**ECS 111: Assignment 2**
**Classification - Census Income Prediction**
**Varshini Peddinti**

**Overview:** The Census Income Classification assignment provides a dataset from the 1994 US Census database. The goal of this assignment is to apply a classification algorithm, in this case, using decision trees, to predict whether certain attributes impact/predict whether an individual's annual income exceeds $50,000. My report provides the classification techniques applied to create and evaluate my model.

## Model and Hyperparameters:

The classification model chosen was a Random Forest Classifier.

```
from sklearn.ensemble import RandomForestClassifier

decision_tree = RandomForestClassifier(n_estimators=100, max_depth=10, random_state=42)
decision_tree.fit(X_train_final, y_train)
Executed at 2025.05.08 10:06:48 in 2s 957ms
```

```
RandomForestClassifier
RandomForestClassifier(max_depth=10, random_state=42)
```

```
from sklearn.model_selection import GridSearchCV

# Set up a grid of values to search
param_grid = {
    'n_estimators': [50, 100],
    'max_depth': [10, 15]
}

# Use GridSearchCV to try all combinations
grid_search = GridSearchCV(RandomForestClassifier
 (random_state=42), param_grid, cv=3, scoring='accuracy')
grid_search.fit(X_train_final, y_train)

# Print best combination
print("Best parameters:", grid_search.best_params_)
print("Best CV score:", grid_search.best_score_)

# Use the best model found
decision_tree = grid_search.best_estimator_
Executed at 2025.05.08 15:47:17 in 24s 915ms
```

```
Best parameters: {'max_depth': 15, 'n_estimators': 100}
Best CV score: 0.8612101885223821
```

Key hyperparameters tuned using GridSearchCV:

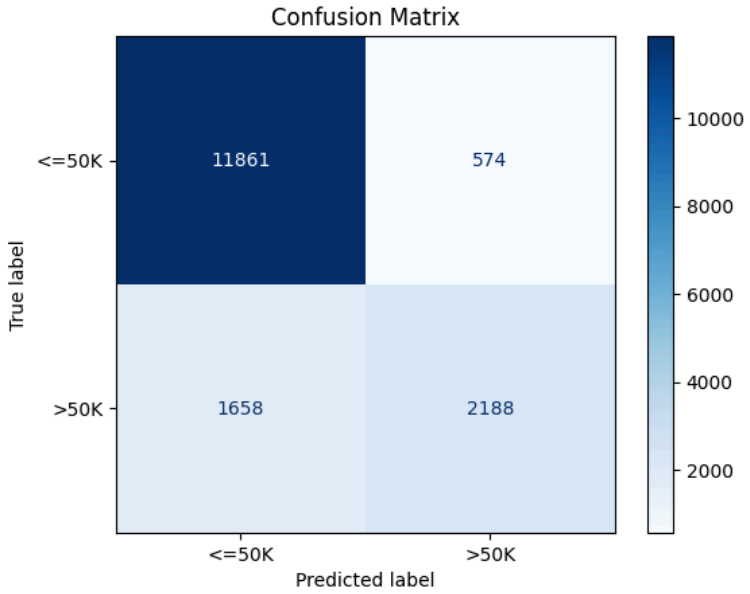
- n_estimators = 100
- max_depth = 15

**These parameters provided the best cross-validation accuracy of 86.12%.**

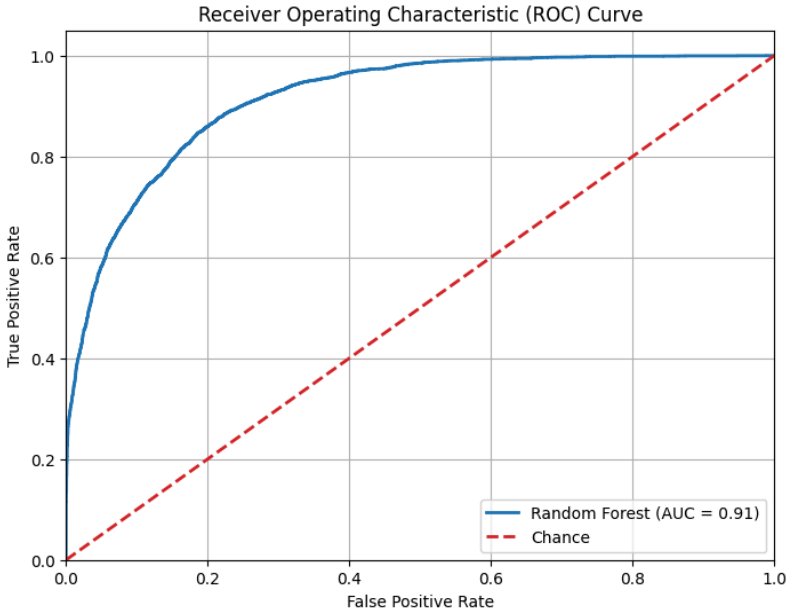
## Evaluation Results

| | |
|---|---|
| Test Accuracy | 0.863 (86.3%) |
| Test Precision | 0.792 (79.2%) |
| Test Recall | 0.569 (56.9%) |
| Test F1-Score | 0.662 (66.2%) |
| Average CV Accuracy | 0.861 (86.1%) |

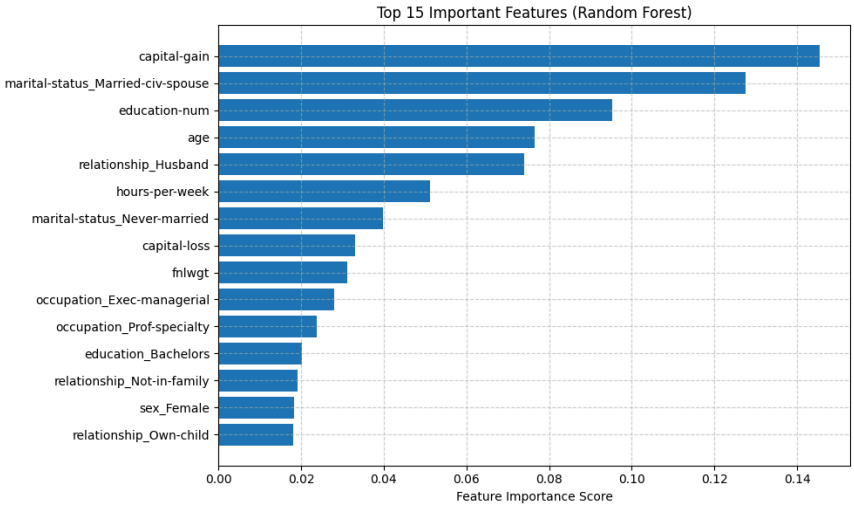
**Confusion Matrix:**



**ROC Curve:**




**Feature Importance (Rankings):**




## Discussion of Model Performance and Observations:

The classifier achieved an accuracy of 86.3% on the test set, indicating strong overall performance. The precision of 79.2% for the high-income class (> $50K) shows that most predicted high earners were truly high earners. However, the recall was 56.9%, meaning the model missed a significant portion of actual high-income individuals, and this suggests a conservative bias in predictions. The F1-score of 66.2% reflects a moderate balance between precision and recall.

Additionally, the model performed consistently across five folds in cross-validation, with an average accuracy of 86.1%, and this confirms its overall ability to generalize. The ROC and confusion matrix indicate that the model balances false positives and false negatives reasonably well, though threshold tuning may help improve recall for the > $50K class. The Feature Importance Chart is helpful in identifying which features are the most important regarding their influence on an individual's annual income. From this chart, we can see that Capital Gain, Marital Status, Education Level, and Age are the Top 4 important features.

While results are strong overall, future improvements could focus on increasing recall to ensure more high-income individuals are correctly identified, potentially by adjusting the classification threshold or rebalancing the training data.

## Fairness and Bias Discussion (Extra Credit)

The goal of our project is to predict whether an individual's income exceeds $50,000 based on features such as age, education, occupation, race, and sex. While this is a valuable predictive task, applying machine learning (ML) to personal income data raises important ethical concerns, especially around fairness, bias, and social impact.

Historical Bias

The dataset is sourced from the 1994 US Census, and during this time, there were social and economic inequalities. As we learned in history, women were historically underpaid and underrepresented in high-paying roles. Additionally, certain races may have had fewer educational or job opportunities due to systemic discrimination.

Label Bias

For our target variable, income, we are predicting an individual's potential to attain an income >50K or <=50K. However, this may not fully reflect the individual's potential, and this number might not best represent different types of populations and communities within society.

Analyzing Model Behavior

From our model, the recall for the >50K was around 57%, which means that the model is missing many actual high-income individuals. However, the model did receive high recall (around 95%) for the <=50K class, which suggests that there is a bias towards predicting low-income outcomes.

Fairness Assessment

If race or sex is correlated with income, this model may discriminate even though those columns aren't directly used within the algorithm. It would be unfair if the model is biased because if it is used in a real-world application, this may continue to reinforce existing inequalities over time.

**Strategies for Improvement: What strategies could be used to mitigate fairness or bias issues?**

- **Remove Sensitive Features:** Dropping features like race or sex can reduce both direct and indirect bias in the model.
- **Reweigh or Resample Training Data:** Balancing the dataset by reweighting or resampling improves fairness during training and helps the model generalize better across groups.
- **Threshold Tuning:** Adjusting classification thresholds can help reduce false negatives for underrepresented or disadvantaged groups.
- **Group-Specific Models:** Training separate models for different subgroups can help capture unique patterns within each group and reduce systemic bias.

## Credit / Resources

To write the descriptions and rationale of my code, I used the Professor's and TA's slides that included information regarding Machine Learning Techniques and Classification Methods that were summarized throughout the assignment.

Some parts of this assignment were developed with the assistance of OpenAI’s ChatGPT to help understand and implement RandomForestClassifier/decision tree technique and the scikit-learn library, design the model and assessment, and resolve any coding errors. All final code was written and reviewed by me.