# Trends in Traffic Stops

Abhay Singh, Krupa Hegde and Rutuja Gurav

# Outline

- Introduction
- Motivation
- The Data
- The Experiments
- Our Implementations
- Analysis and Results
- Conclusion

# 1.

# Introduction

# How did we get started?

- Analyzing traffic data is a hot topic.
- Driving forces
  - Rise of autonomous vehicles
  - Data-driven social policing

How does our data qualify as

# Big Data?

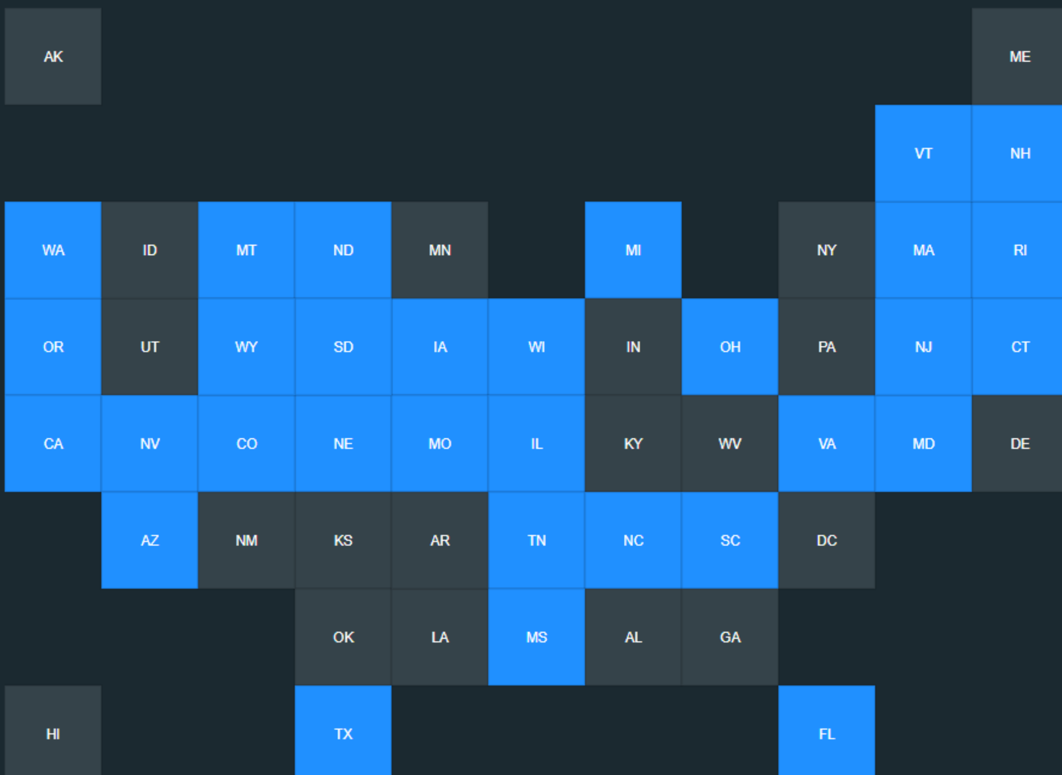
How does our data qualify as

# Big Data?

- Volume
  - Across the United States, police officers make more than 50,000 traffic stops on a typical day.
- Variety
  - Data of these traffic stops is stored in a variety of formats and with high inconsistency across states.

# 2.

# Motivation




# THE STANFORD **OPEN POLICING** PROJECT

On a typical day in the United States, police officers make more than 50,000 traffic stops. Our team is gathering, analyzing, and releasing records from millions of traffic stops by law enforcement agencies across the country. Our goal is to help researchers, journalists, and policymakers investigate and improve interactions between police and the public.

VIEW DATA

# 3.

# The Data

## Features:

- Stop Date
- Stop Time
- Stop Location
- Driver Race
- Driver Gender
- Driver Age
- Stop Reason
- Search Conducted
- Search Type
- Is Arrested
- Contraband Found
- Stop Outcome

| State | Stops | Time Range | Stop Date | Stop Time | Stop Location | Driver Race | Driver Gender | Driver Age | Stop Reason | Search Conducted | Search Type | Contraband Found | Stop Outcome |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Arizona | 2,251,992 | 2009–2015 | ■ | ■ | ■ | ■ | ■ | | | ■ | | ■ | ■ |
| California | 31,778,515 | 2009–2016 | ■ | | ■ | ■ | ■ | | ■ | ■ | ■ | | ■ |
| Colorado | 2,584,744 | 2010–2016 | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | |
| Connecticut | 318,669 | 2013–2015 | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| Florida | 5,421,446 | 2010–2016 | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | ■ |
| Illinois | 4,715,031 | 2004–2015 | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| Iowa | 2,441,335 | 2006–2016 | ■ | ■ | | | | | ■ | | | | ■ |
| Maryland | 1,113,929 | 2007–2014 | | | | ■ | ■ | | ■ | ■ | | ■ | ■ |
| Massachusetts | 3,418,298 | 2005–2015 | ■ | | ■ | ■ | ■ | ■ | | ■ | ■ | ■ | ■ |
| Michigan | 709,699 | 2001–2016 | ■ | ■ | ■ | | | | ■ | | | | ■ |
| Mississippi | 215,304 | 2013–2016 | ■ | | ■ | ■ | ■ | ■ | ■ | | | | |
| Missouri | 2,292,492 | 2010–2015 | | | | ■ | | | | ■ | | ■ | |
| Montana | 825,118 | 2009–2016 | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | ■ |
| Nebraska | 4,277,921 | 2002–2014 | | | | ■ | | | | ■ | | | |
| Nevada | 737,294 | 2012–2016 | ■ | | | ■ | | ■ | ■ | | | | ■ |
| New Hampshire | 259,822 | 2014–2015 | ■ | ■ | ■ | | ■ | | ■ | | | | ■ |
| New Jersey | 3,845,335 | 2009–2016 | ■ | ■ | ■ | ■ | ■ | | ■ | | | | ■ |
| North Carolina | 9,558,084 | 2000–2015 | ■ | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| North Dakota | 330,063 | 2010–2015 | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | | | |
| Ohio | 6,165,997 | 2010–2015 | ■ | ■ | ■ | ■ | ■ | | | ■ | | | |
| Oregon | 1,143,017 | 2010–2016 | | | | ■ | | | | | | | |
| Rhode Island | 509,681 | 2005–2015 | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| South Carolina | 8,440,934 | 2005–2016 | ■ | | ■ | ■ | ■ | ■ | | ■ | | ■ | ■ |
| South Dakota | 281,249 | 2012–2015 | ■ | ■ | ■ | | ■ | | ■ | | | | ■ |
| Tennessee | 3,829,082 | 1996–2016 | ■ | ■ | ■ | ■ | ■ | | ■ | | | | ■ |
| Texas | 23,397,249 | 2006–2015 | ■ | ■ | ■ | ■ | ■ | | ■ | ■ | ■ | ■ | ■ |
| Vermont | 283,285 | 2010–2015 | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| Virginia | 5,006,725 | 2006–2016 | ■ | | ■ | ■ | | | | ■ | | | |
| Washington | 8,624,032 | 2009–2016 | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| Wisconsin | 1,059,033 | 2010–2016 | ■ | ■ | ■ | ■ | ■ | | ■ | ■ | ■ | ■ | ■ |
| Wyoming | 173,455 | 2011–2012 | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | | | |

| id | state | stop_date | stop_time | location_raw | county_name | county_fips | fine_grained_location | police_department | driver_gender | driver_age_raw | driver_age | driver_race_raw | driver_race | violation_raw |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FL-2010-000013 | FL | 2010-01-15 | 13:26 | LEON | Leon County | 12073 | | | | | | W | White | SPEED |
| FL-2010-000014 | FL | 2010-01-15 | 13:55 | GADSDEN | Gadsden County | 12039 | | | M | 32 | 32.0 | B | Black | EXPIRED TAG (6 MONTHS OR LESS)\|FAULTY EQUIPMENT |
| FL-2010-000015 | FL | 2010-01-16 | 08:23 | CALHOUN | Calhoun County | 12013 | | | F | 45 | 45.0 | W | White | DUI |
| FL-2010-000016 | FL | 2010-01-16 | 09:50 | TAYLOR | Taylor County | 12123 | PERRY | | M | 71 | 71.0 | B | Black | |
| FL-2010-000017 | FL | 2010-01-17 | 01:05 | GADSDEN | Gadsden County | 12039 | | | M | 57 | 57.0 | W | White | NO REGISTRATION\|SP |
| FL-2010-000018 | FL | 2010-01-17 | 02:22 | LEON | Leon County | 12073 | TALLAHASSEE | | F | 22 | 22.0 | B | Black | FAULTY EQUIPMEN REGISTRATION\|OT TAG / REGISTRATI VIOLATIONS |
| FL-2010-000019 | FL | 2010-01-17 | 11:10 | GADSDEN | Gadsden County | 12039 | | | M | 25 | 25.0 | B | Black | FAILURE TO EXHIB UPON DEMAND\|SP |
| FL-2010-000020 | FL | 2010-01-17 | 12:49 | GADSDEN | Gadsden County | 12039 | MIDWAY | | F | 20 | 20.0 | W | White | SPEED |
| FL-2010-000021 | FL | 2010-01-17 | 13:21 | TAYLOR | Taylor County | 12123 | PERRY | | F | 20 | 20.0 | W | White | |

# 3.

# The Experiments

- Perform aggregations to find yearly number of stops at state level to see increase or decrease in trend.
- Number of stops conducted for different age groups, gender or race.
- Decision tree to predict likelihood of events.
- Build a logistic regression model to see if age, race or gender determines, even weakly the possibility of being arrested, being searched or having found a contraband.

# 4.

# Our Implementation

# Spark DataFrame API

## Spark SQL

```
Dataset<Row> genStopData =
bucketedStopData.select("state","stop_date","county_name","driver
_gender")
.groupBy(year(col("stop_date")),col("driver_gender"),col("state")
)
.count()
.withColumnRenamed("year(stop_date)","yearStop")
.withColumnRenamed("count","Count");
```

## Spark Mllib

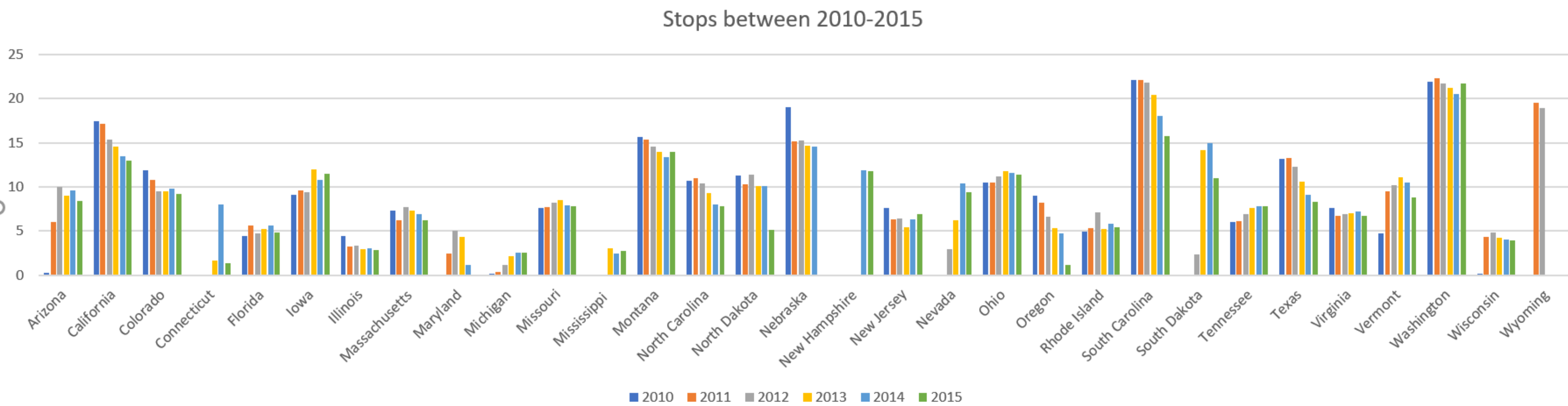
- Bucketizer
- VectorIndexerModel
- StringIndexer
- VectorAssembler
- Pipeline

- Bucketizer
  - Transforms a column of continuous features to a column of feature buckets.
- VectorIndexerModel
  - Helps index categorical features in datasets of vectors.
- StringIndexer
  - Encodes a string column of labels to a column of label indices.
- VectorAssembler
  - A transformer that combines a given list of columns into a single vector column.
- Pipeline
  - A Pipeline chains multiple Transformers and Estimators together to specify an ML workflow.

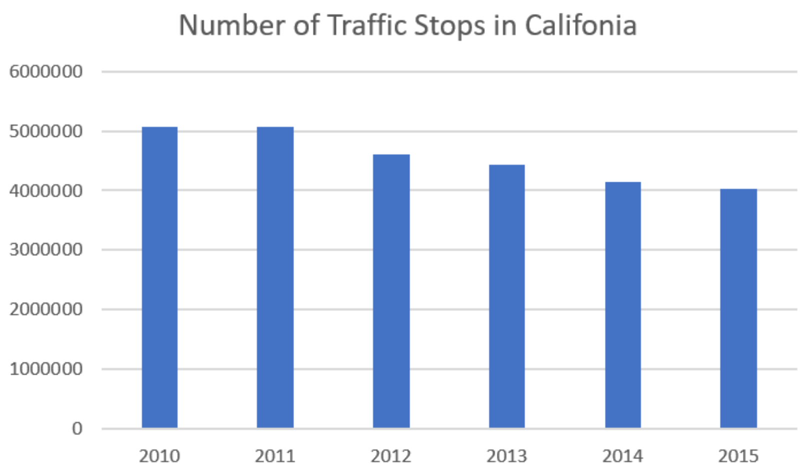
# 4.

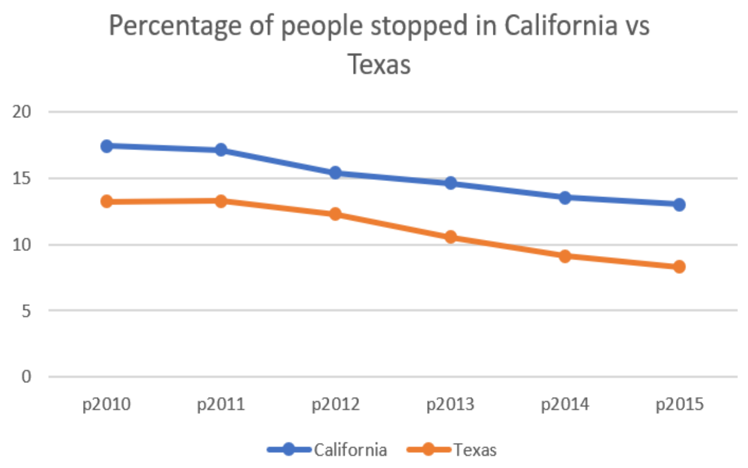
# Analysis Results

Aggregations to find yearly number of stops at state level to see increase or decrease in trend.

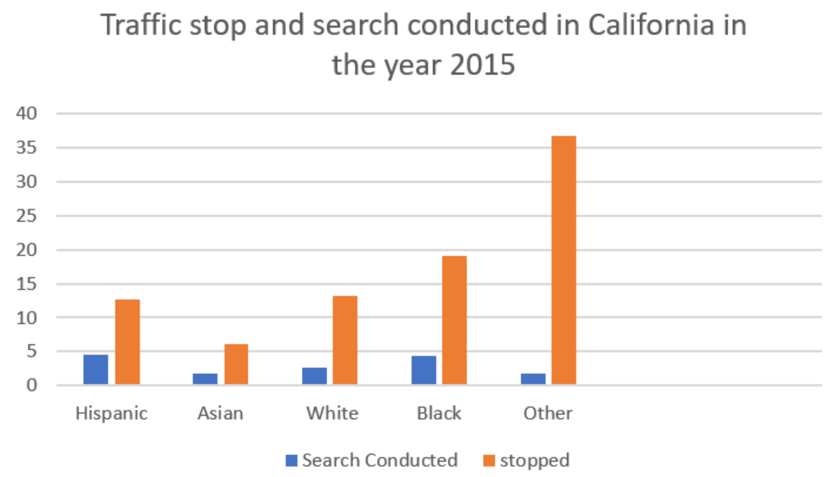


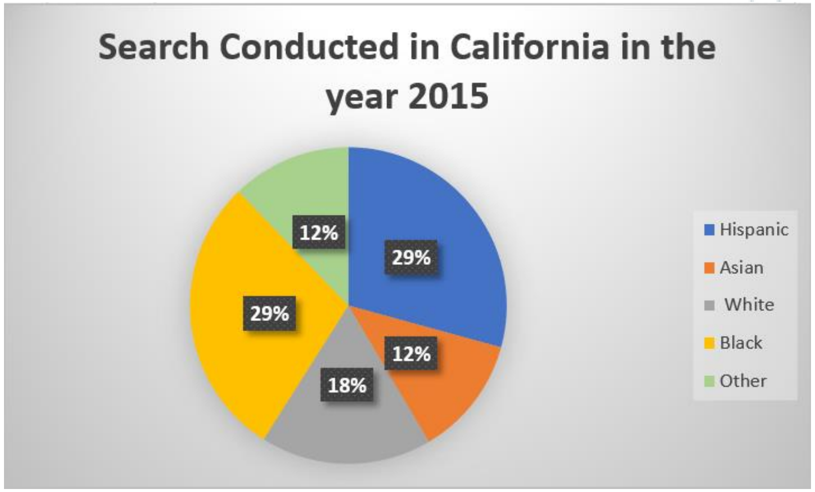
Aggregations to find yearly number of stops at state level to see increase or decrease in trend.

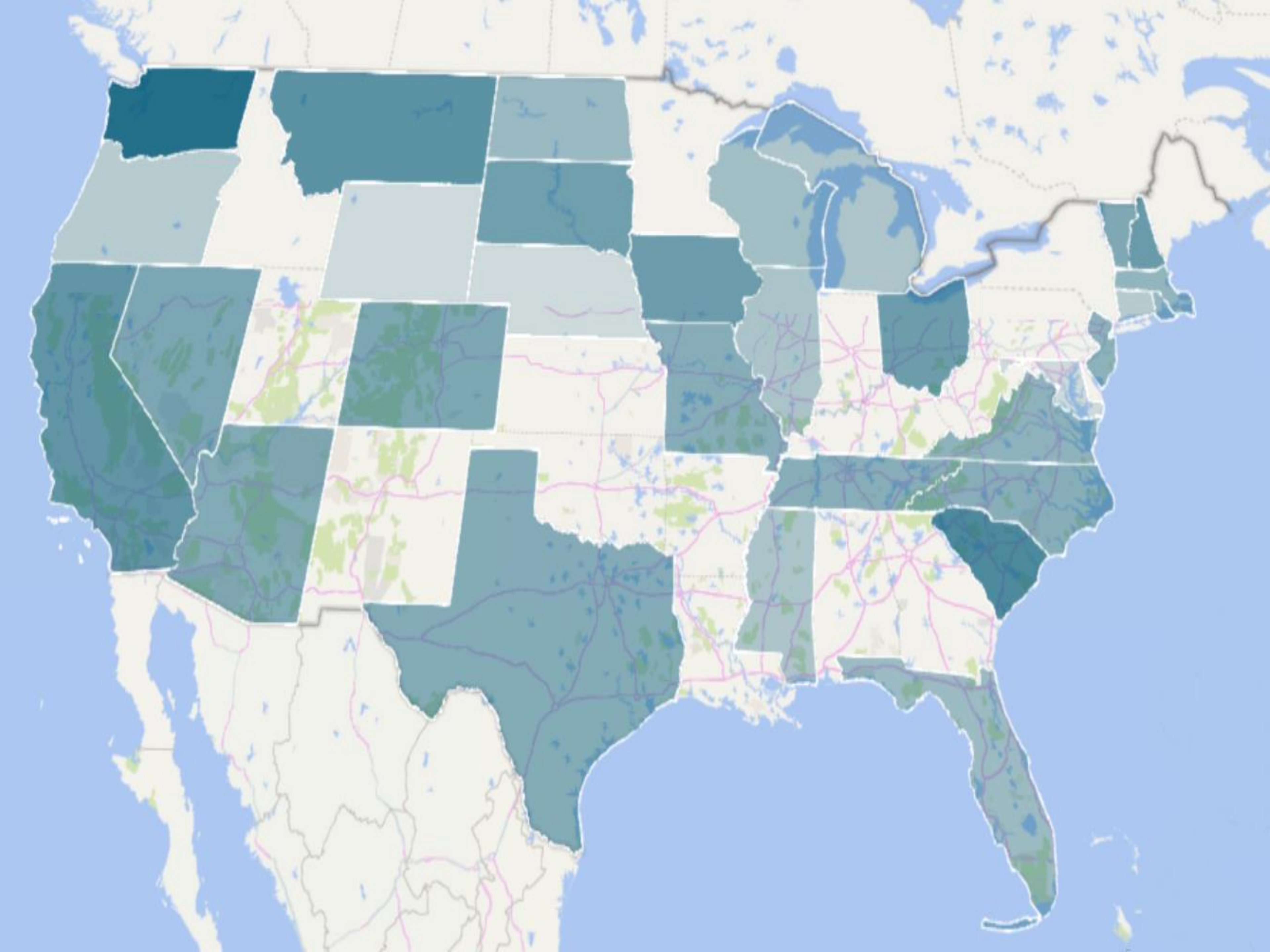


# Search Conducted



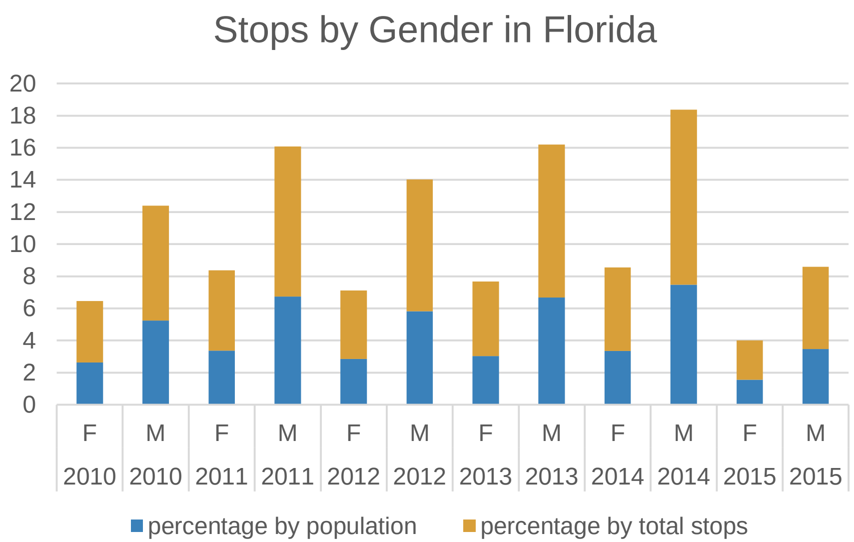



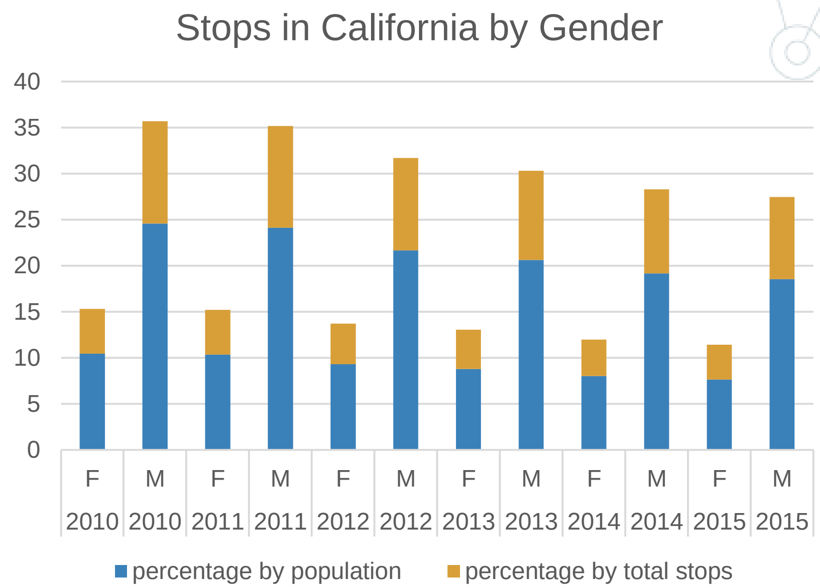


# Decision Tree

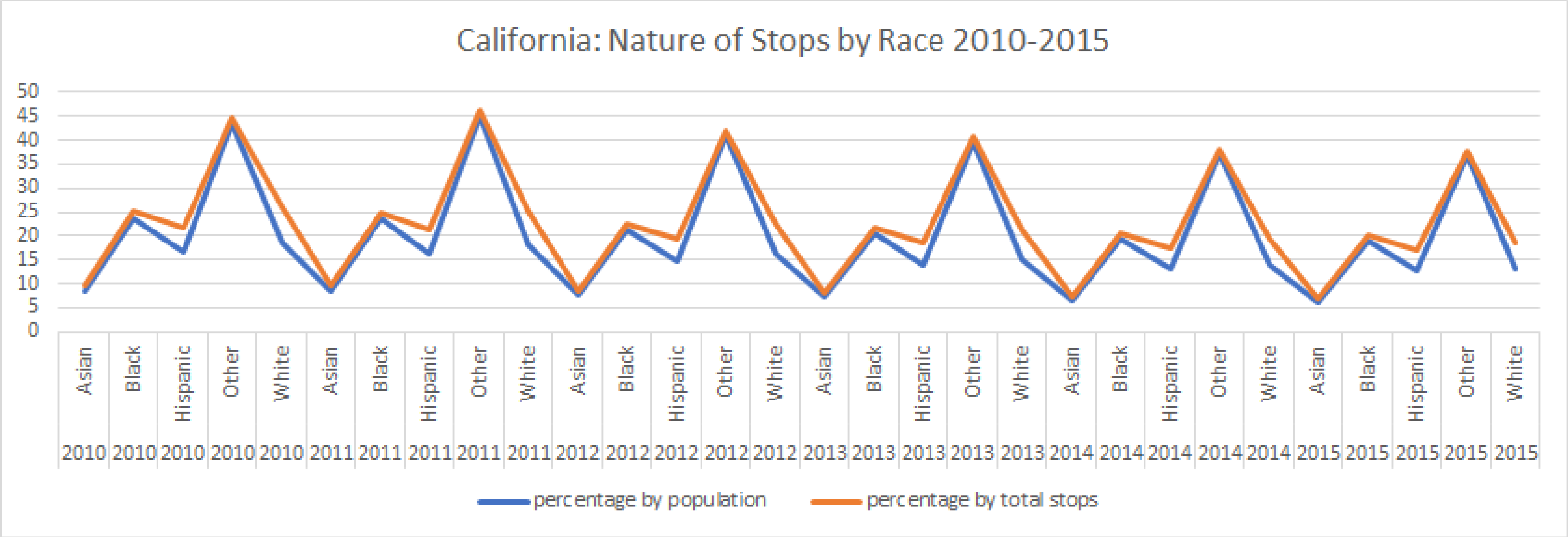
- Feature Columns: Driver Race, Gender, Age, Officer Race
- Label Column: Search_conducted
- Challenges: Unbalanced Data-Difficult to sample
- Without sampling
  - Test Error= 0.0039620514864309175
  - Weighted Precision= 0.9920915948791192
- With Sampling
  - Weighted Precision=0.673742738866459
  - Weighted Recall =0.681081081081081
  - Test Error = 0.3189189189189189Message Input

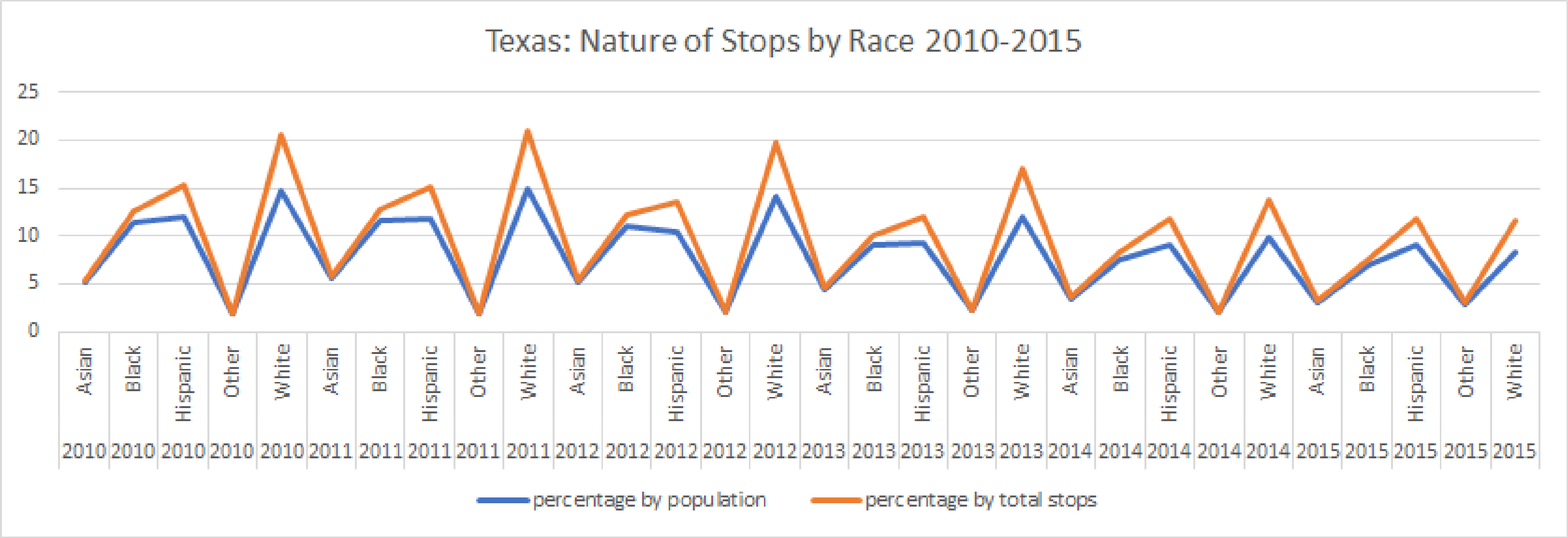
# Trends in Number of stops conducted by Gender.



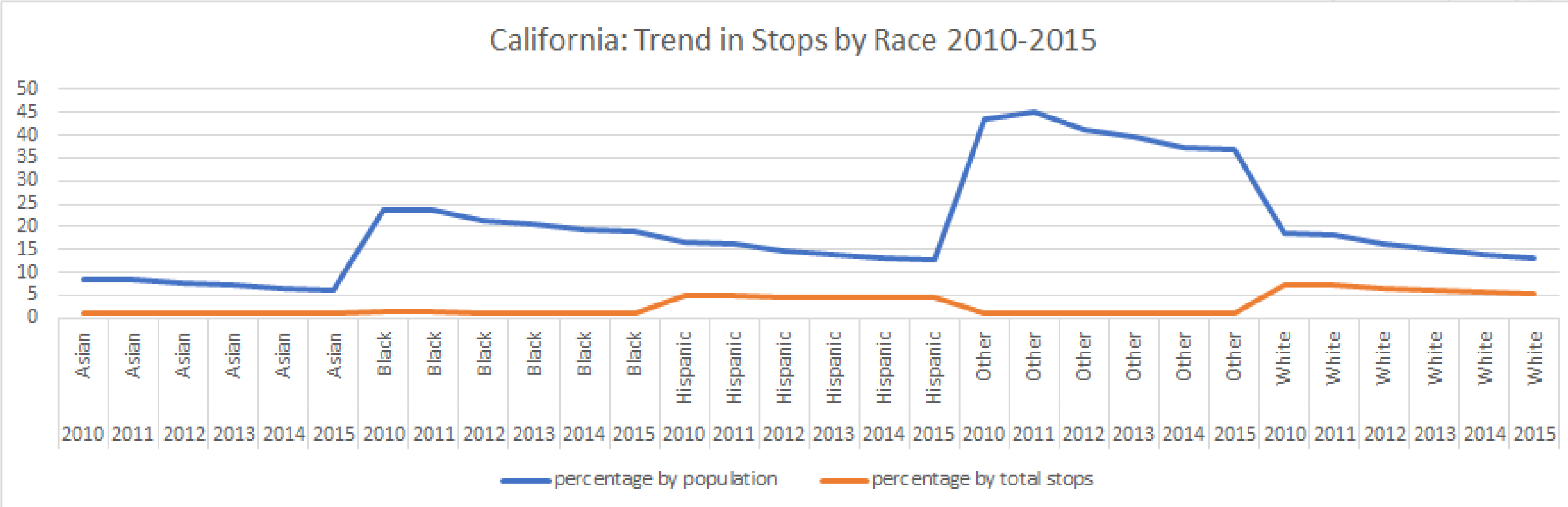



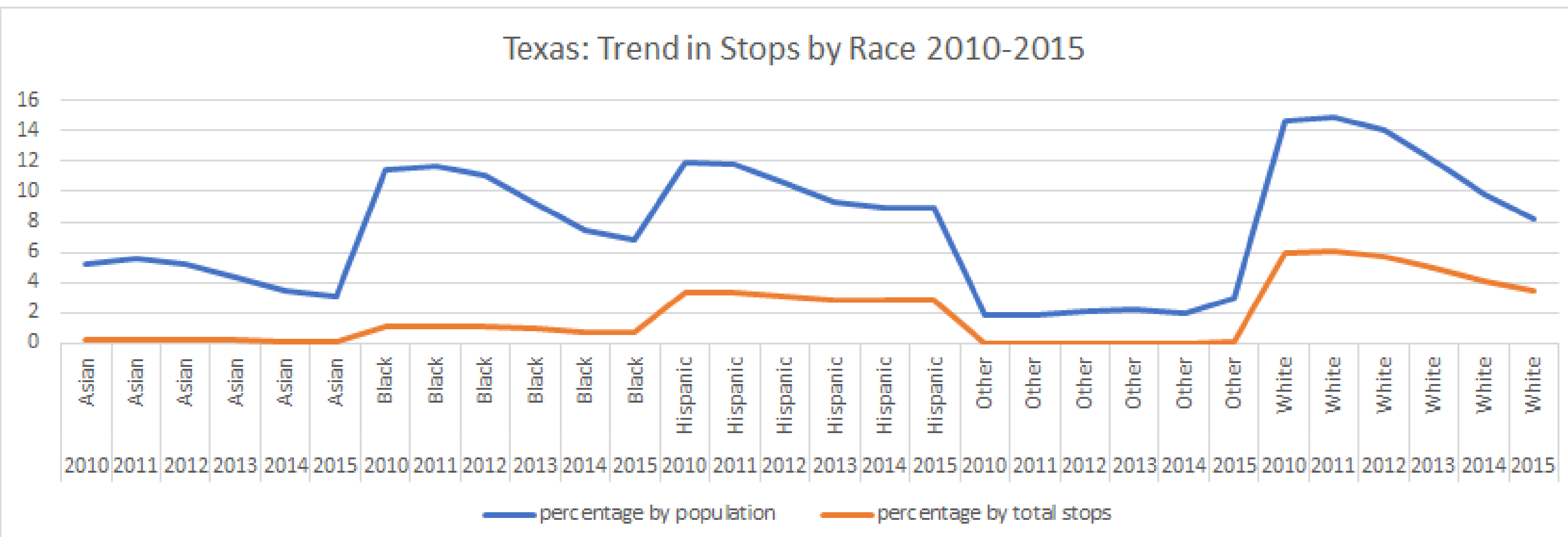
# Number of stops conducted by Race



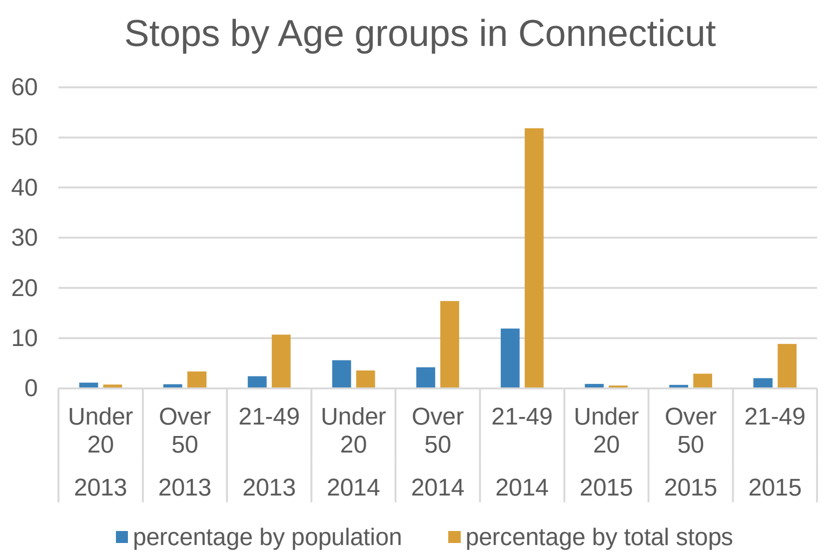



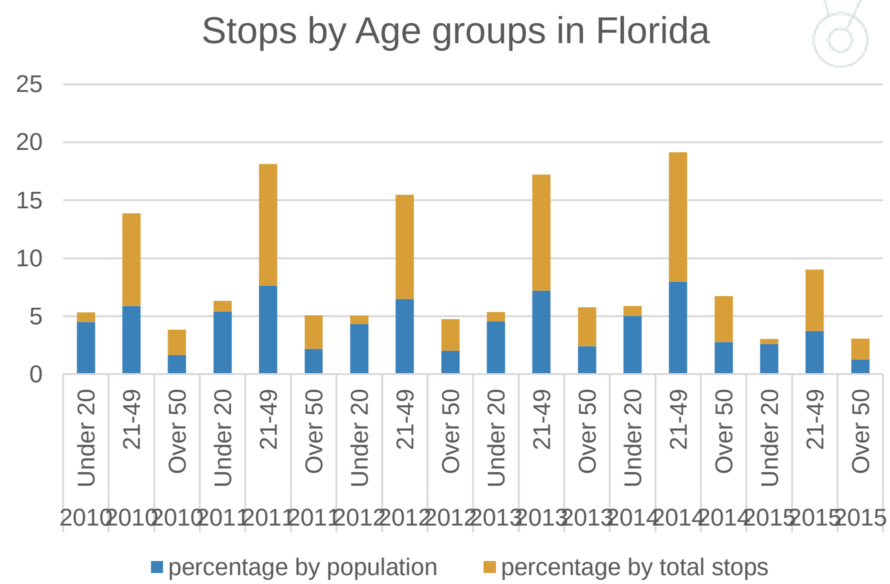
# Number of stops conducted by Race



# Trends in Number of stops conducted by Age group.

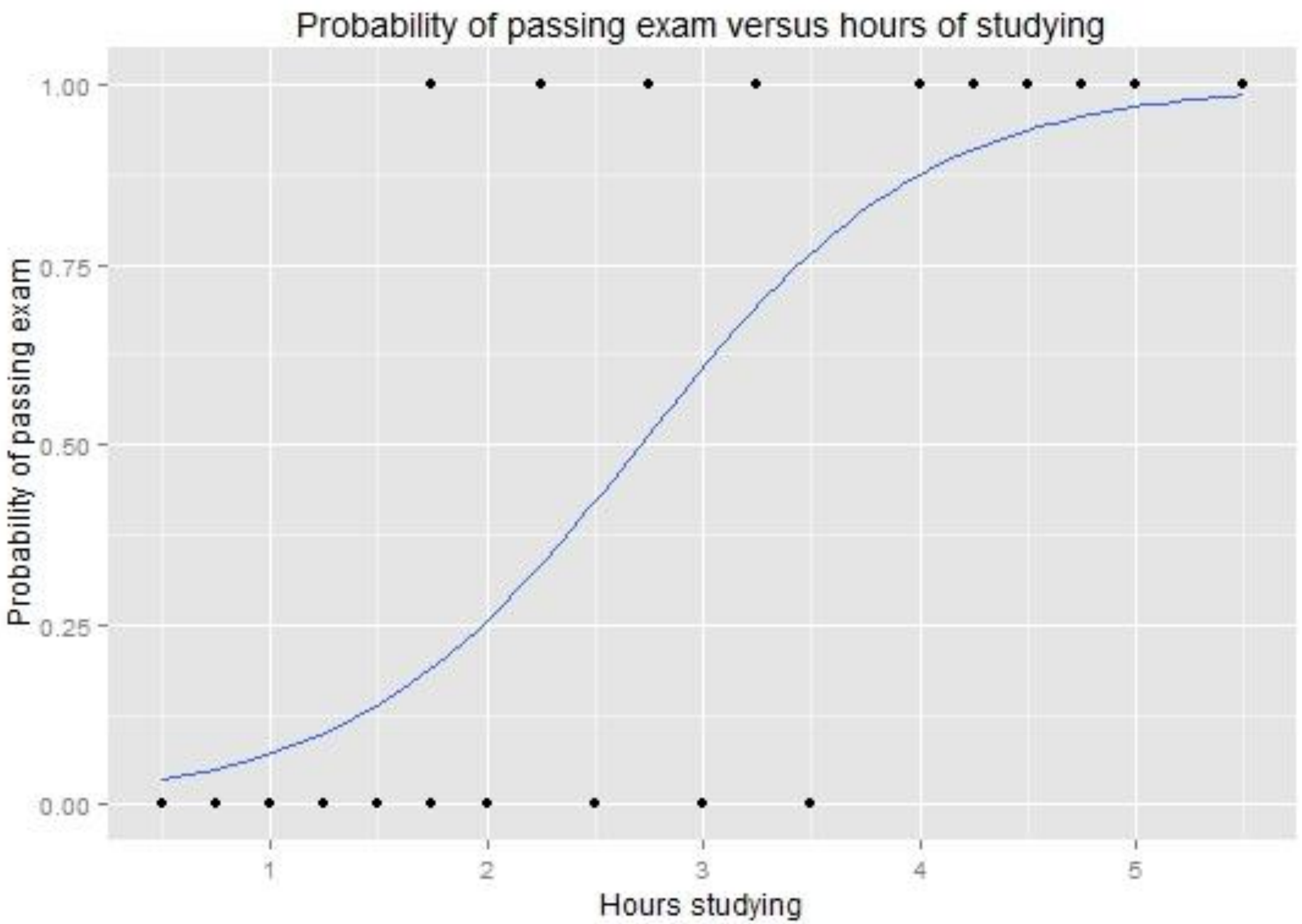


## Logistic Regression Model

- Basics
- Background
- Implementation
- Results

# Logistic Regression Model I



# Logistic Regression Model II

- Predictors: Age, Gender and race.
- Labels:
  - Arrested? True or false
  - Searched? True or false
  - Contraband Found? True or false

- Challenges:
  - Highly skewed data – Sampling.
  - Data Inconsistency

## Logistic Regression Model III

- Implementation similar to the decision tree.
  - Bucketizer
  - Vector assembler
  - String indexer
  - Pipeline

- Data Inconsistency- Results from 23 States only.
  - Only 11 states have all required data.
  - Others have some of either predictors or one of the labels missing.

# Logistic Regression Model III

```
// Load training data
val training = spark.read.format("libsvm").load("data/mllib/sample_libsvm_data.txt")

val lr = new LogisticRegression()
  .setMaxIter(10)
  .setRegParam(0.3)
  .setElasticNetParam(0.8)

// Fit the model
val lrModel = lr.fit(training)

// Print the coefficients and intercept for logistic regression
println(s"Coefficients: ${lrModel.coefficients} Intercept: ${lrModel.intercept}")

// We can also use the multinomial family for binary classification
val mlr = new LogisticRegression()
  .setMaxIter(10)
  .setRegParam(0.3)
  .setElasticNetParam(0.8)
  .setFamily("multinomial")

val mlrModel = mlr.fit(training)

// Print the coefficients and intercepts for logistic regression with multinomial family
println(s"Multinomial coefficients: ${mlrModel.coefficientMatrix}")
println(s"Multinomial intercepts: ${mlrModel.interceptVector}")
```

# Logistic Regression Model IV

```
STATE: FL_cleaned
Search Conducted
Coefficients: [0.0,0.0,0.0] Intercept: -Infinity
CB Found
Coefficients: [0.0,0.0,0.0] Intercept: -Infinity
Arrested
Coefficients: [0.0,0.0,0.0] Intercept: -3.397836071937741
```

```
STATE: NC_cleaned
Search Conducted
Coefficients: [0.0,0.0,0.0] Intercept: -4.921746000872651
CB Found
Coefficients: [0.0,0.0,0.0] Intercept: -6.846470158514971
Arrested
Coefficients: [0.0,0.0,0.0] Intercept: -4.283387394464548
```

```
STATE: CA_cleaned
Search Conducted
Coefficients: 0.0  0.0   Intercept: -6.49133464365397
CB Found
Coefficients: [0.0,0.0] Intercept: -6.904458564073236
Arrested
Coefficients: [0.0,0.0] Intercept: -5.158507305503332
Y=-5.158507305503332
```

```
STATE: CT_cleaned
Search Conducted
Coefficients: [0.0,0.0,0.0] Intercept: -4.066580837275998
CB Found
Coefficients: [0.0,0.0,0.0] Intercept: -5.149841767414899
Arrested
Coefficients: [0.0,0.0,0.0] Intercept: -3.7361211600616446
Y=-3.7361211600616446
```

# Logistic Regression Model IV

## Sampled Model

```
STATE: CT_cleaned_biased
Search Conducted
Coefficients: 0.0  0.0  0.0   Intercept: Infinity
CB Found
Coefficients: [0.0,0.0,0.0] Intercept: -0.6530401389639741
Arrested
Coefficients: [0.0,0.0,0.0] Intercept: -0.9617135042224441
```

## Original

```
STATE: CT_cleaned
Search Conducted
Coefficients: [0.0,0.0,0.0] Intercept: -4.066580837275998
CB Found
Coefficients: [0.0,0.0,0.0] Intercept: -5.149841767414899
Arrested
Coefficients: [0.0,0.0,0.0] Intercept: -3.7361211600616446
Y=-3.7361211600616446
```

# Closing Words

# Thank You

## Q&A