

CSC 3220 FINAL PRESENTATION

# Predicting Income Class Using Census Data

Produced By: Isaiah Chastain, Dylan Smith, Ryan Naleway, Derek Nelson, and Lucas Dowlen



# Project Roadmap

- 01 - Receiving our data
- 02 - Clean our data
- 03 - Explore the data and different prediction/classification models
- 04 - Compare these models
- 05 - Choose a model that performs the best for our desired metrics
- 06 - Export the best model and integrate it into the back-end of the app

## End Goal

The end goal of our project was to develop a model that could predict someone's income class based off of the information provided in a census. If this was possible, we would build an app integrating this model in a UI.

## Receiving and Cleaning our data

# 1994 US Census Data

We got our data from the UCI Machine Learning repository. This dataset has 15 columns: age, workclass, fnlwgt, education, education-num, marital-status, occupation, relationship, race, sex, capital-gain, capital-loss, hours-per-week, native-country, and class.

## NA's:

Our data contained NA values, however these were formatted as “?” values instead of NA values. Our first step was replacing the “?” with NA, and then removing the rows containing said NA values.

## Column Names:

Our original .data file containing our data had the column names as illegible values (X39, X77516, X13, etc.). We realized that this had taken the values of the top row and made it a column name. We duplicated the row and changed these column names to be legible.

## DIRTY DATA

	X39	State.gov	X77516	Bachelors	X13	Never.married	Adm.c
7	52	Self-emp-not-inc	209642	HS-grad	9	Married-civ-spouse	Exec-r
8	31	Private	45781	Masters	14	Never-married	Prof-s
9	42	Private	159449	Bachelors	13	Married-civ-spouse	Exec-r
10	37	Private	280464	Some-college	10	Married-civ-spouse	Exec-r
11	30	State-gov	141297	Bachelors	13	Married-civ-spouse	Prof-s
14	40	Private	121772	Assoc-voc	11	Married-civ-spouse	Craft-r
19	43	Self-emp-not-inc	292175	Masters	14	Divorced	Exec-r
20	40	Private	193524	Doctorate	16	Married-civ-spouse	Prof-s
25	56	Local-gov	216851	Bachelors	13	Married-civ-spouse	Tech-s
27	54	?	180211	Some-college	10	Married-civ-spouse	?
38	31	Private	84154	Some-college	10	Married-civ-spouse	Sales
45	57	Federal-gov	337895	Bachelors	13	Married-civ-spouse	Prof-s
52	47	Private	51835	Prof-school	15	Married-civ-spouse	Prof-s
?	50	Federal-gov	245487	Assoc-acdm	12	Divorced	?

## CLEAN DATA

age	workclass	fnlwgt	education	education-num	marital-status	occ
50	Self-emp-not-inc	83311	Bachelors		13	Married-civ-spouse
38	Private	215646	HS-grad		9	Divorced
53	Private	234721	11th		7	Married-civ-spouse
28	Private	338409	Bachelors		13	Married-civ-spouse
37	Private	284582	Masters		14	Married-civ-spouse
49	Private	160187	9th		5	Married-spouse-absent
52	Self-emp-not-inc	209642	HS-grad		9	Married-civ-spouse
31	Private	45781	Masters		14	Never-married
42	Private	159449	Bachelors		13	Married-civ-spouse
37	Private	280464	Some-college		10	Married-civ-spouse
30	State-gov	141297	Bachelors		13	Married-civ-spouse
23	Private	122272	Bachelors		13	Never-married
32	Private	205019	Assoc-acdm		12	Never-married
34	Private	245487	7th-8th		4	Married-civ-spouse

# Exploring the Data

## Correlations

We decided we would find correlations to find the direction and strength of a variable's influence on another variable.

## Correlation Coefficients

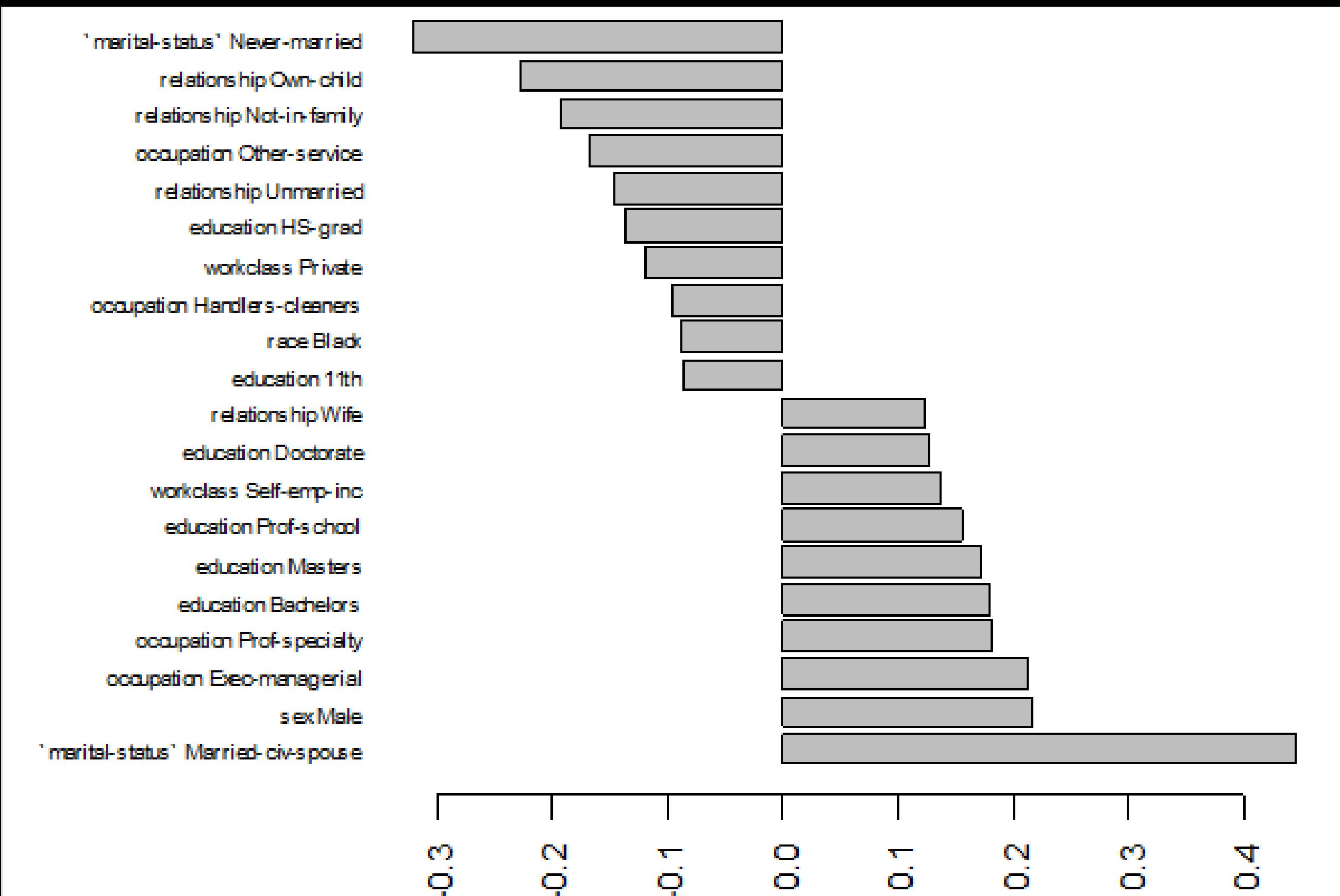
We analyzed the correlation coefficients between all other unique variables and likelihood of the income class being ">\$50K".

## Top Gain Coefficients (Descending Order)

Married-civ-spouse, Male, Exec-managerial, Prof-specialty, Bachelors, Master, Prof-school, Self-emp-inc, Doctorate, Wife

## Top Loss Coefficients (Ascending Order)

Never-married, Own-child, Not-in-family, Other-service, Unmarried, HS-grad, Private, Handlers-cleaners, Black, 11th



## Model Exploration

### Naive Bayes:

Why we chose not to use for our app: Fast, but inaccurate compared to other models

### Random Forest:

Why we chose not to use for our app: Slow, and not as accurate as XGBoost model

### Logistic Model:

Why we chose not to use for our app: Relatively fast, but less accurate compared to Random Forest model

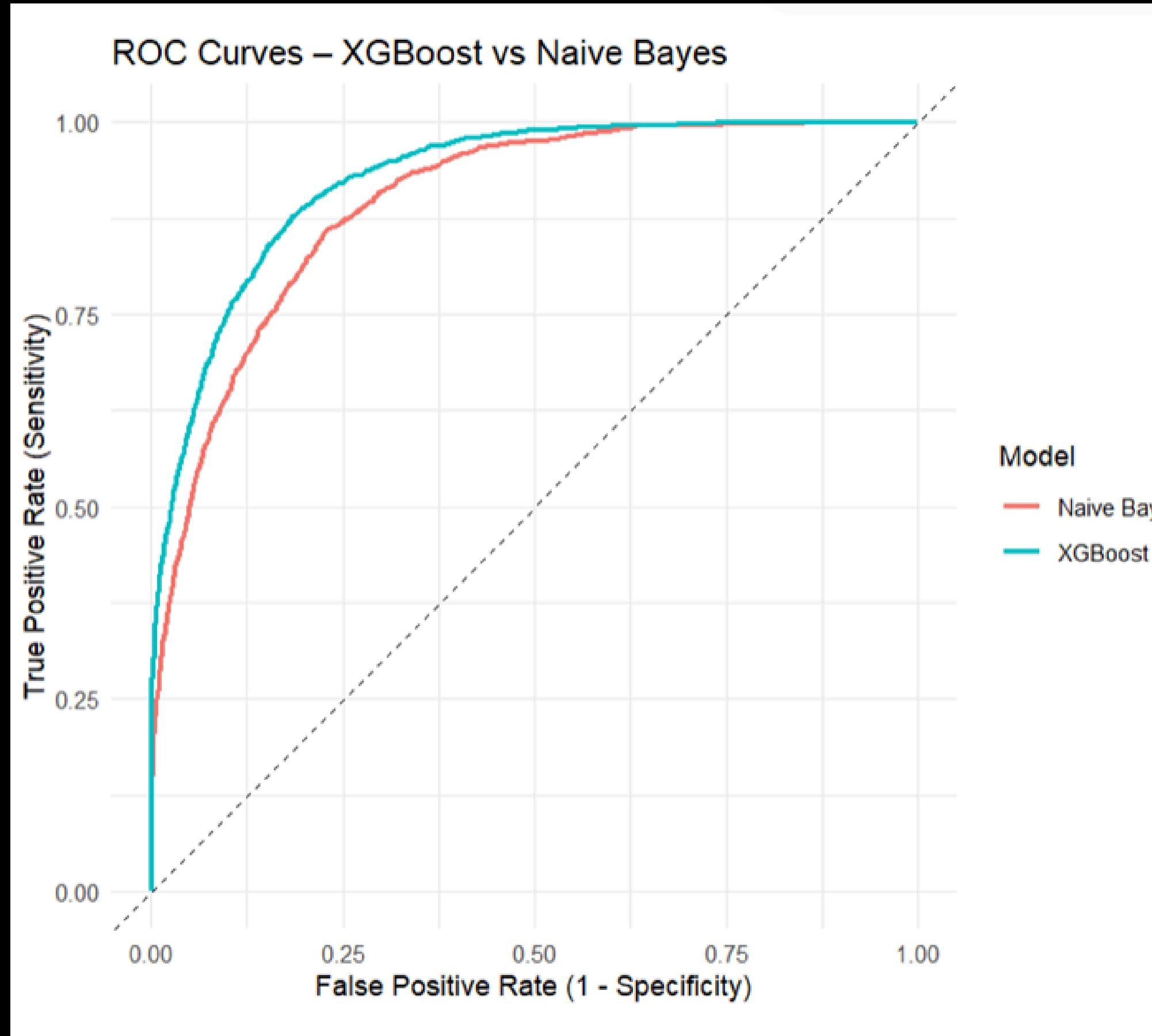
### XGBoost:

Why we chose to use for our app: Only slower than Naive Bayes, and higher scores on all metrics than other models

Model	Accuracy	F1	ROC AUC
Naive Bayes	0.770	0.867	0.897
Logistic	0.847	0.901	0.907
Random Forest	0.862	0.911	0.901
XGBoost	0.866	0.913	0.927

XGBOOST AND NAIVE BAYES

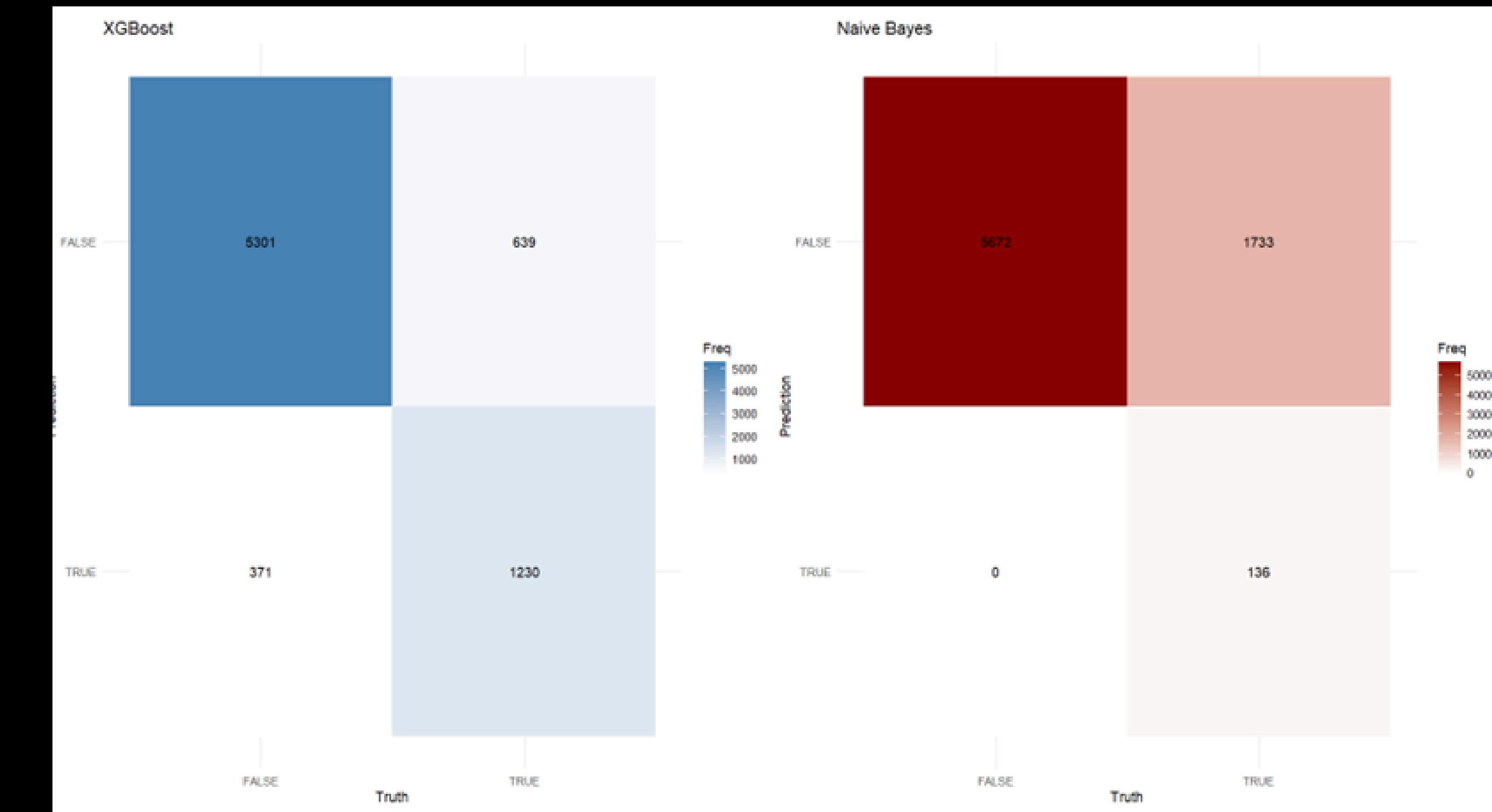
# Confusion Matrix

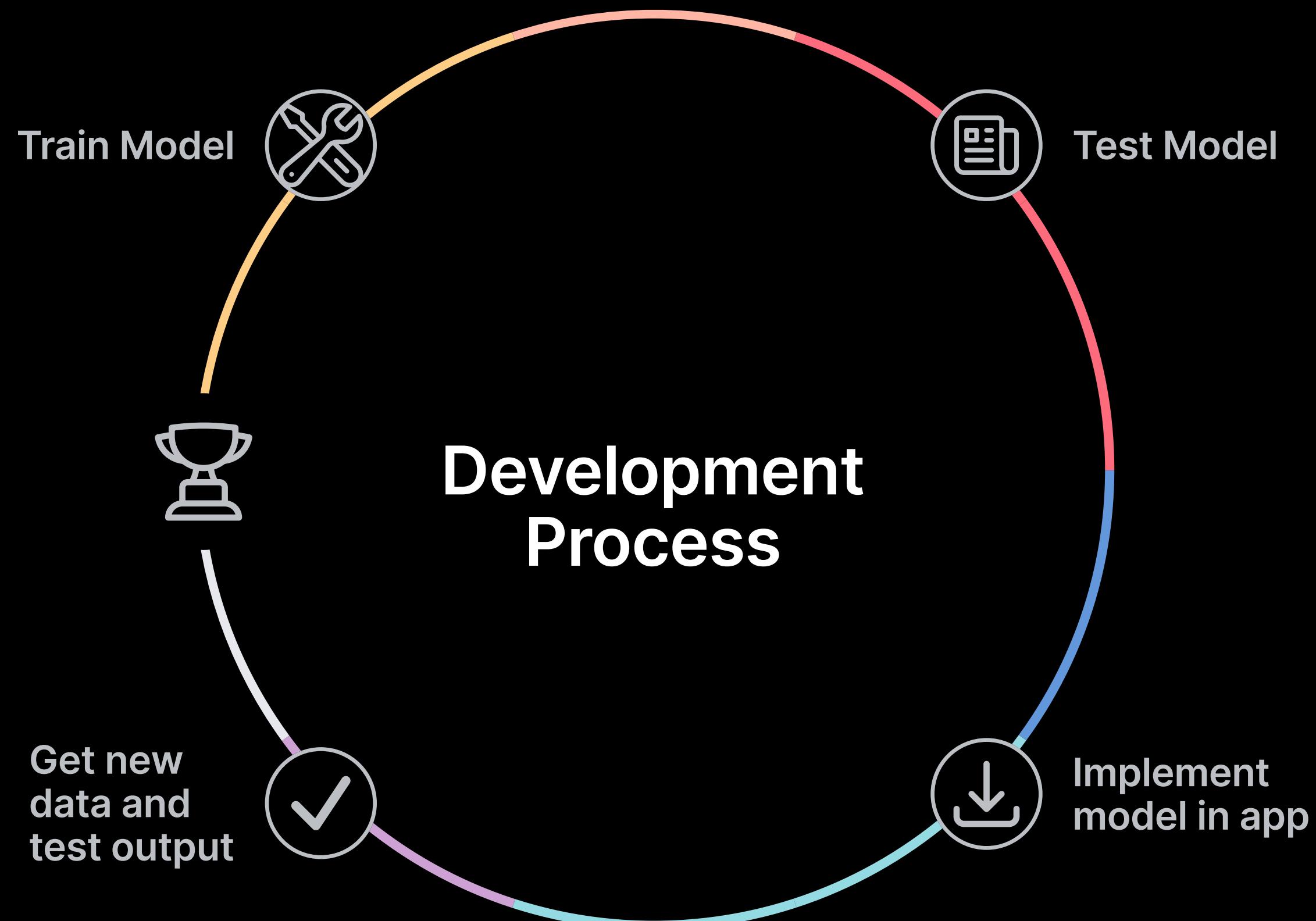


## Overall Pattern

XGBoost effectively identifies true positive and negative cases while maintaining a balanced trade-off between false negatives and false positives.

In this particular implementation, the Naive Bayes model exhibits conservative behavior, showing high specificity (predicted true false very well) but low sensitivity (did not predict true positive very well).





APP DEVELOPMENT

# Creating an Application with our Model - Demonstration

With the data and model we trained, we chose to build a shiny application that we would like to demonstrate for you!