BA 810: Team Project

INCOME PREDICTIONS: A CENSUS SNAPSHOT

Team4: Alima Abdirova, Gunjan Sharma, Oumou Barry, Raiymbek Ordabayev

Agenda

KEY TOPICS DISCUSSED IN THIS PRESENTATION

- Problem Statement
- Data Overview
- EDA
- Modeling
- Results
- Challenges
- Conclusion

Problem Statement

PREDICT INCOME LEVELS (ABOVE/BELOW $50,000)

Using machine learning, considering **demographics** (age, education, marital status) and **occupation** details (job type, industry, employment status).

Importance of the Problem

PREDICTING INCOME LEVELS HAS SIGNIFICANT REAL-WORLD IMPLICATIONS:

1. Policy Making
   - Informs targeted government policies
2. Social Impact
   - Addresses inequality for a fairer society
3. Individual Financial Planning
   - Empowers informed career and financial choices

Data Overview

Source: Extracted from the 1994 Census Bureau Database by Ronny Kohavi and Barry Becker.

Dataset Statistics:

- Records: 32,561
- Columns: 15

Exploring the 1994 Census Bureau Database

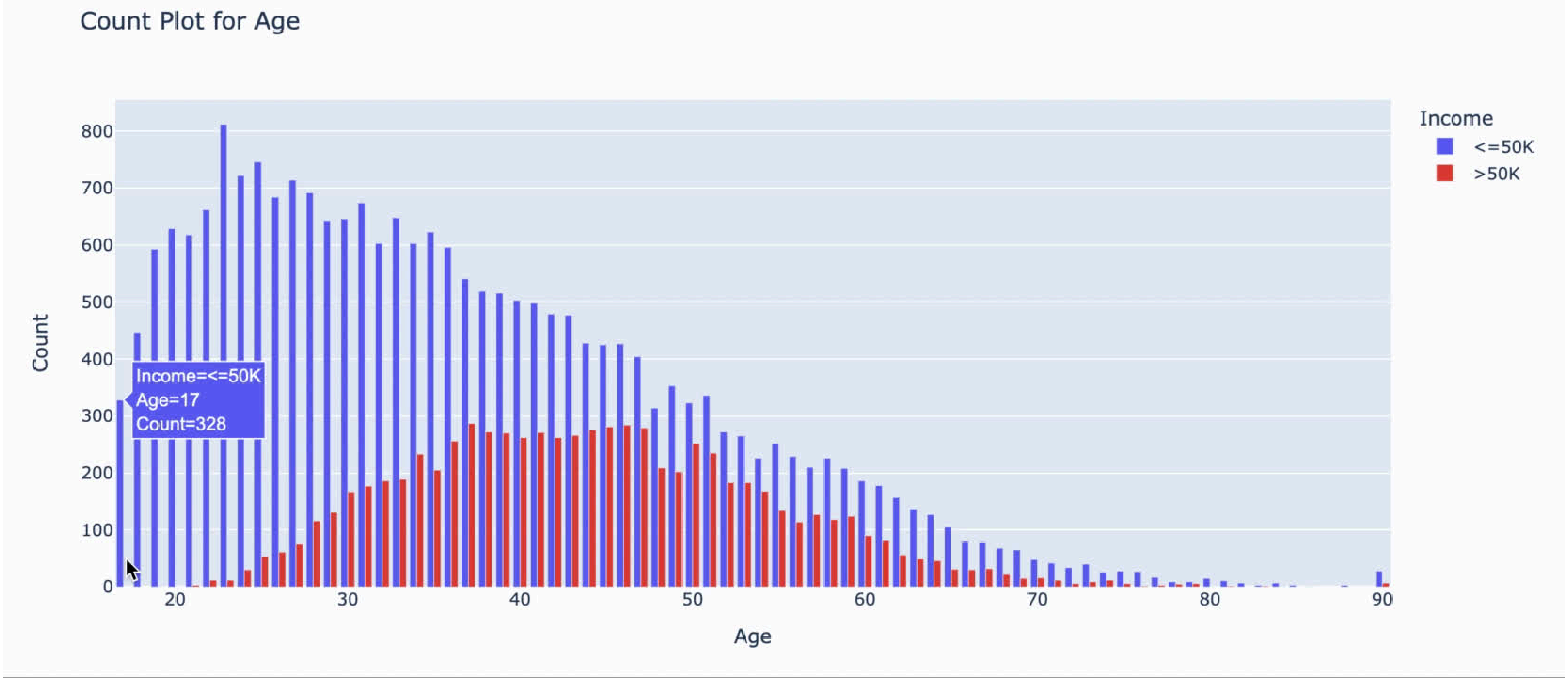
6 Integer columns

9 categorical columns

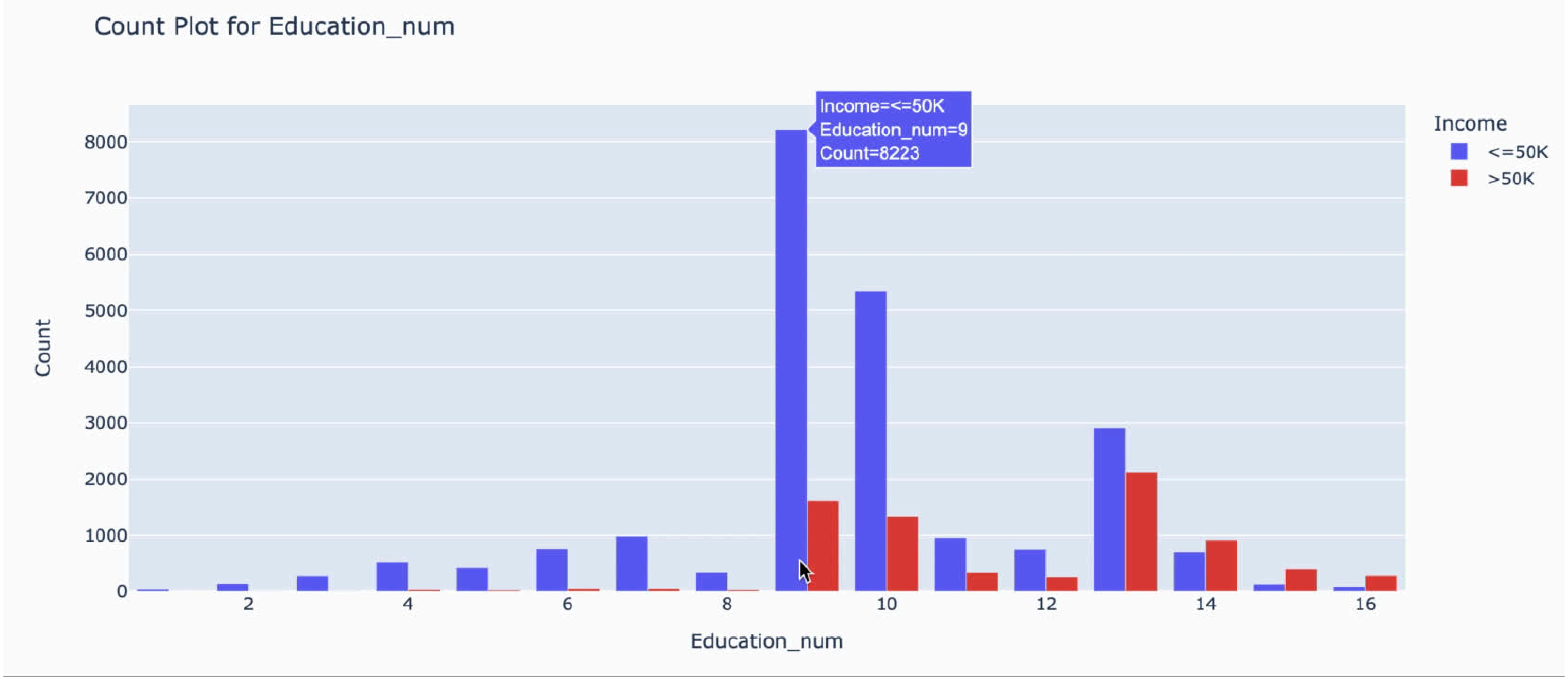
EDA

Income Distribution
Numerical Count Plots
Categorical Count Plots

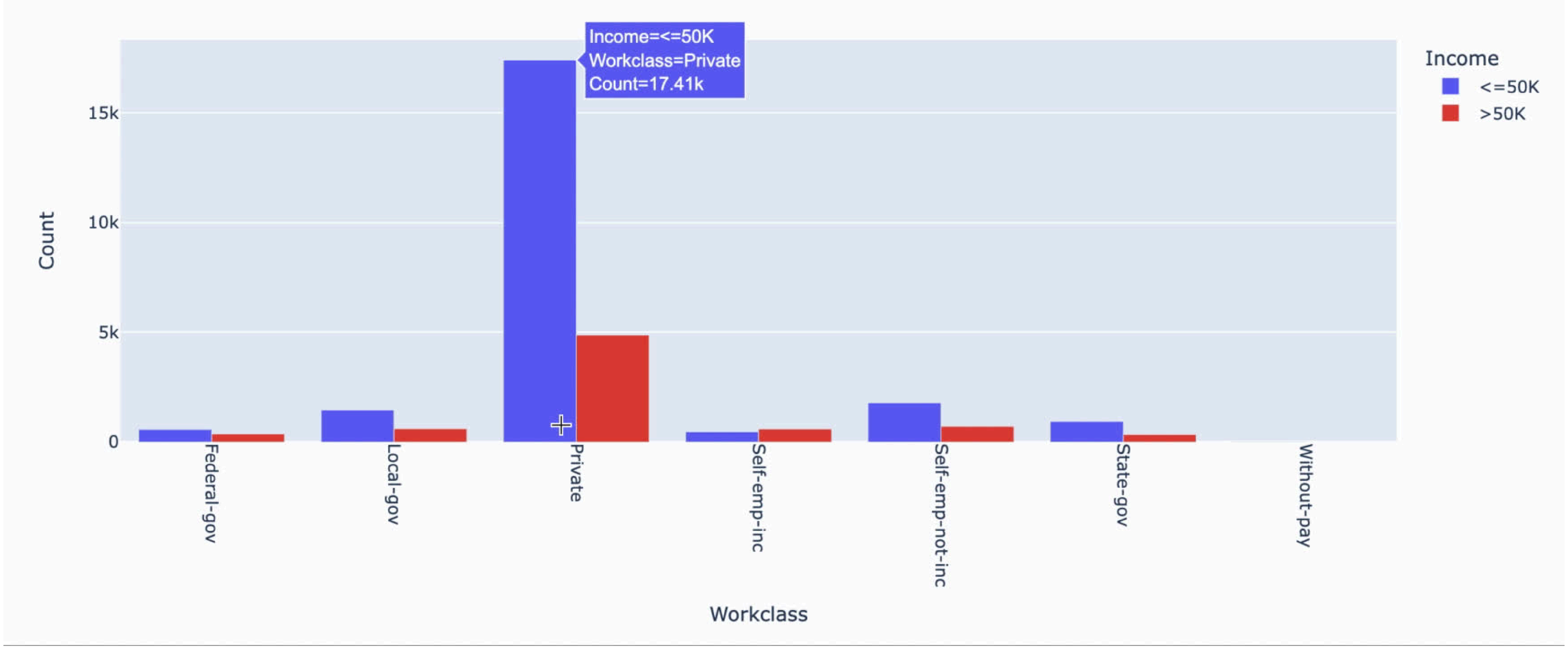
Age VS Income



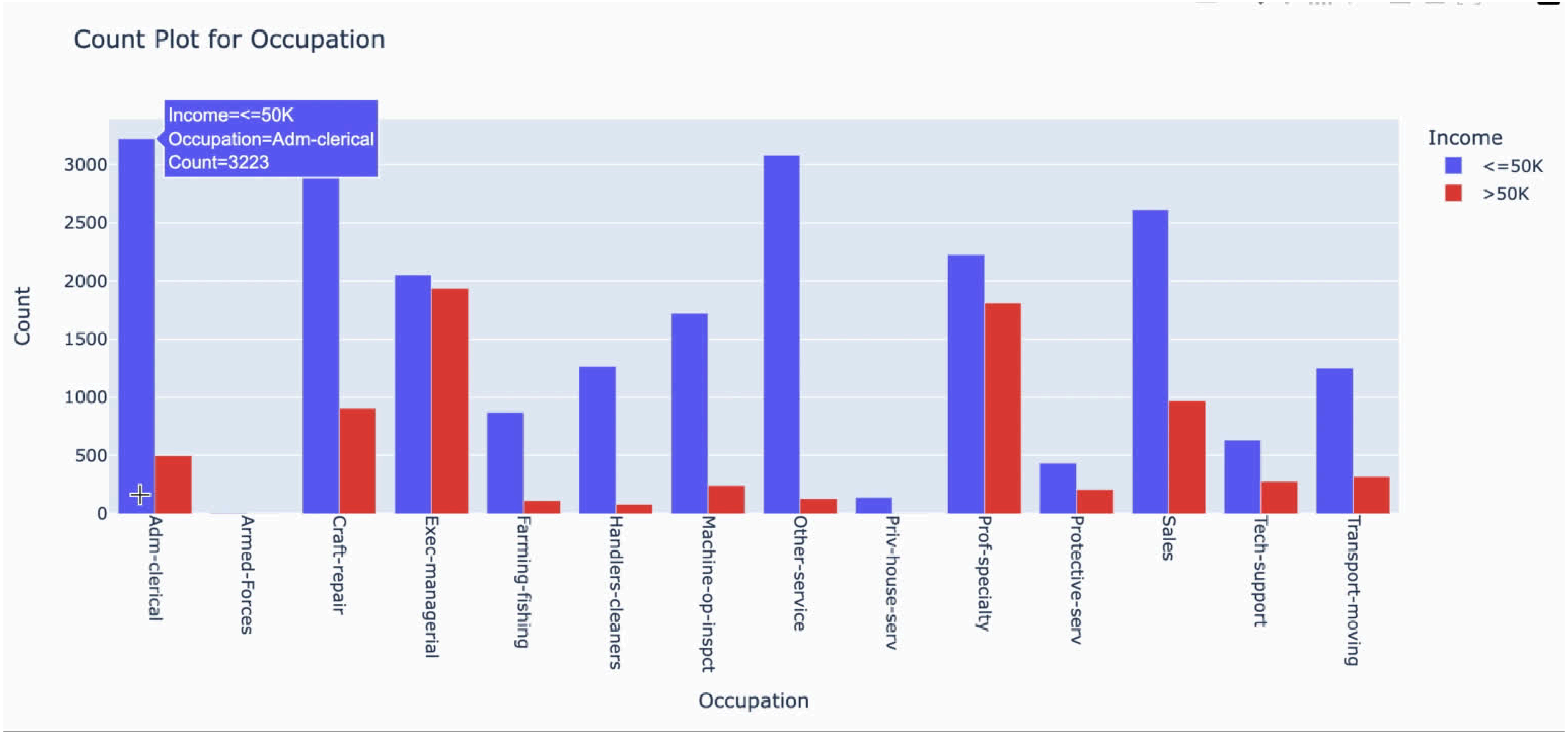
Years of Education VS Income



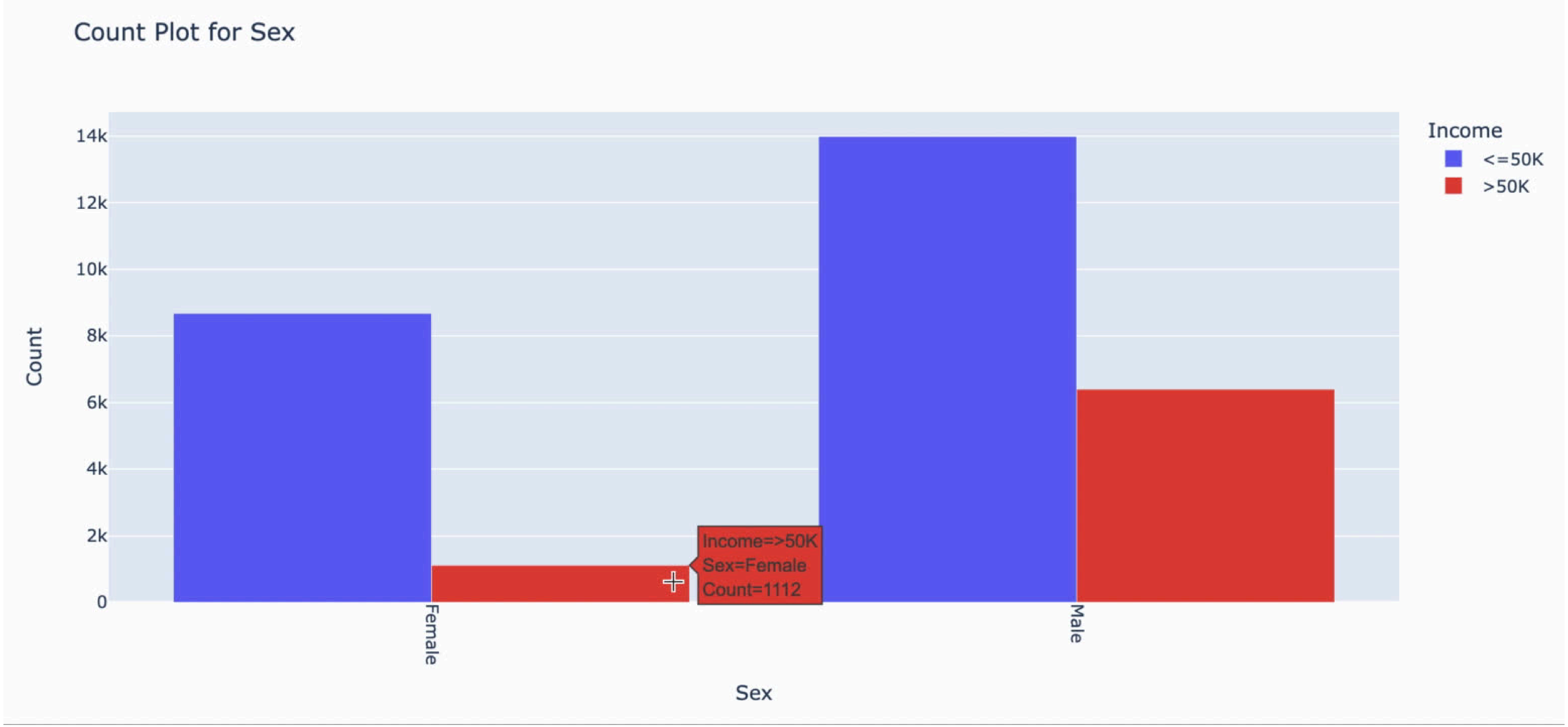
Worklass VS Income



Occupation VS Income



Gender VS Income



Models

PROCESS

Data Preprocessing
Train Test Split
Recursive & Sequential Selection
Grid & Random Search
Balanced Accuracy

LOGISTIC REGRESSION

Best Parameters:
{'C': 2.782559402207126, 'max_iter': 100, 'penalty': 'l2', 'solver': 'liblinear'}
Train Balanced accuracy:
76.71
Test Balanced accuracy:
75.02

RANDOM FOREST

Best Parameters:
{'max_depth': 10, 'min_samples_leaf': 1}
Train Balanced accuracy:
76.32
Test Balanced accuracy:
74.96

KNN

Best Parameters:
{'metric': 'euclidean', 'n_neighbors': 7, 'weights': 'uniform'}
Train Balanced accuracy:
80.87.
Test Balanced accuracy:
75.43

SVC

Best Parameters:
{'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}
Train Balanced accuracy:
78.69
Test Balanced accuracy:
75.37

Voting

Hard Voting or majority vote

Accuracy 84.87 (balanced 73.93)

Higher predicition than all except random forest

Stacking

Accuracy 84.83

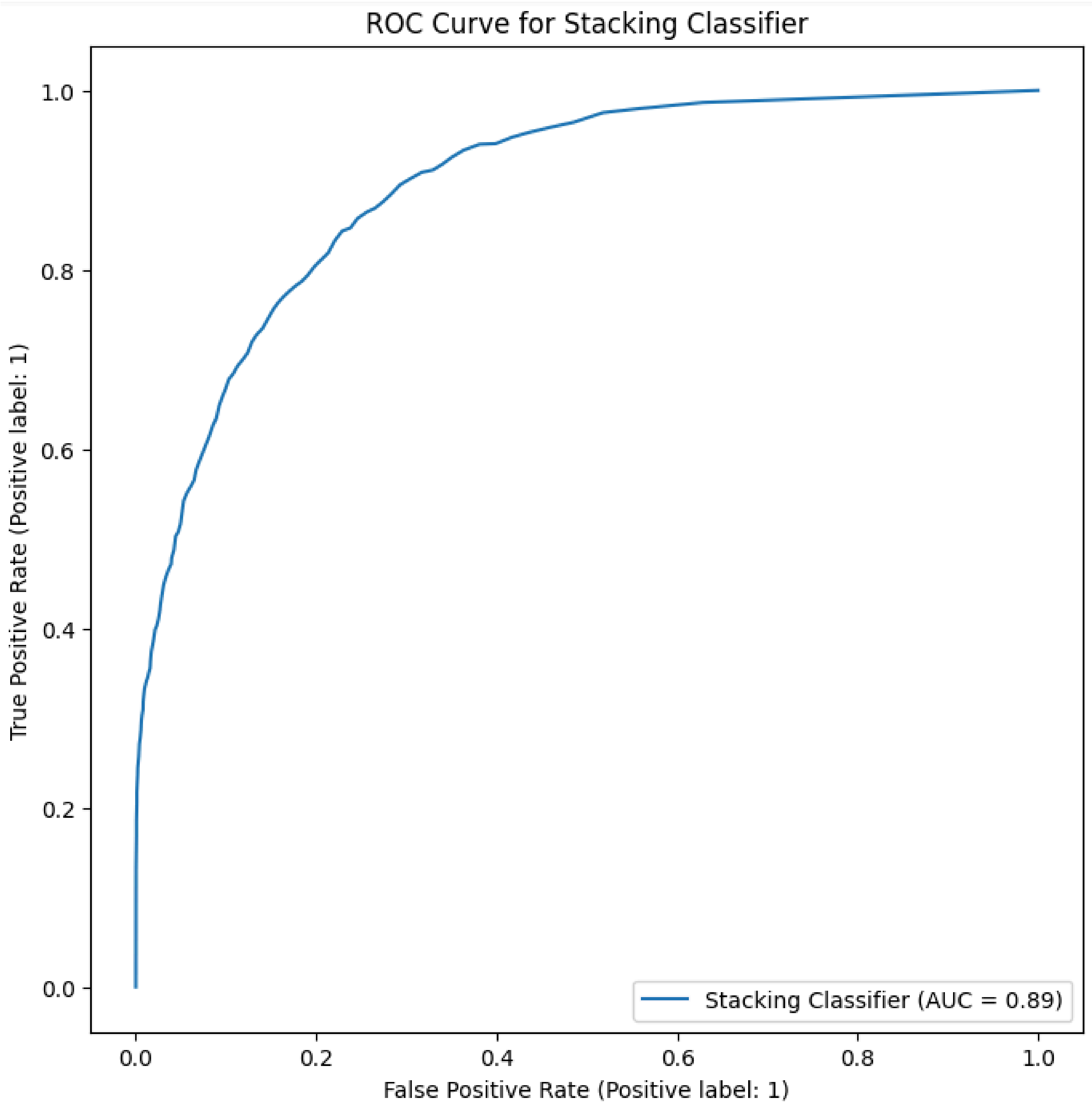
Balanced accuracy 75.12

Higher than all except random forest

CONFUSION MATRIX

True Negatives 4210	False Positives 323
False Negatives 592	True Positives 908

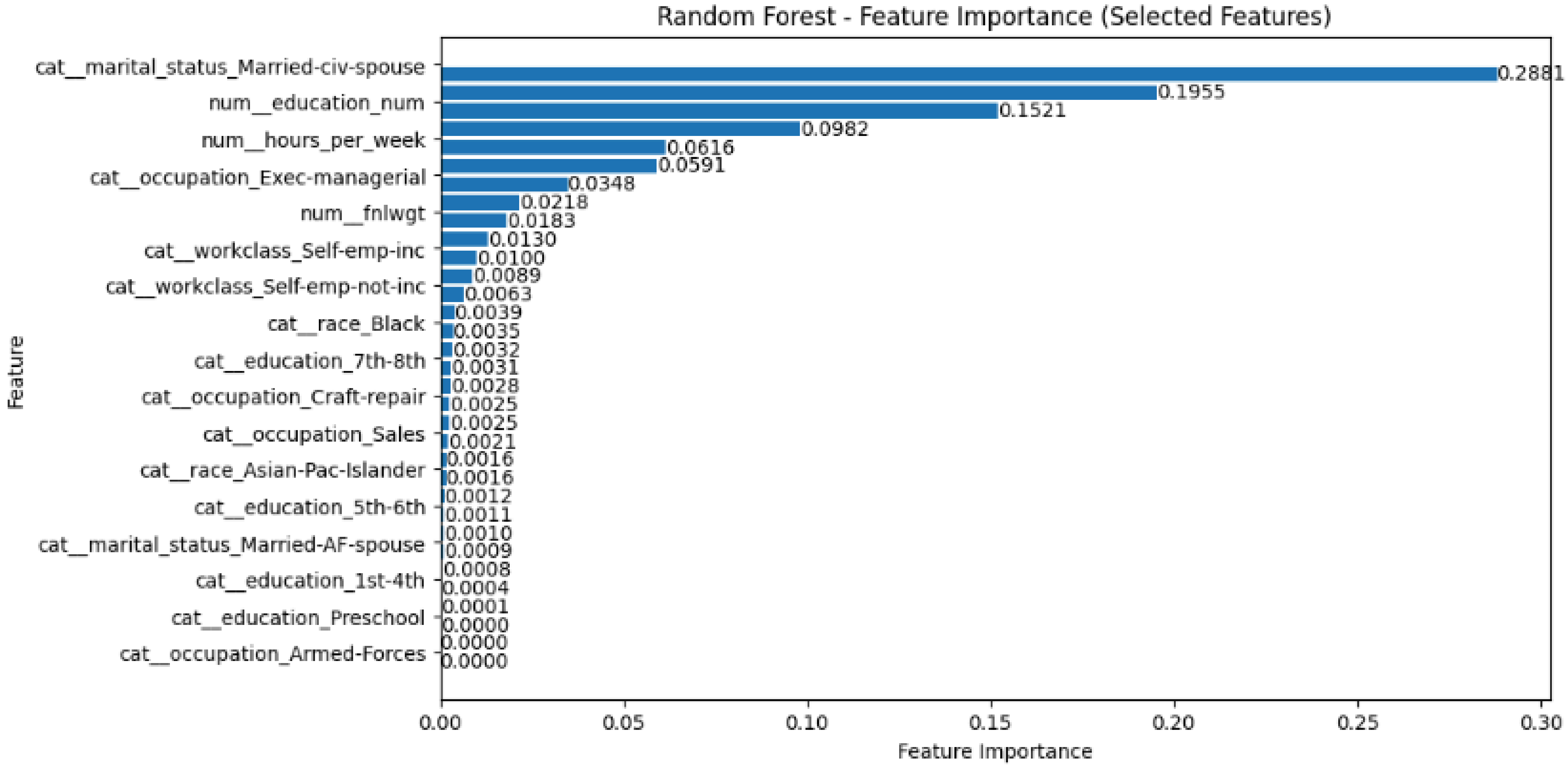
	Precision	Recall	F1-Score	Support
Class 0	0.88	0.93	0.90	4533
Class 1	0.74	0.61	0.66	1500
Accuracy			0.85	6033
Macro Avg	0.81	0.77	0.78	
Weighted Avg	0.84	0.85	0.84	



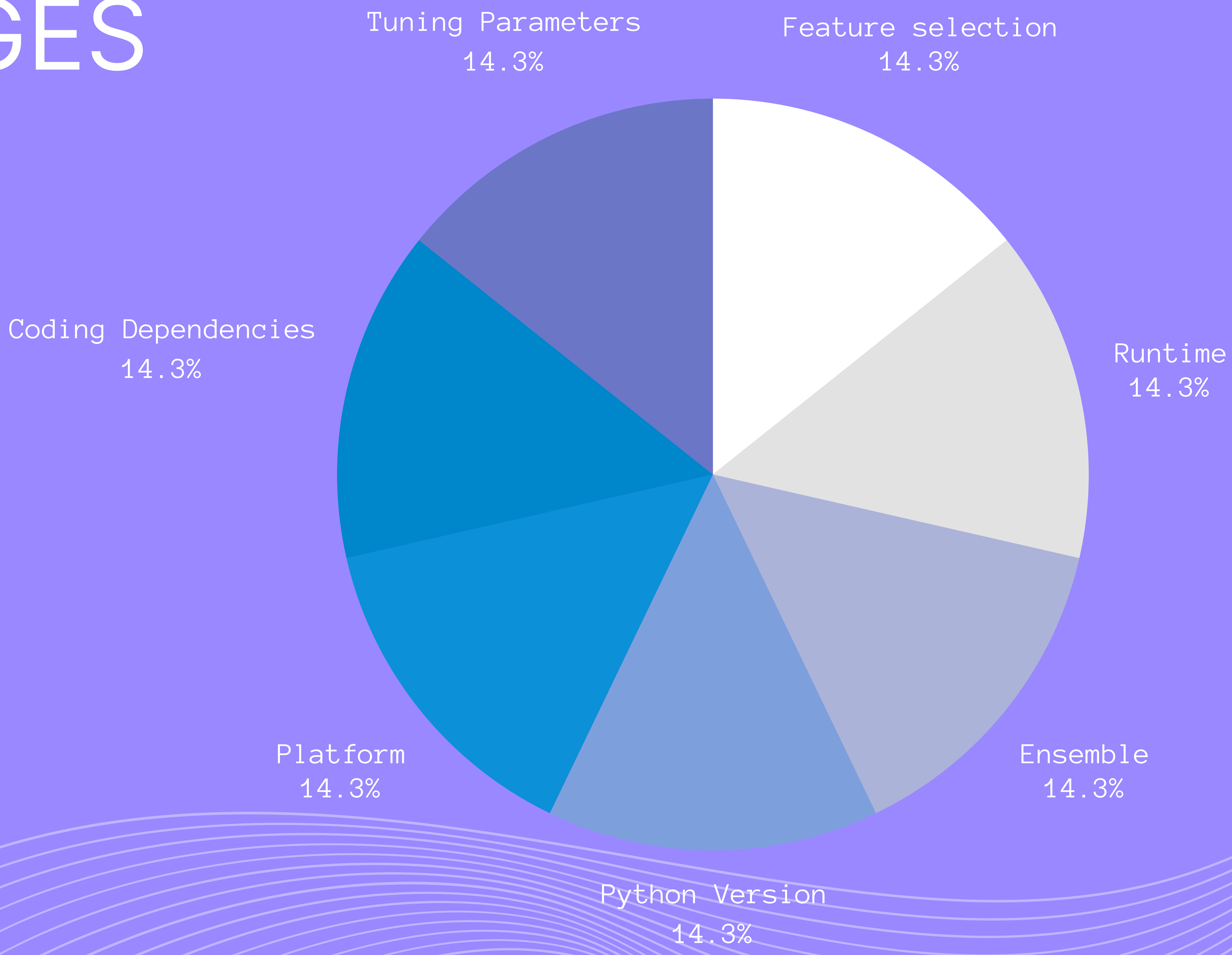
ROC CURVE

- The stacking model has an AUC of 0.89.
- Model with strong discriminative power between positive and negative instances.
- Performs well across various thresholds.

Feature Importance



CHALLENGES


Tuning Parameters
14.3%
Feature selection
14.3%
Coding Dependencies
14.3%
Runtime
14.3%
Platform
14.3%
Ensemble
14.3%
Python Version
14.3%

Runtime Issues

- Addressed prolonged runtimes from large datasets and complex models affecting experimentation and tuning.

Platform Selection - Jupyter vs. Colab

- Chose Google Cloud Console for efficiency over Jupyter and Colab, due to dataset size.

Feature Importance and Selection

- Overcame challenges in discerning key features from 60+ variables using correlation and importance ranking.

Managing Coding Dependencies

- Ensured library compatibility and consistent package installation in a collaborative environment.

Ensemble Model Voting Decisions

- Resolved the dilemma of choosing voting strategies in ensemble models to improve accuracy and effectiveness.

Hyperparameter Tuning Complexity

- Utilized various tuning methods to optimize model performance, balancing computational load.

Python Version Compatibility

- Managed Python version compatibility issues to ensure smooth code execution.

THANK YOU

Link to the Colab Notebook:

https://colab.research.google.com/drive/1ZWZtRzKUGgH236VqpcKboYUOQ1jLzBpT?usp=sharing