# Stress Testing Machine Learning Population Change Projections with ACS MOEs

By Collin McCarter

*Opinions and ideas are my own disclaimer ... Feel free to reach out on LinkedIn for any data digression!*

Applied Demography Conference Annual Meeting, February 6th, 2024

https://www.linkedin.com/in/collin-mccarter/ ... https://github.com/collinmccarter-fl/stresstest_ACS_MOEs/

| | |
|---|---|
| 2022 - TODAY | ... Data Science, Northwestern Mutual |
| 2020 - 2022 | ... Technology & Product Management, Trellance (Credit Unions) |
| 2016 - 2020 | ... Demographer, Nielsen (TV Ratings) |
| 2014 - 2016 | ... PK-12 School Enrollment Analyst, State of Florida |

# Background

## Life as a data scientist ...

- Organizations ***link ACS data with a “client dataset”*** by small-area geographies
- Organizations are always searching for ***"new data sources"*** to differentiate the organization
- Discovering "new data sources" comes with a ***learning curve***
- ACS MOE's can be a ***time consuming*** learning curve for big data practitioners, especially those without a survey background
- ***Stress testing examples*** may help to lessen the learning curve, which is possible with MOE-adjusted ACS variables

## Life as an applied demographer ...

- Open source tools are ***efficient, accessible, and reproducible*** but are infrequently used in applied demography
- PIP packages are ***maturing*** in their documentation, peer-review credibility, and flexible options.
- Machine Learning practitioners are using these tools and data ***without domain knowledge***
- Applied Demographers ***have the domain knowledge*** to maximize value from machine learning tools used in demographic analysis

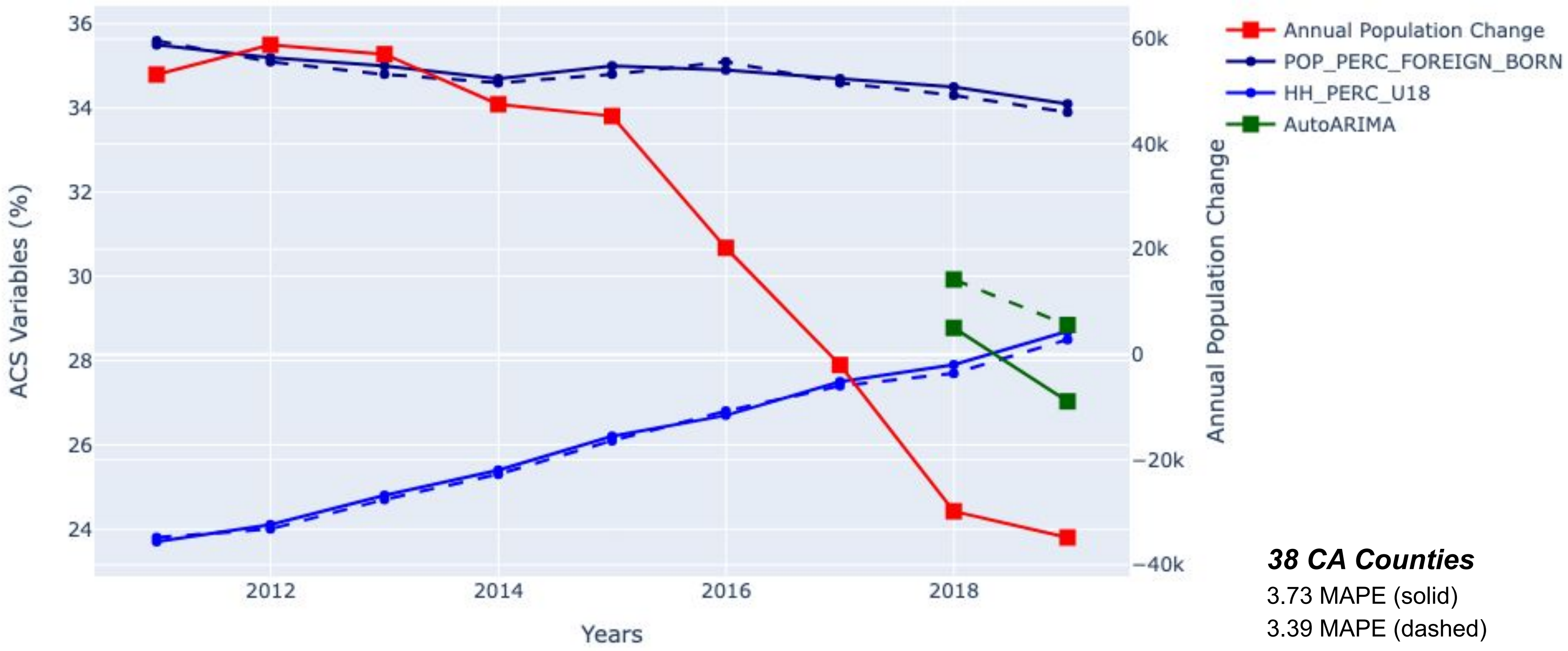
# What I did ... stress tested ACS1yr CA County projections

Python Jupyter Notebook completes the below in ***< 3min***

- ***1 sec*** Gather packages of functions (aka procedures) from PIP
- ***9 sec*** Pull select api.census.gov data
  - ACS 1yr Data Profile
  - PEP Vintage 2010
  - PEP Vintage 2020
  - PEP Vintage 2022
- ***1 sec*** Transform and Merge US Census data into a training dataset for machine learning packages
- ***150 sec*** Train (fit), Predict, and Evaluate different projections based on below adjustments:
  - ACS variables used as predictive features vs univariate timeseries
  - Adjusted ACS variables using MOEs vs ACS variable point estimates
  - Continuous Regression metrics (MAE, MSE, MAPE, r^2)
  - Models available on PIP (structureboost, statsforecast, catboost)

*Note: stress testing as shown does not account for other uncertainty methods such as bayesian weighting with MOEs.*

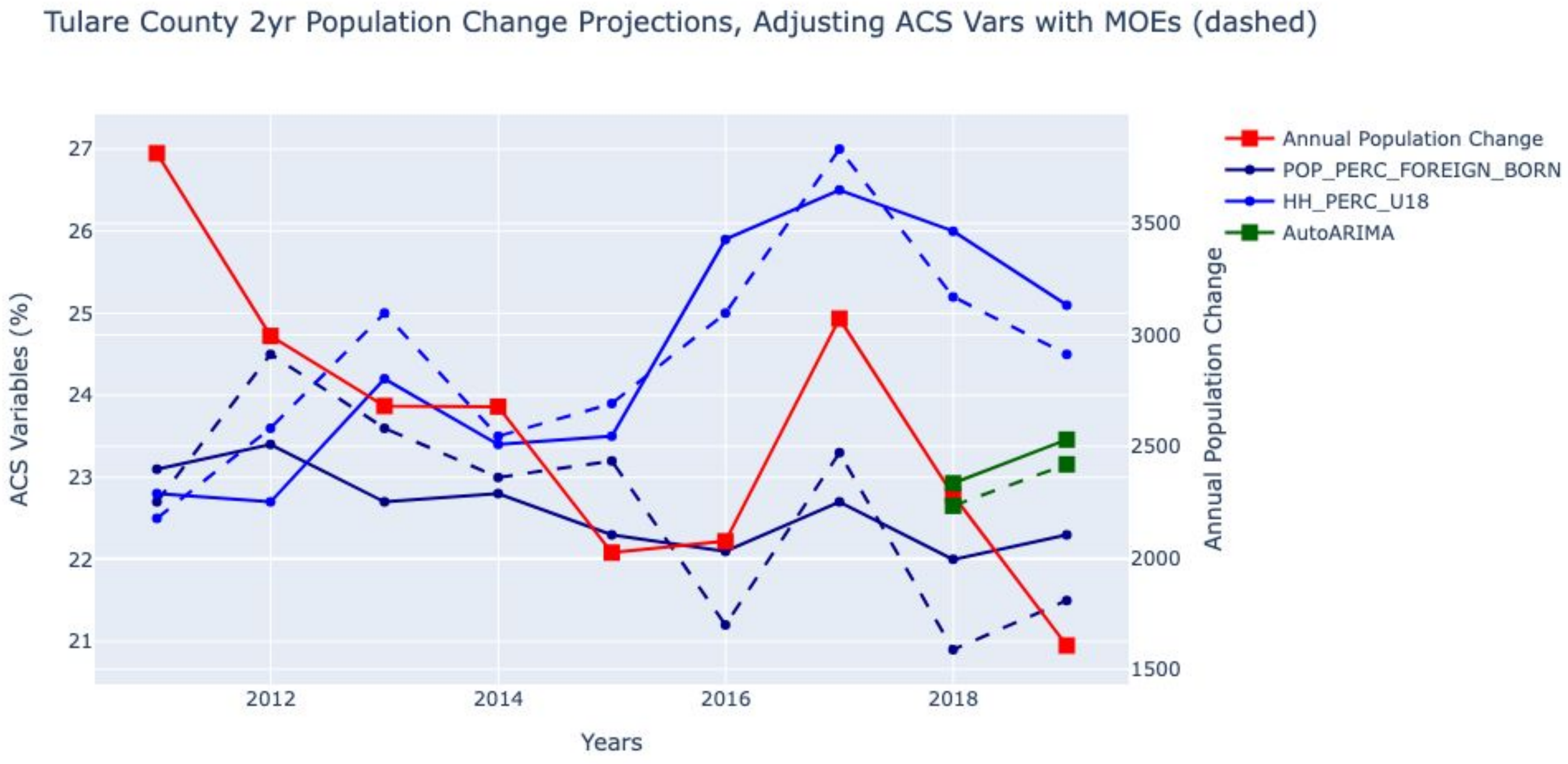
## Los Angeles County 2yr Population Change Projections, Adjusting ACS Vars with MOEs (dashed)



***38 CA Counties***

3.73 MAPE (solid)

3.39 MAPE (dashed)



Tulare County 2yr Population Change Projections, Adjusting ACS Vars with MOEs (dashed)

1

# What I did ... PIP (package installer for Python)

- ***1 sec*** Gather packages of functions (aka procedures) from PIP

| | | | |
|---|---|---|---|
| scikit learn | nixtla | numeristical | |
| **General Machine Learning** | **Time Series Forecasting** | **StructureBoost** | **Pygris - TIGER Mapping** |
| **...** | **...** | **...** | **...** |
| **National Institute for Research in Digital Science and Technology** | **Start-up from industry experts funded by investors** | **Brian Lucena, PhD (Brown) consultant** | **Kyle Walker, Associate Professor Texas Christian University** |

2

# What I did ... api.census.gov

- ***9 sec*** Pull select api.census.gov data

| | 2008 | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 | 2021 | 2022 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ACS 1yr Profile | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | | | |
| PEPANNRES | | | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 |

### ACS 1yr Profile

```
state_fips = '06' # California
api_table  = 'acs/acs1/profile'
var_strlist = 'DP03_0085PE,DP03_0085PM,..,DP03_0004PE,DP03_0004PM'
years        = ['2019','2018',...,'2009','2008']

url =
f"https://api.census.gov/data/{year}/{api_table}?get={var_strlist}&for=county:*
&in=state:{state_fips}&key={api_key}"
```

### PEPANNRES

```
state_fips = '06' # California
api_table   = 'pep/population'
var_strlist = 'POP,DATE_CODE,DATE_DESC,NAME,GEO_ID'
year            = '2019'

url =
f"https://api.census.gov/data/{year}/{api_table}?get={var_strlist}&for=county:*&in
=state:{state_fips}&key={api_key}"
```

*PEP2022 required using ftp site with csv file due to API availability

3

# What I did ... transform for machine learning

- ***1 sec*** Transform US Census data for machine learning packages

### Supervised Machine Learning - Data Prep

```
forecast_horizon_years = 2

dates   = Y_df['ds'].unique()
dtrain  = dates[:-h_var]
dtest   = dates[-h_var:]
Y_train = Y_ts.query('ds in @dtrain')
Y_test  = Y_ts.query('ds in @dtest')
X_train = X_ts.query('ds in @dtrain')
X_test  = X_ts.query('ds in @dtest')

train = Y_train.merge(X_ts, how = 'left', on = ['unique_id', 'ds'])
```

### Adjust ACS 1yr Estimates within MOE Range

```
e_adj = np.round(
          np.random.uniform(
                    acs_e  -  acs_moe , acs_e  +  acs_moe),1)
```

4

# What I did ... train, predict, evaluate projections

- ***150 sec*** Train (fit), Predict, and Evaluate different projections adjustments

## Train & Predict

```
models = [Naive(), AutoARIMA()] # Add more models from Statsforecast

sf = StatsForecast(models=models,freq='A',n_jobs=-1)

horizon = 2
level    = [95]

fcst = sf.forecast(df=train, h=horizon, X_df=X_test, level=level, fitted=True)
```

## Evaluate

```
metrics = [mae_calc, mse_calc, mape_calc, r2_calc]

for metric in metrics:
      print(metric(fcst,0))
```

## What just happened?

PIP Packages
US Census API
Machine Learning Insights
Role for Applied Demographers
ACS MOEs in Machine Learning

# Thank You!

Collin McCarter