Income Classification Model

Introduction

The income dataset used in this project was extracted from the 1994 U.S. Census database. The dataset contains demographic and socioeconomic information about individuals, which is used to predict whether a person earns more than $50,000 annually.

The Importance of Census Statistics

The census is a comprehensive activity conducted once every decade to gather detailed information about the population. It provides insights into demographic, social, and economic characteristics, including age, gender, education, employment, and housing conditions. This data is crucial for planning public services, improving quality of life, and addressing societal issues. It also enables citizens to evaluate government decisions and ensure they align with public needs.

Objective of the Project

The primary goal of this project is to build a machine learning model that predicts whether an individual earns more than $50,000 annually based on their demographic and socioeconomic attributes. Various classification techniques were explored, with the Random Forest model yielding the best results.

Data Analysis Process

Fetching Data

Import Packages and Data

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import sklearn
%matplotlib inline

income_df = pd.read_csv("income_evaluation.csv")
income_df.head()
income_df.describe()
```

Data Dictionary

1. Categorical Attributes

- **workclass:** Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
- **education:** Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
- **marital-status:** Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
- **occupation:** Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
- **relationship:** Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
- **race:** White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
- **sex:** Female, Male.
- **native-country:** United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

2. Continuous Attributes

- **age:** Age of an individual.
- **education-num:** Number of years of education.
- **fnlwgt:** Final weight (population representation).
- **capital-gain:** Capital gains.
- **capital-loss:** Capital losses.
- **hours-per-week:** Hours worked per week.

Data Cleaning

Dealing with Missing Values

- Missing values were identified in workclass, occupation, and native-country fields.
- Records with missing values were dropped to ensure data quality.

```
income_df.isnull().sum()
income_df.age = income_df.age.astype(float)
income_df['hours-per-week'] = income_df['hours-per-week'].astype(float)
my_df = income_df.dropna()
my_df['predclass'] = my_df['income']
del my_df['income']
my_df['education-num'] = my_df['educational-num']
del my_df['educational-num']
```

```
my_df.info()
my_df.isnull().sum()
```

Feature Engineering

Predclass

Target variable encoding:

- 1 for income >50K
- 0 for income <=50K

Education

Grouped into categories:

- Dropout: Preschool, 1st-4th, 5th-6th, 7th-8th, 9th, 10th, 11th, 12th
- HighGrad: HS-grad, HS-Grad
- CommunityCollege: Some-college, Assoc-acdm, Assoc-voc
- Bachelors: Bachelors
- Masters: Masters, Prof-school
- Doctorate: Doctorate

Marital Status

Marital status was simplified into:

- Married: Married-civ-spouse, Married-AF-spouse.
- NotMarried: Never-married, Married-spouse-absent.
- Separated: Separated, Divorced.
- Widowed: Widowed.

Occupation

Occupation categories were analyzed, with manual labor jobs being the most common.

Age

Age was discretized into bins for better analysis.

Race

Race was analyzed in relation to income levels.

Hours of Work

Hours worked per week were binned into 10 categories.

Crossing Feature: Age + Hours of Work

A new feature age-hours was created by multiplying age and hours-per-week.

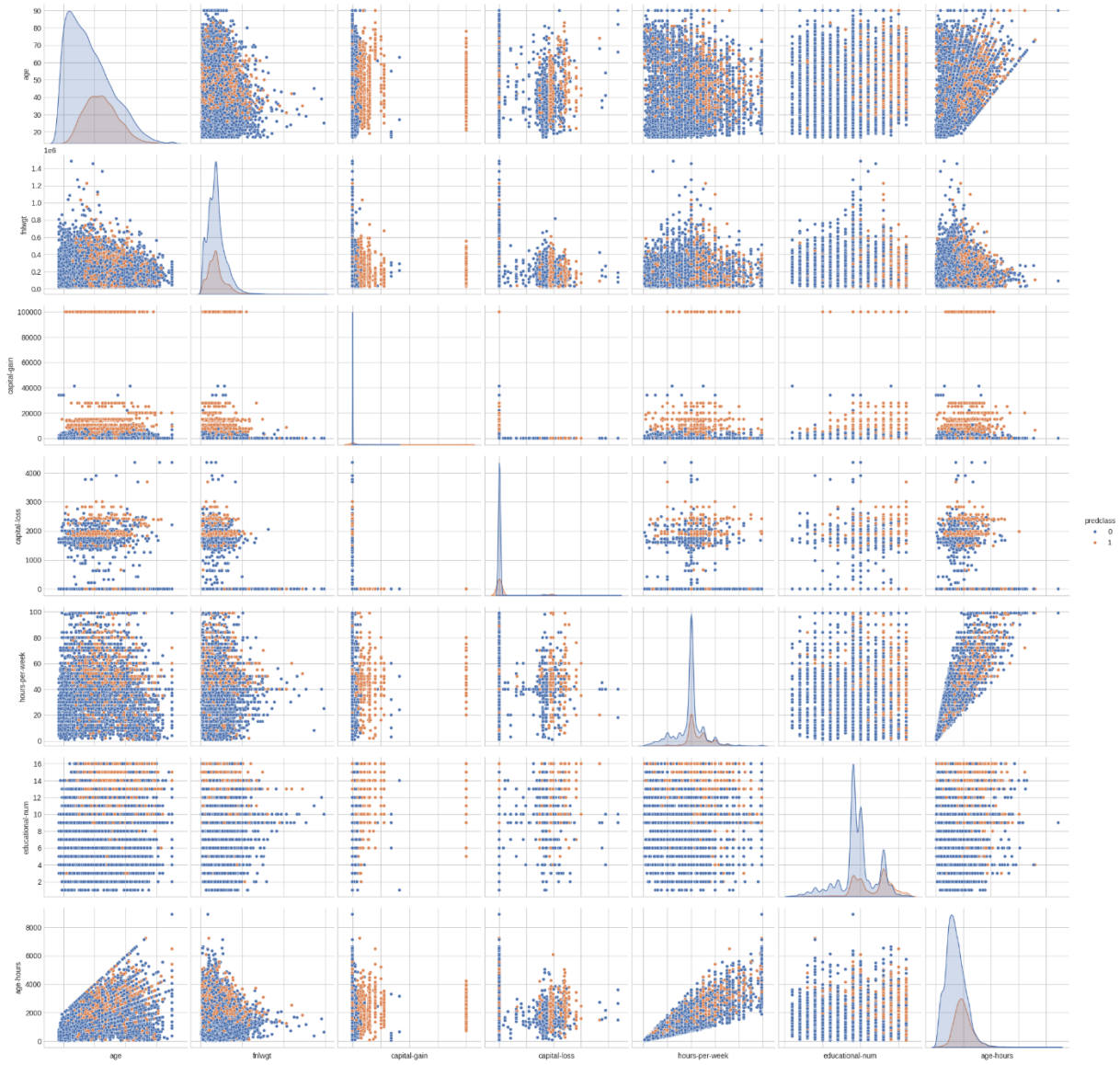
Exploratory Data Analysis (EDA)

Pair Plot

Pair plots were used to visualize relationships between features.

```
pp = sns.pairplot(my_df, hue='predclass', palette='deep', size=3,
diag_kind='kde', diag_kws=dict(shade=True), plot_kws=dict(s=20))

pp.set(xticklabels=[])
```



Correlation Heatmap

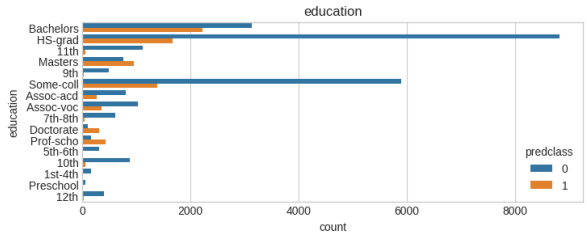
A heatmap was used to identify correlations between features.

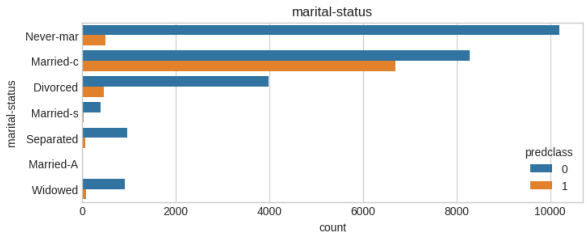
`correlation_heatmap(my_df)`

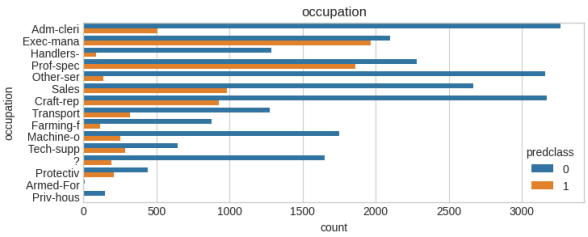
Bivariate Analysis

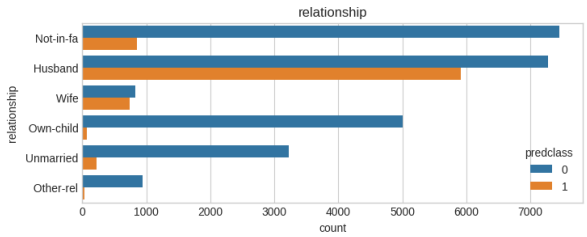
Bivariate analysis was performed to understand the relationship between categorical features and income levels.

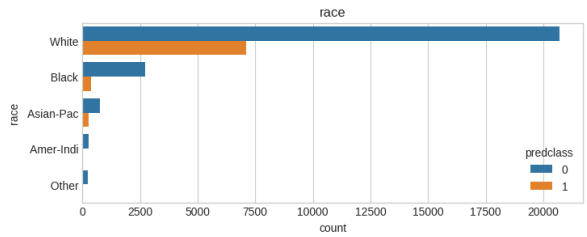
age	workclass	fnlwgt	education	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	predclass	educational-num	age_bin	hours-per-week_bin	age-hours	age-hours_bin
27.0	Private	257302	Assoc-acdm	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38.0	United-States	01	12	(24.3, 27.95]	(30.4, 40.2]	1026.0	(909.9, 1798.8]
40.0	Private	154374	HS-grad	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40.0	United-States	19	9	(38.9, 42.55]	(30.4, 40.2]	1600.0	(909.9, 1798.8]
58.0	Private	151910	HS-grad	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40.0	United-States	09	9	(57.15, 60.8]	(30.4, 40.2]	2320.0	(1798.8, 2687.7]
22.0	Private	201490	HS-grad	Never-married	Adm-clerical	Own-child	White	Male	0	0	20.0	United-States	09	9	(20.65, 24.3]	(10.8, 20.6]	440.0	(12.111, 909.9]
52.0	Self-emp-inc	287927	HS-grad	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40.0	United-States	19	9	(49.85, 53.5]	(30.4, 40.2]	2080.0	(1798.8, 2687.7]

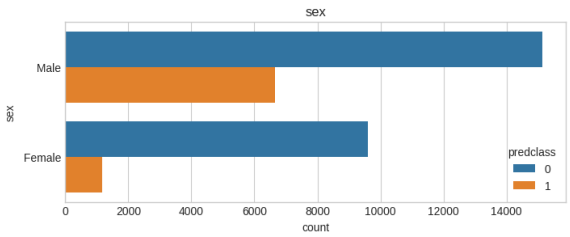


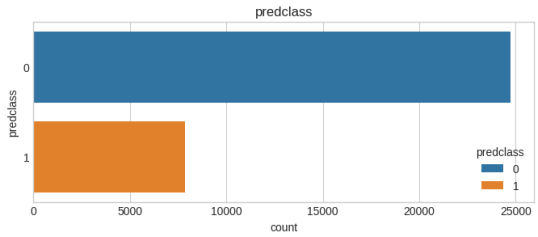






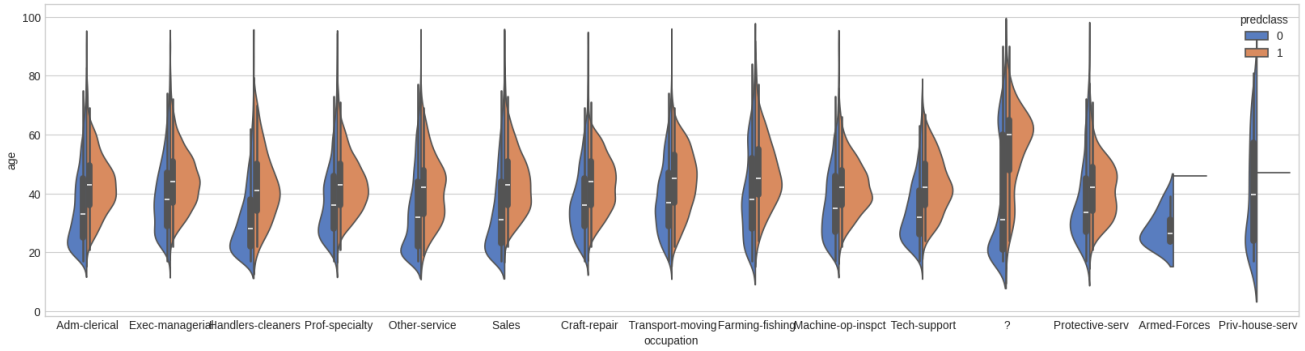






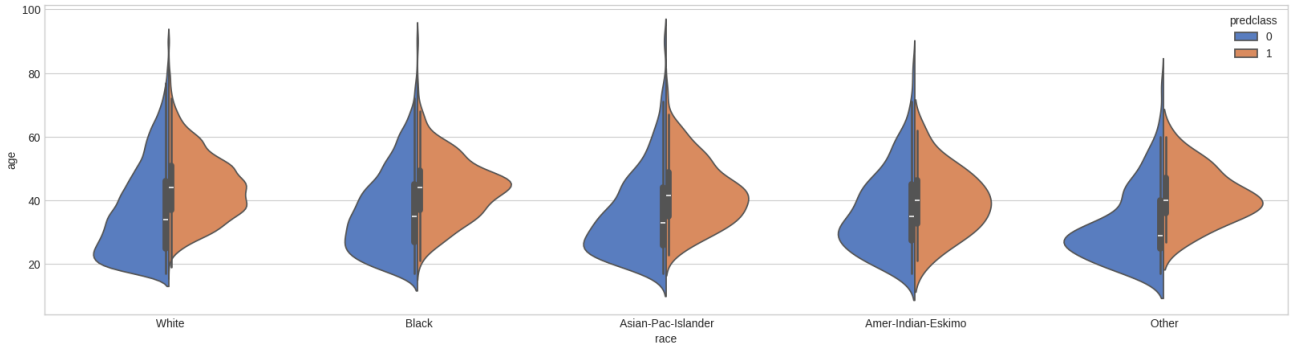
Occupation vs. Income Level

A violin plot was used to analyze the distribution of age across occupations and income levels.



Race vs. Income Level

A similar analysis was performed for race and income levels.



Machine Learning Implementation

Feature Encoding

```
from sklearn.preprocessing import LabelEncoder
my_df = my_df.apply(LabelEncoder().fit_transform)
```

Index	Age	Workclass	Fnlwgt	Education	Marital-Status	Occupation	Relationship	Race	Sex	Capital-Gain	Capital-Loss	Hours-Per-Week	Native-Country	Predclass	Educational-Num	Age_Bin	Hours-Per-Week_Bin	Age-Hours	Age-Hours_Bin
0	22	7	2671	9	4	1	1	4	1	25	0	39	39	0	12	6	3	631	1
1	33	6	2926	9	2	4	0	4	1	0	0	12	39	0	12	9	1	290	0
2	21	4	14086	11	0	6	1	4	1	0	0	39	39	0	8	5	3	620	1
3	36	4	15336	1	2	6	0	2	1	0	0	39	39	0	6	9	3	810	2
4	11	4	19355	9	2	10	5	2	0	0	0	39	5	0	12	3	3	477	1

Train-Test Split

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=2)
```

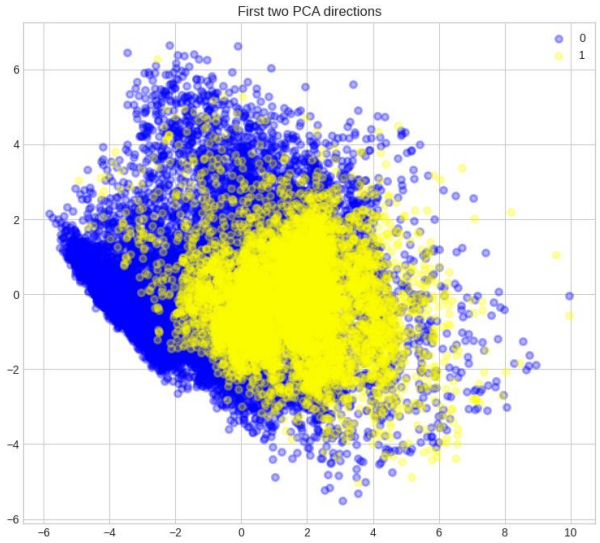
Principal Component Analysis (PCA)

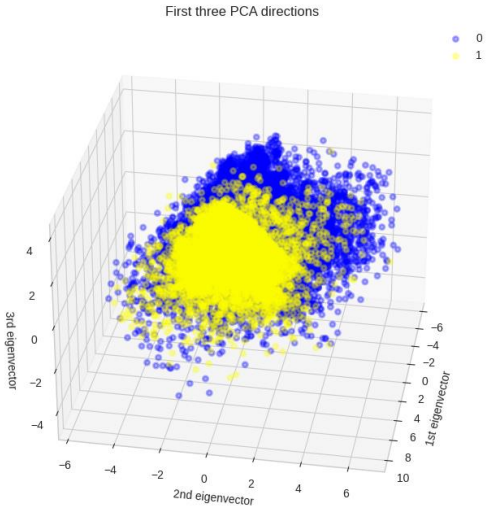
PCA was applied to reduce dimensionality and visualize data in 2D and 3D.

```
from sklearn.decomposition import PCA

pca = PCA(n_components=2)

X_r = pca.fit(X).transform(X)
```





- The first two PCA directions provide a 2D view of the data, showing how well the classes can be separated based on the two most significant components.
- The first three PCA directions extend this to a 3D view, offering a more comprehensive understanding of the data structure.
- The eigenvectors indicate the directions in which the data varies the most, helping to identify the most important features for classification.

This visualization is crucial for understanding the underlying structure of the data and can guide further analysis, such as feature selection or model building.

Classification Models

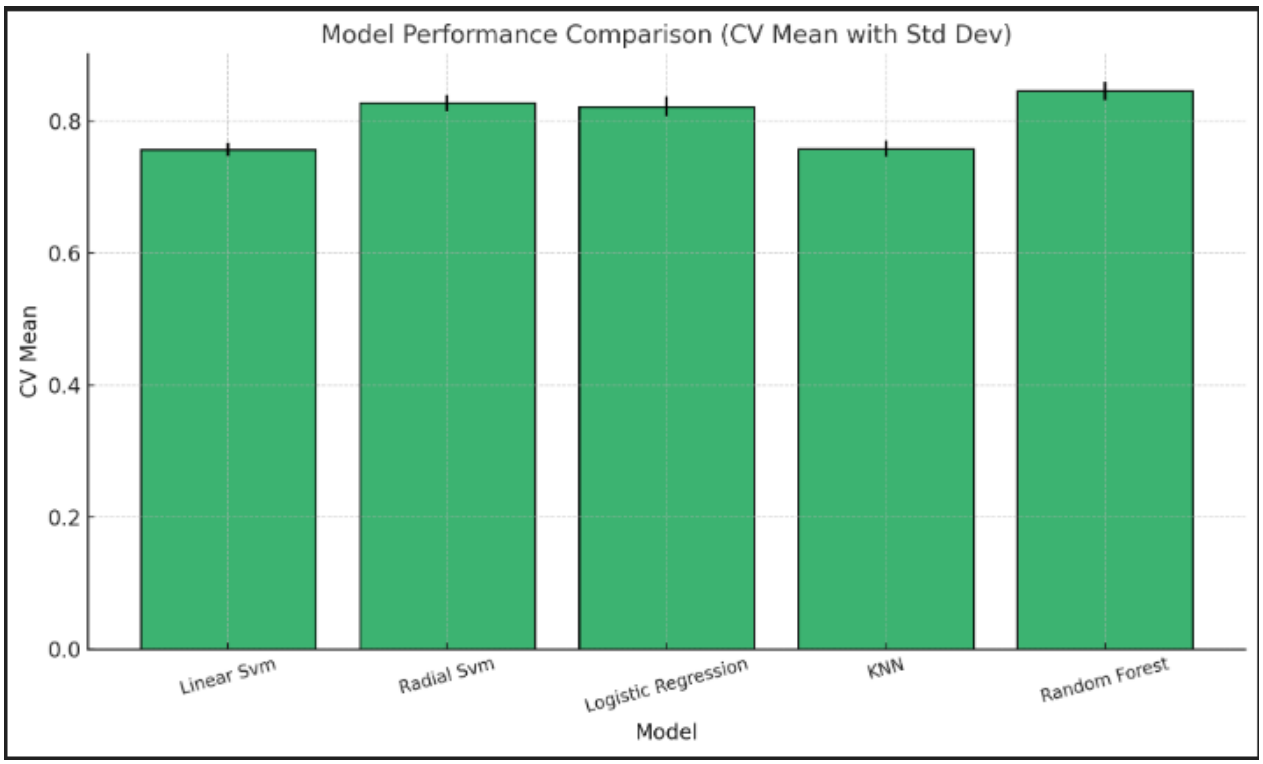
- Linear Support Vector Machine (SVM)
- Radical Support Vector Machine (SVM)
- Logistic Regression
- Random Forest
- K-Nearest Neighbors (KNN)

Cross Validation

Cross-validation was performed to evaluate model performance.

```
from sklearn.model_selection import cross_val_score

score_ppn = cross_val_score(ppn, X, y, cv=5)
```



Model	CV Mean	Std
Linear Svm	0.757101	0.010
Radial Svm	0.827400	0.012
Logistic Regression	0.822355	0.015
KNN	0.758022	0.012
Random Forest	0.845693	0.014

Summary of Performance:

Random Forest:
The highest CV mean (0.8457), indicating it is the most effective model for this dataset.

Moderate variability with a standard deviation of 0.014.

Logistic Regression:
Second-highest CV mean (0.8224).

Slightly higher variability (Std = 0.015) compared to Random Forest, which suggests it may not be as consistent.

KNN:
CV mean (0.7580), slightly higher than both SVM models but significantly lower than Random Forest and Logistic Regression.

Standard deviation of 0.012 suggests reasonable consistency.

Linear SVM and Radial SVM:

The initial model, LSM, achieved an accuracy of 0.757101. However, by utilizing the Kernel RSVM (Reduced Support Vector Machine), the accuracy improved significantly to 0.8274. This demonstrates that the Kernel RSVM is better suited for capturing the underlying patterns in the data, potentially due to its ability to handle non-linear relationships more effectively compared to the LSM.

Results and Takeaways

Key Learnings

- PCA is useful for dimensionality reduction and visualization
- Proper data cleaning and feature engineering significantly improve model performance
- Random Forest emerged as the most accurate model for this dataset

Tradeoffs

- The dataset may suffer from selection bias, as it excludes individuals with multiple races
- The data is outdated (from 1994), which may limit its applicability to current scenarios
- The large dataset size increases computational time, especially for GridSearch