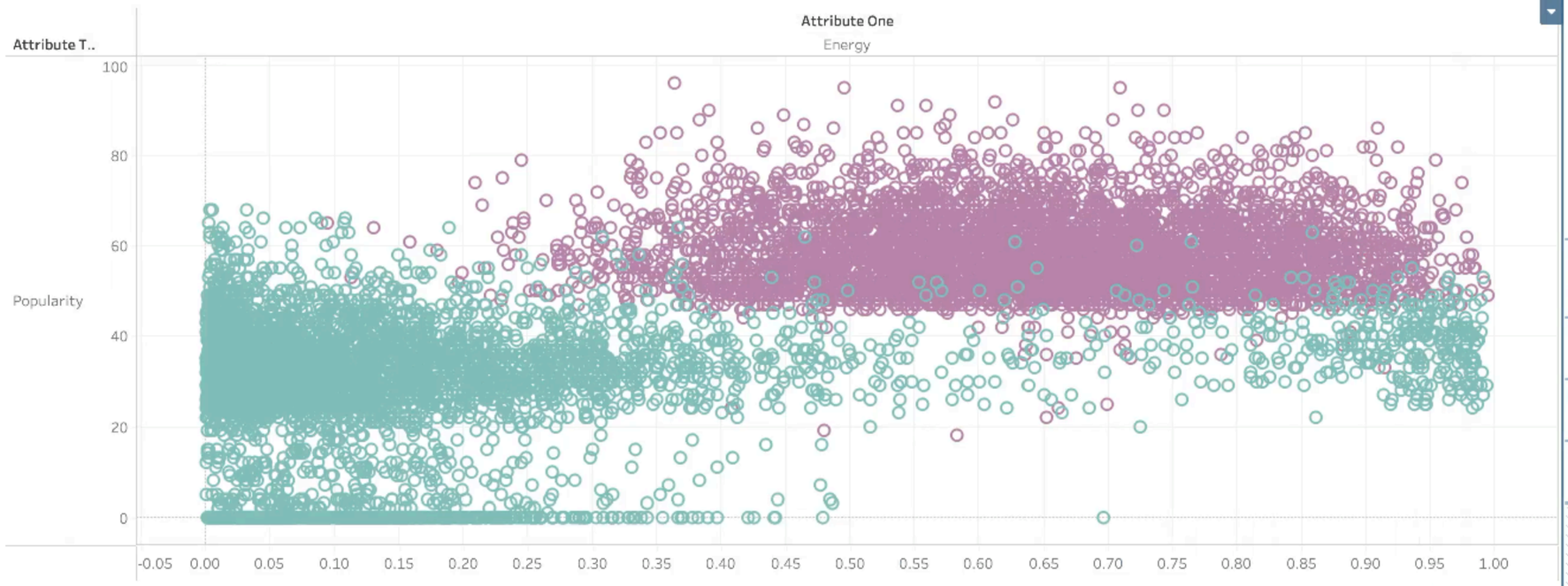
Objects

- Horizontal
- Vertical
- Text
- Image
- Web Page
- Data Story
- Blank
- Navigation
- Download
- Extension
- Ask Data

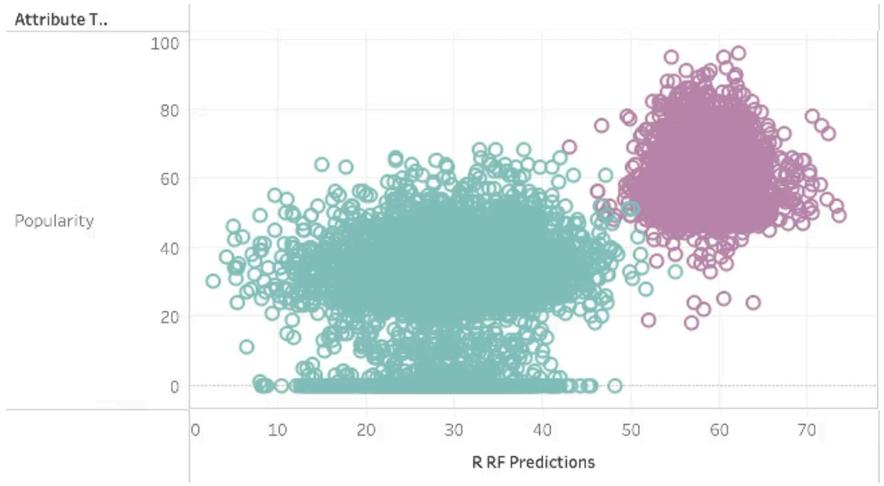
Tiled | Floating

Show dashboard title

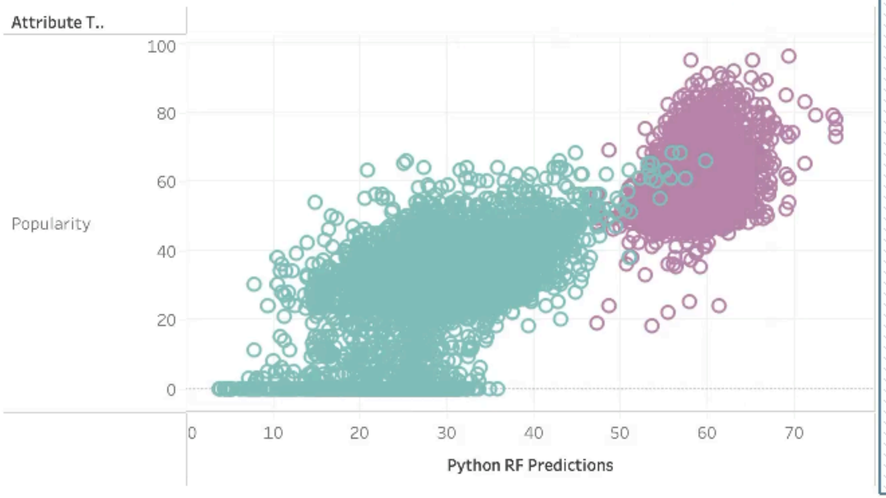
Music Attribute Explorer



R RF Model



Python RF Model



Music Genre

- (All)
- Alternative
- Anime
- Blues
- Classical
- Country
- Electronic
- Hip-Hop
- Jazz
- Rap
- Rock

Music Genre

- Classical
- Hip-Hop

Highlight Music Genre

Highlight Music Genre

Attribute One

Energy

Attribute Two

Popularity

Data Source | Scatter | R RF | Python RF | Music Explorer

James Blair

Use the right combination of tools for the job, and deliver a **symphony** of insight

Resources

- plumbertableau
- fastapitableau
- shinytableau
- Examples
- Talk repo

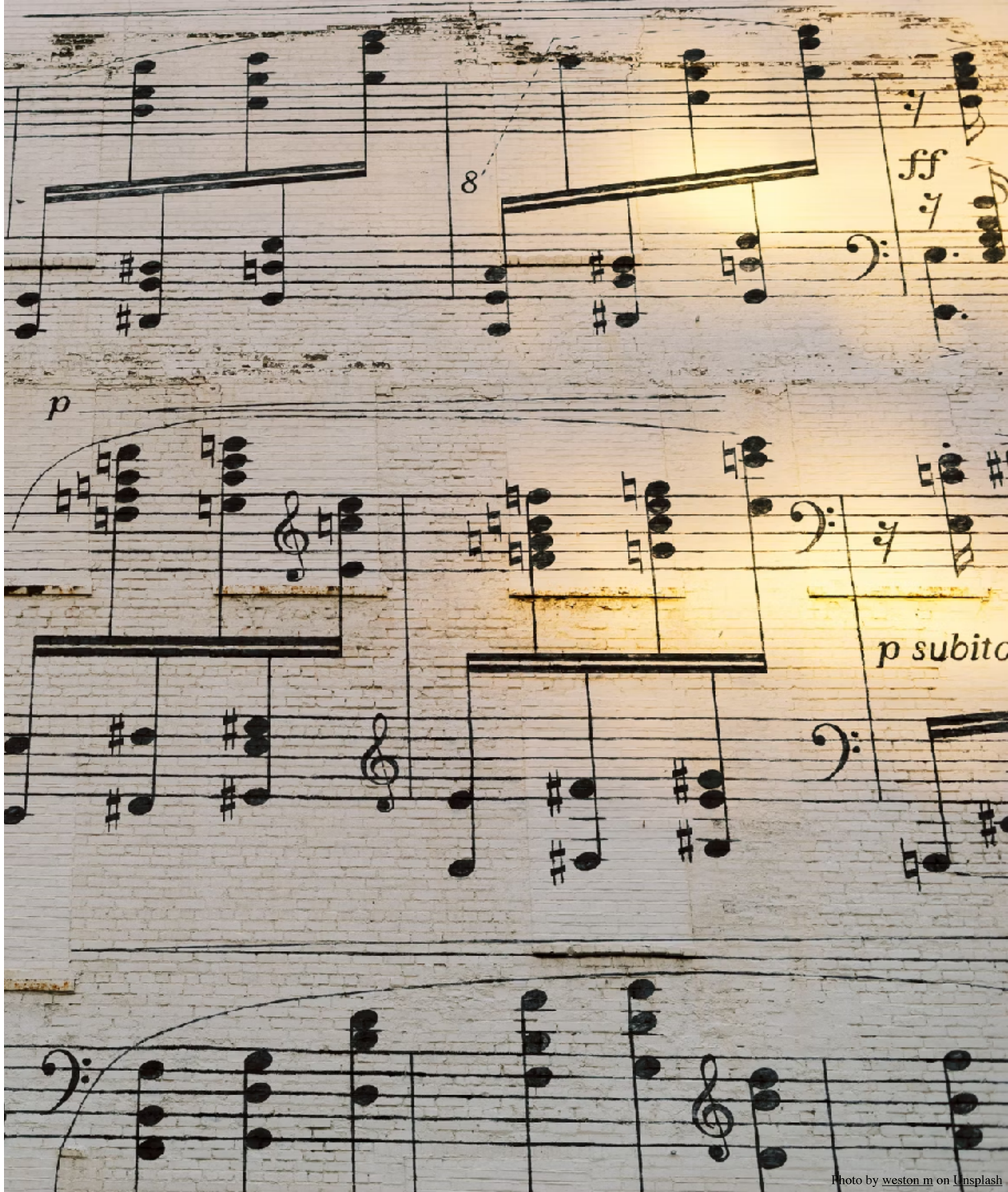


Photo by weston m on Unsplash

Photo by Joah Legg on Unsplash