Intelligent Data Analysis Machine Learning I SoSe 2025: Final Project Income Prediction

Mattia D'Agostini

Universität Potsdam

September 18, 2025

Problem Overview

- **Input:**
  - Type: Tabular data
  - Attributes: Categorical and numerical variables
  - Context: Sociological and demographic information
- **Target:** Binary income class ($> \$50K$)
- **Goal:** The goal was to predict whether an individual earned more than $50K per year.
- In the Machine Learning problem taxonomy, **this problem is a classification problem** (predicting a target value $y \in Y$ where Y is a finite set)

Data Description

- Attributes: 14 attributes
  - Categorical: 8 categorical attributes
  - Numerical: 6 numerical attributes

	employment_type	employment_area	education_level	marital_status	partnership	ethnicity	country_of_birth	gender
0	State-gov	Adm-clerical	Bachelors	Never-married	Not-in-family	White	United-States	Male
1	Self-emp-not-inc	Exec-managerial	Bachelors	Married-civ-spouse	Husband	White	United-States	Male
2	Private	Handlers-cleaners	HS-grad	Divorced	Not-in-family	White	United-States	Male
3	Private	Handlers-cleaners	11th	Married-civ-spouse	Husband	Black	United-States	Male
4	Private	Prof-specialty	Bachelors	Married-civ-spouse	Wife	Black	Cuba	Female

	age	fwf	schooling_period	financial_gains	financial_losses	weekly_working_time
0	39	77516	13	2174	0	40
1	50	83311	13	0	0	13
2	38	215646	9	0	0	40
3	53	234721	7	0	0	40
4	28	338409	13	0	0	40

Data Description: Missing Values

The dataset contains 759 missing values. The missing values are distributed in the columns as follows

- Employment type: 331 missing values
- Employment area: 331 missing values (Correlated to Employment type)
- Country of Birth: 97 missing values

Considering that the missing values appear only in categorical attributes they were not removed and were labeled as unknown.

Data Description: Missing Values

Additionaly, most instances have a missing income value. It is not possible to use these instances neither for training or testing. These instances were dropped.

Data Description: Categorical Attributes

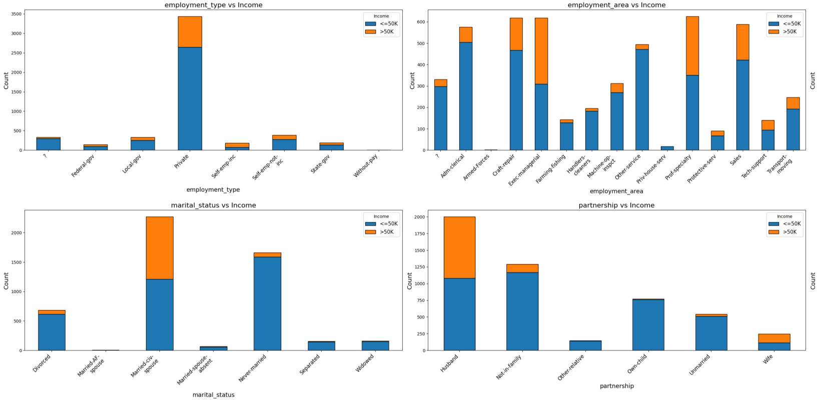


Figure: Distribution of categorical attributes in the dataset (1).

Data Description: Categorical Attributes

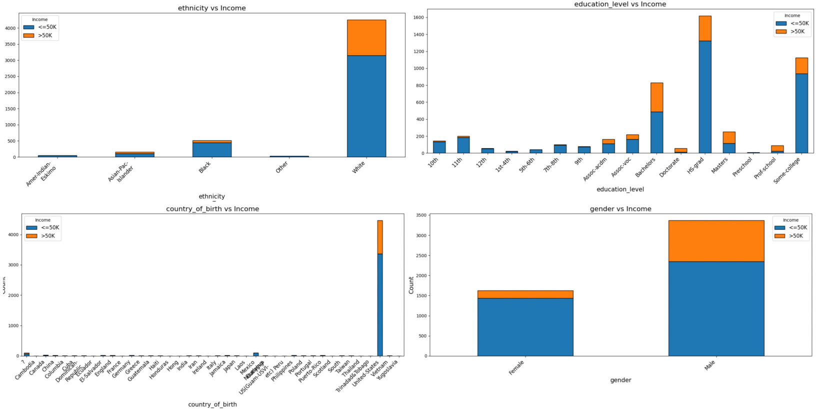
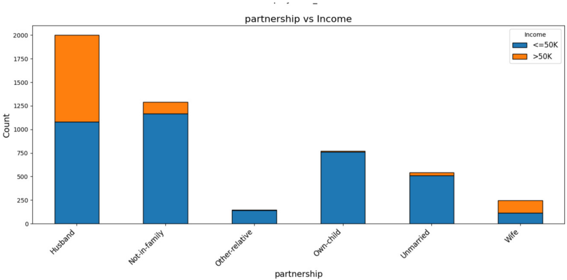


Figure: Distribution of categorical attributes in the dataset (2).

Data Description: Categorical Attributes



- The partnership attribute was very sparse and most of the categories showed a clear predominance of a specific income type. This attribute was mapped into married and not married in order to reduce dimensionality and prevent overfitting.

Data Description: Numerical Attributes

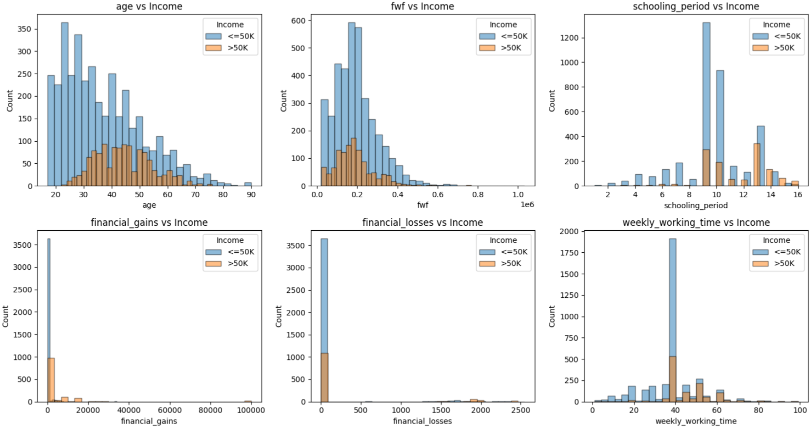
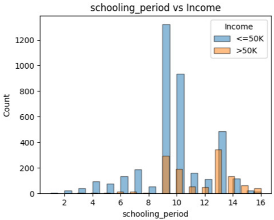
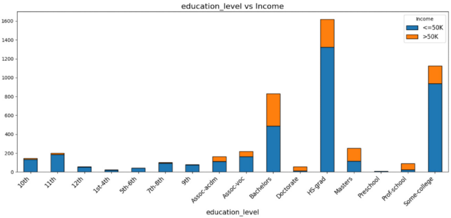


Figure: Distribution of numerical attributes in the dataset.

Data Description: Numerical Attributes





- Education level and schooling period attributes had the same meaning. To favour categorical attributes, the schooling period column was dropped.

Data Preprocessing

1. Categorical attributes were preprocessed using one-hot encoding. For each attribute, this approach generated as many columns as there were distinct attribute values. Each column contained a binary indicator (0 or 1), where 1 denoted the presence of the corresponding attribute value for a given subject, and 0 otherwise. This ensured that all categorical attributes were represented by numerical values.
2. Numerical attributes were normalized using a standard scaler (with $\mu = 0$ and $\sigma = 1$). This ensured that gradient descent–based models, such as logistic regression, learned patterns in the data more effectively and without overfitting.

Training

After preprocessing, the dataset was split into Train+Evaluation and Test sets, with the Train+Evaluation set corresponding to 90% of the entire dataset. During the experiment the following models were evaluated:

1. Decision Tree Classifier
2. Random Forest Classifier
3. Support Vector Machine
4. Logistic Regression

Training and Evaluation with Default Parameters

When trained and evaluated with the default parameters, the models obtained the following scores.

Model	**Accuracy**	**F1 Score**	**AUC**
Logistic Regression	0.838	0.609	0.900
Random Forest	0.836	0.620	0.898
SVM	0.856	0.640	0.895
Decision Tree	0.810	0.596	0.749

Training and Evaluation with Default Parameters

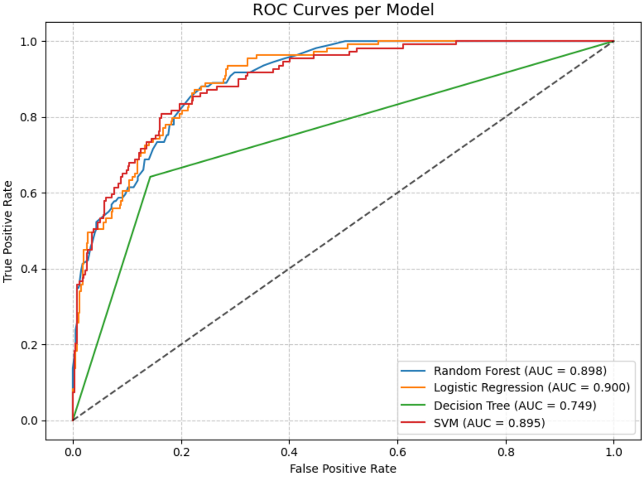


Figure: ROC curves for baseline models

Hyperparameter Tuning

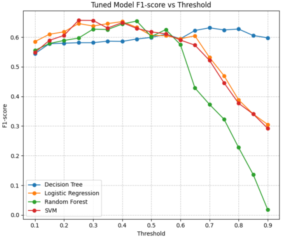
The models' hyperparameters were tuned using a grid search approach, maximizing the AUC score. The grid search was executed with a 5-fold cross validation to ensure the best hyperparameters were chosen.

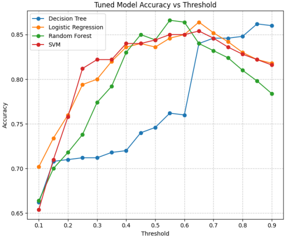
Model	**Hyperparameters Considered**
Decision Tree	max_features,min_samples_split, min_samples_leaf,criterion,max_depth, class_weight
Logistic Regression	C, penalty, solver, max_iter, class_weight
Random Forest	n_estimators,criterion,max_depth, class_weight
SVM	C,gamma,kernel,class_weight

Table: Hyperparameters considered for each model.

Hyperparameter Tuning

The models Hyperparameters were tuned using a grid search approach, maximizing the AUC score. The grid search was executed with a 5-fold cross validation to ensure the best Hyperparameters were chosen. After training, the models were evaluated in terms of accuracy and f1 score by varying the classification threshold. The optimal classification threshold turned out to be not 0.5





Hyperparameter Tuning

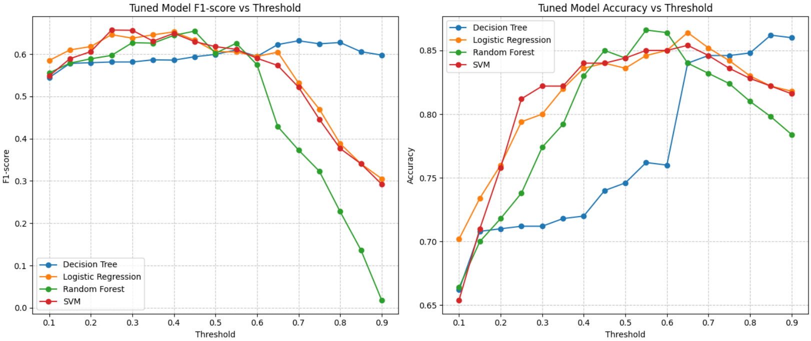


Figure: Accuracy and F1 scores with multiple thresholds on tuned models

Hyperparameter Tuning

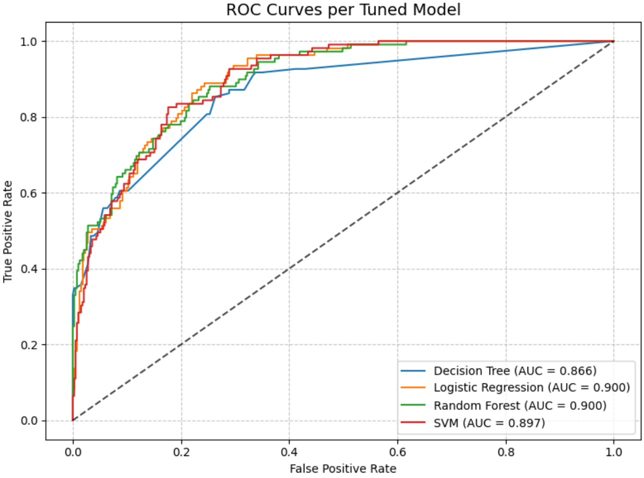


Figure: ROC curves for tuned models

Final Results

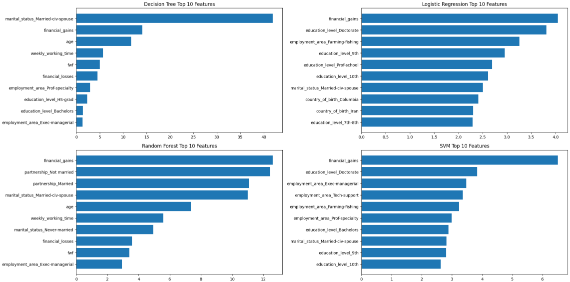
The final results show improvements in Accuracy, F1 score, and AUC after hyperparameter tuning. Although models still struggle to classify higher-income subjects correctly, tuning led to measurable gains.

Model	**Accuracy**	**F1 Score**	**AUC**	**Optimal Threshold**
SVM	0.842	0.615	0.897	0.25
Random Forest	0.844	0.602	0.900	0.45
Logistic Regression	0.836	0.606	0.900	0.40
Decision Tree	0.746	0.599	0.866	0.70

Table: Model performance after hyperparameter tuning.

Final Results: Feature Importance

The most relevant features for each model are reported below. Notably, the Random Forest classifier selected the **married** feature as one of the main decision variables, providing evidence that the preprocessing steps were effective and meaningful.



Final Results: Features Importance

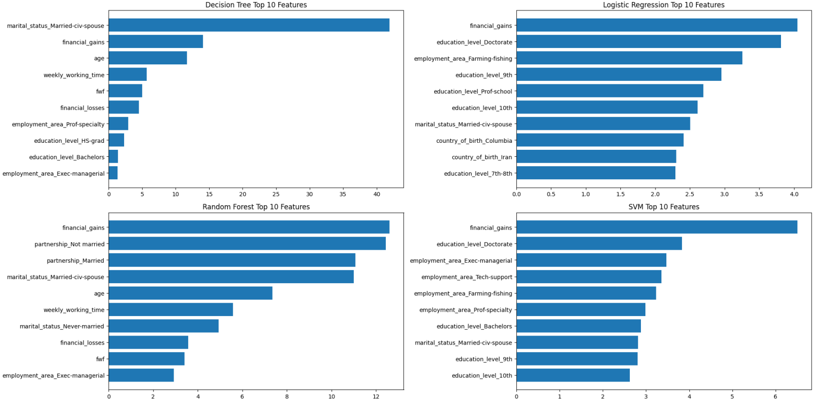


Figure: Most important decision features per tuned model

Using Trained models to predict unknown data

Tuned models can now be trained on the whole dataset (Train Set+ Test Set) and be used to predict unknown income values on the complete dataset.

```python
#Preprocess the unknown labels database
X_unk = preprocess_complete(data_path,names=columns,drop_unk=False)
```

```python
#Preprocess the complete known labels database
X_complete,y_complete = preprocess_complete(data_path,names=columns,drop_unk=True)
```

```python
#Initialize predictions dictionary and predict unknown labels
preds_complete={}
for model_name in models.keys():
    model = models[model_name]
    model.fit(X_complete,y_complete)
    preds = model.predict(X_unk)
    preds_complete[model_name]=preds
```

Figure: Code used to predict the unknown labels

Conclusions

Thank you for your attention
Do you have any questions?