ADULT CENSUS DATASET ANALYSIS

Group Members:

Khine Lin, Thet
Mathew, Benny
Rajesh Narayan, Ashwin
Ben Zekri, Samar

ABOUT DATASET

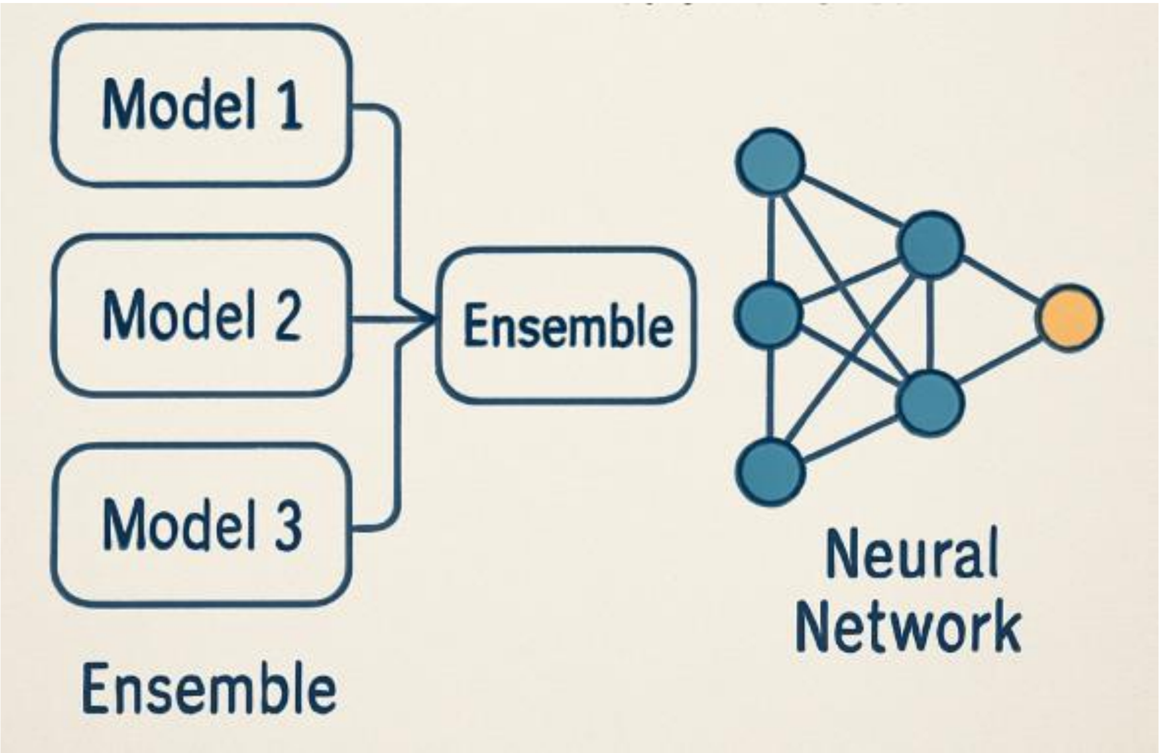
- Adult Census Data from UCI Machine Learning Repository.
- The dataset goal is to predict whether an individual's income exceeds $50K/year.
- It has 48,841 samples and 14 features
- Features are demographic attributes like age, sex, education, race, and etc.
- Mix of categorical and numerical features with missing values.

	B	C	D	E	F	G	H	I	J	K	L	M	N
	workclass	fnlwgt	education	education_	marital_st	occupatio	relationshi	race	sex	capital_ga	capital_lo	hours_per_	native_
39	State-gov	77516	Bachelors	13	Never-ma	Adm-cleric	Not-in-fan	White	Male	2174	0	40	United
50	Self-emp-r	83311	Bachelors	13	Married-ci	Exec-mana	Husband	White	Male	0	0	13	United
38	Private	215646	HS-grad	9	Divorced	Handlers-c	Not-in-fan	White	Male	0	0	40	United
53	Private	234721	11th	7	Married-ci	Handlers-c	Husband	Black	Male	0	0	40	United
28	Private	338409	Bachelors	13	Married-ci	Prof-speci	Wife	Black	Female	0	0	40	Cuba
37	Private	284582	Masters	14	Married-ci	Exec-mana	Wife	White	Female	0	0	40	United
49	Private	160187	9th	5	Married-sp	Other-serv	Not-in-fan	Black	Female	0	0	16	Jamaic
52	Self-emp-r	209642	HS-grad	9	Married-ci	Exec-mana	Husband	White	Male	0	0	45	United
31	Private	45781	Masters	14	Never-ma	Prof-speci	Not-in-fan	White	Female	14084	0	50	United
42	Private	159449	Bachelors	13	Married-ci	Exec-mana	Husband	White	Male	5178	0	40	United
37	Private	280464	Some-coll	10	Married-ci	Exec-mana	Husband	Black	Male	0	0	80	United
30	State-gov	141297	Bachelors	13	Married-ci	Prof-speci	Husband	Asian-Pac-	Male	0	0	40	India
23	Private	122272	Bachelors	13	Never-ma	Adm-cleric	Own-child	White	Female	0	0	30	United
32	Private	205019	Assoc-acd	12	Never-ma	Sales	Not-in-fan	Black	Male	0	0	50	United
40	Private	121772	Assoc-voc	11	Married-ci	Craft-repa	Husband	Asian-Pac-	Male	0	0	40	
34	Private	245487	7th-8th	4	Married-ci	Transport-	Husband	Amer-India	Male	0	0	45	Mexico
25	Self-emp-r	176756	HS-grad	9	Never-ma	Farming-fi	Own-child	White	Male	0	0	35	United
32	Private	186824	HS-grad	9	Never-ma	Machine-c	Unmarried	White	Male	0	0	40	United
38	Private	28887	11th	7	Married-ci	Sales	Husband	White	Male	0	0	50	United
43	Self-emp-r	292175	Masters	14	Divorced	Exec-mana	Unmarried	White	Female	0	0	45	United
40	Private	193524	Doctorate	16	Married-ci	Prof-speci	Husband	White	Male	0	0	60	United
54	Private	302146	HS-grad	9	Separated	Other-serv	Unmarried	Black	Female	0	0	20	United
35	Federal-go	76845	9th	5	Married-ci	Farming-fi	Husband	Black	Male	0	0	40	United
43	Private	117037	11th	7	Married-ci	Transport-	Husband	White	Male	0	2042	40	United
59	Private	109015	HS-grad	9	Divorced	Tech-supp	Unmarried	White	Female	0	0	40	United

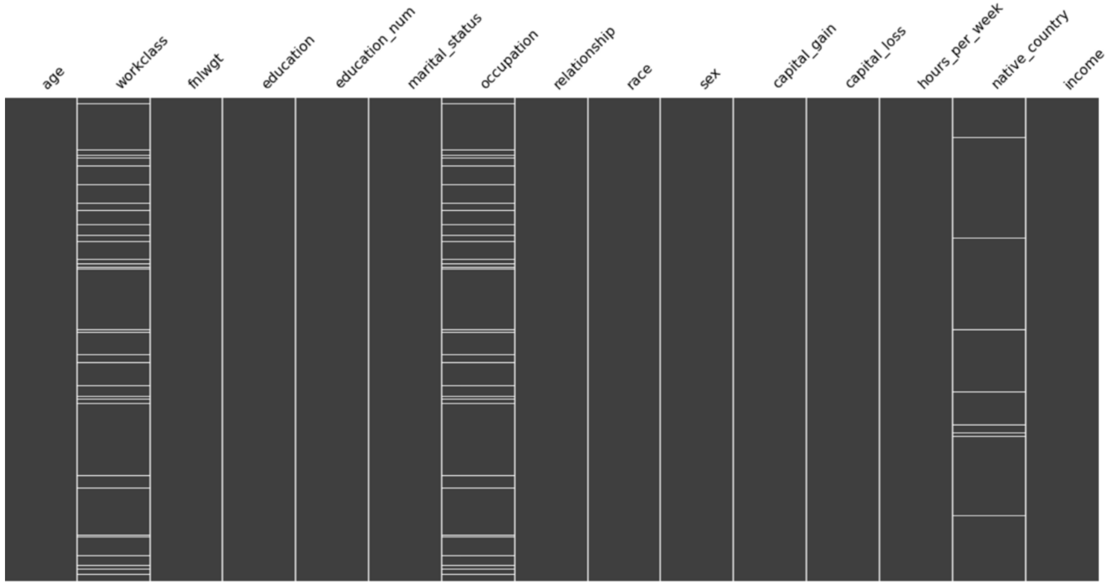
MOTIVATION OF USING PARALLELIZATION

01 The dataset is large and complex.

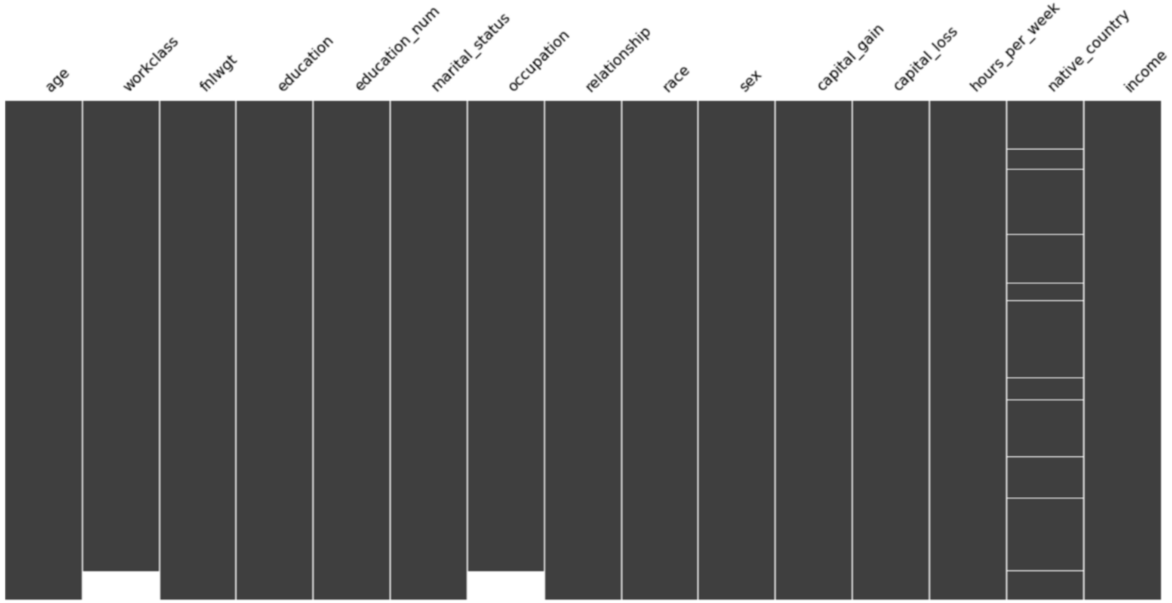
02 Models used are ensemble learning models and neural network model.



MISSING DATA EXPLORATION



Nullity Matrix Before Sorting by occupation



Nullity Matrix After Sorting by occupation.

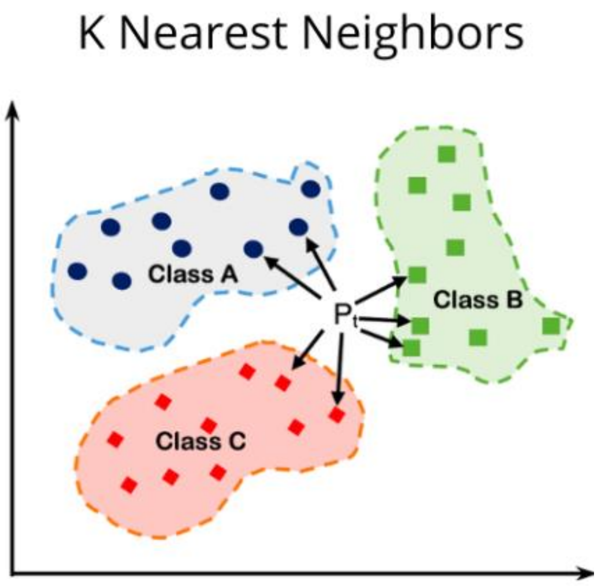
- Missing not at random in workclass and occupation.
- Missing at random in native_country.
- Decided to drop the samples contain missing value in native_country.

MISSING DATA IMPUTATION AND VALIDATION

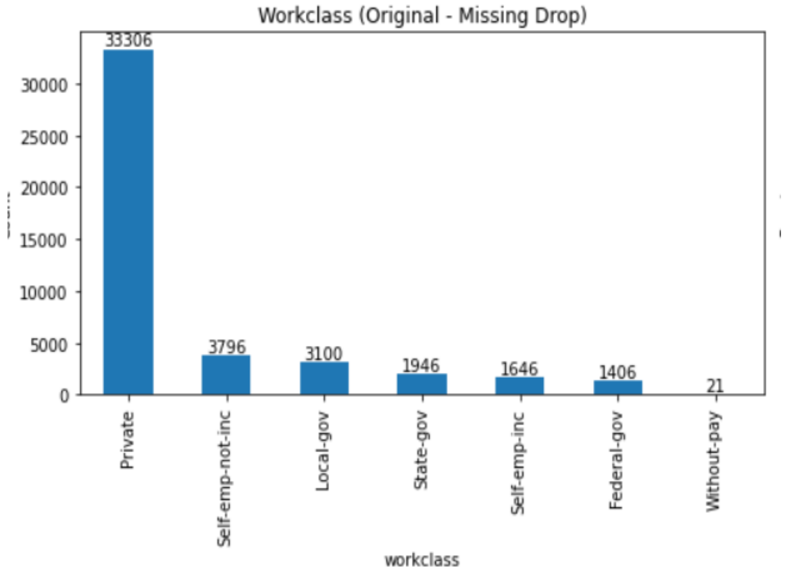
01 Impute the missing values using KNN

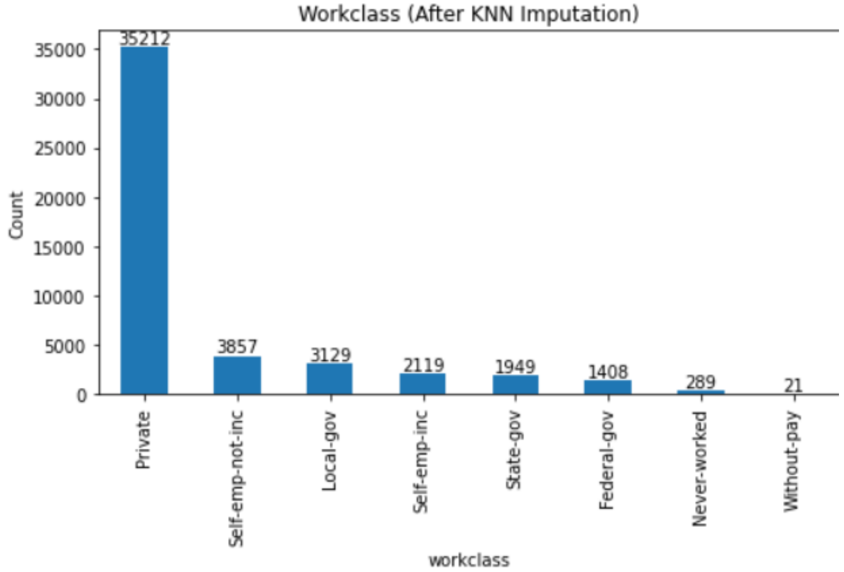
02 Validate the dataset:

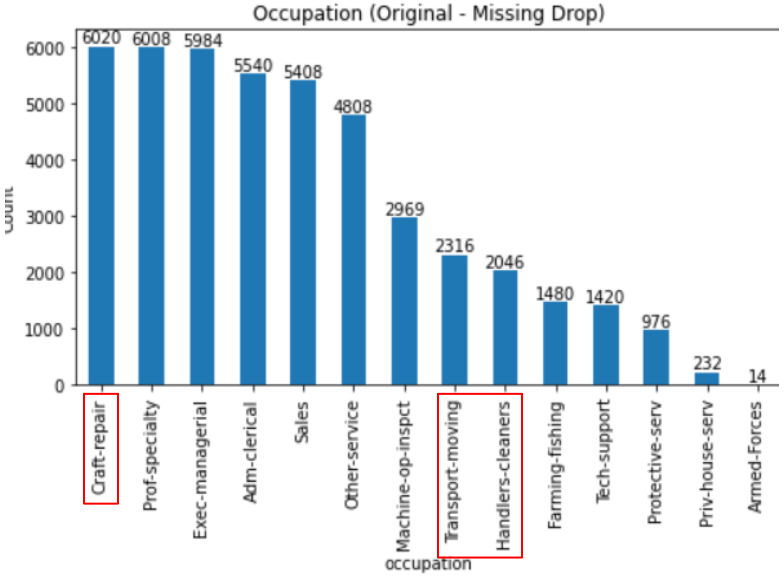
- Check Distribution of Before and After Imputation.
- Check Order of Statistical Strength of Before and After Imputation.
- Check ML Model performance of Before and After Imputation

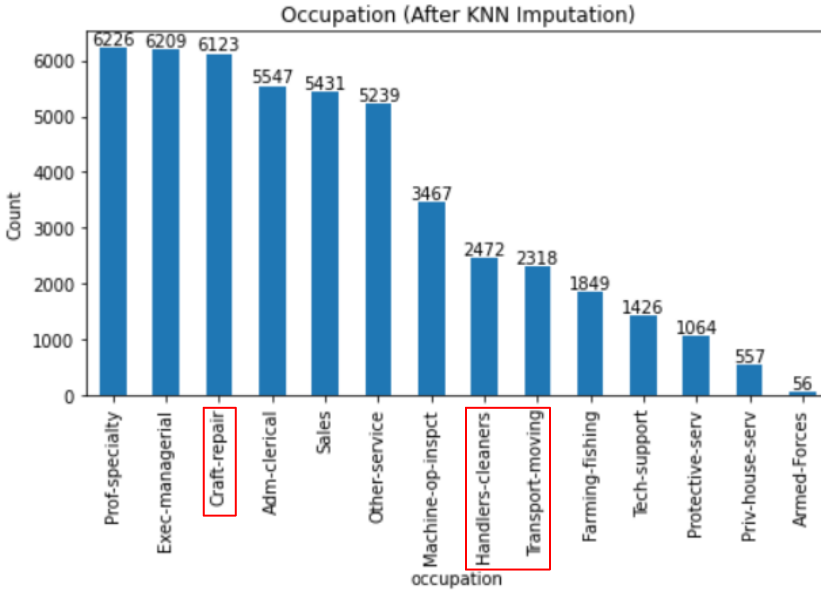


DISTRIBUTION OF BEFORE AND AFTER IMPUTATION









STATISTICAL STRENGTH OF BEFORE AND AFTER IMPUTATION

Cramer's V Before

	Target	Compared With	Cramér's V
13	occupation	sex	0.4357
15	occupation	income	0.3460
8	occupation	workclass	0.2169
9	occupation	education	0.1967
11	occupation	relationship	0.1770
10	occupation	marital_status	0.1304
12	occupation	race	0.0818
14	occupation	native_country	0.0726
2	workclass	occupation	0.2169
7	workclass	income	0.1634
5	workclass	sex	0.1440
0	workclass	education	0.1097
3	workclass	relationship	0.0887
1	workclass	marital_status	0.0774
4	workclass	race	0.0596
6	workclass	native_country	0.0488

Cramer's V After

	Target	Compared With	Cramér's V
13	occupation	sex	0.4076
15	occupation	income	0.3402
8	occupation	workclass	0.1904
9	occupation	education	0.1885
11	occupation	relationship	0.1693
10	occupation	marital_status	0.1242
12	occupation	race	0.0765
14	occupation	native_country	0.0627
2	workclass	occupation	0.1904
7	workclass	income	0.1444
5	workclass	sex	0.1322
0	workclass	education	0.0987
3	workclass	relationship	0.0850
1	workclass	marital_status	0.0759
4	workclass	race	0.0569
6	workclass	native_country	0.0450

Eta Squared Before

	Target	Compared With	Eta Squared
8	occupation	education_num	0.3351
11	occupation	hours_per_week	0.0936
6	occupation	age	0.0433
9	occupation	capital_gain	0.0155
10	occupation	capital_loss	0.0066
7	occupation	fnlwgt	0.0030
0	workclass	age	0.0524
2	workclass	education_num	0.0356
5	workclass	hours_per_week	0.0247
3	workclass	capital_gain	0.0057
1	workclass	fnlwgt	0.0035
4	workclass	capital_loss	0.0016

Eta Squared After

	Target	Compared With	Eta Squared
8	occupation	education_num	0.3060
11	occupation	hours_per_week	0.0897
6	occupation	age	0.0371
9	occupation	capital_gain	0.0144
10	occupation	capital_loss	0.0062
7	occupation	fnlwgt	0.0021
0	workclass	age	0.0507
2	workclass	education_num	0.0330
5	workclass	hours_per_week	0.0165
3	workclass	capital_gain	0.0043
1	workclass	fnlwgt	0.0036
4	workclass	capital_loss	0.0013

Imputed dataset is consistent with the original dataset's statistical relationships.

ML MODEL PERFORMANCE OF BEFORE AND AFTER IMPUTATION

- Used Random Forest model
    - Accuracy (Before Imputation): 84.93 %
    - Accuracy (After Imputation): 85.14 %

Imputed dataset is valid and doesn’t effect too much on the model performance.

FEATURE SELECTION WITH STATISTICAL TESTS

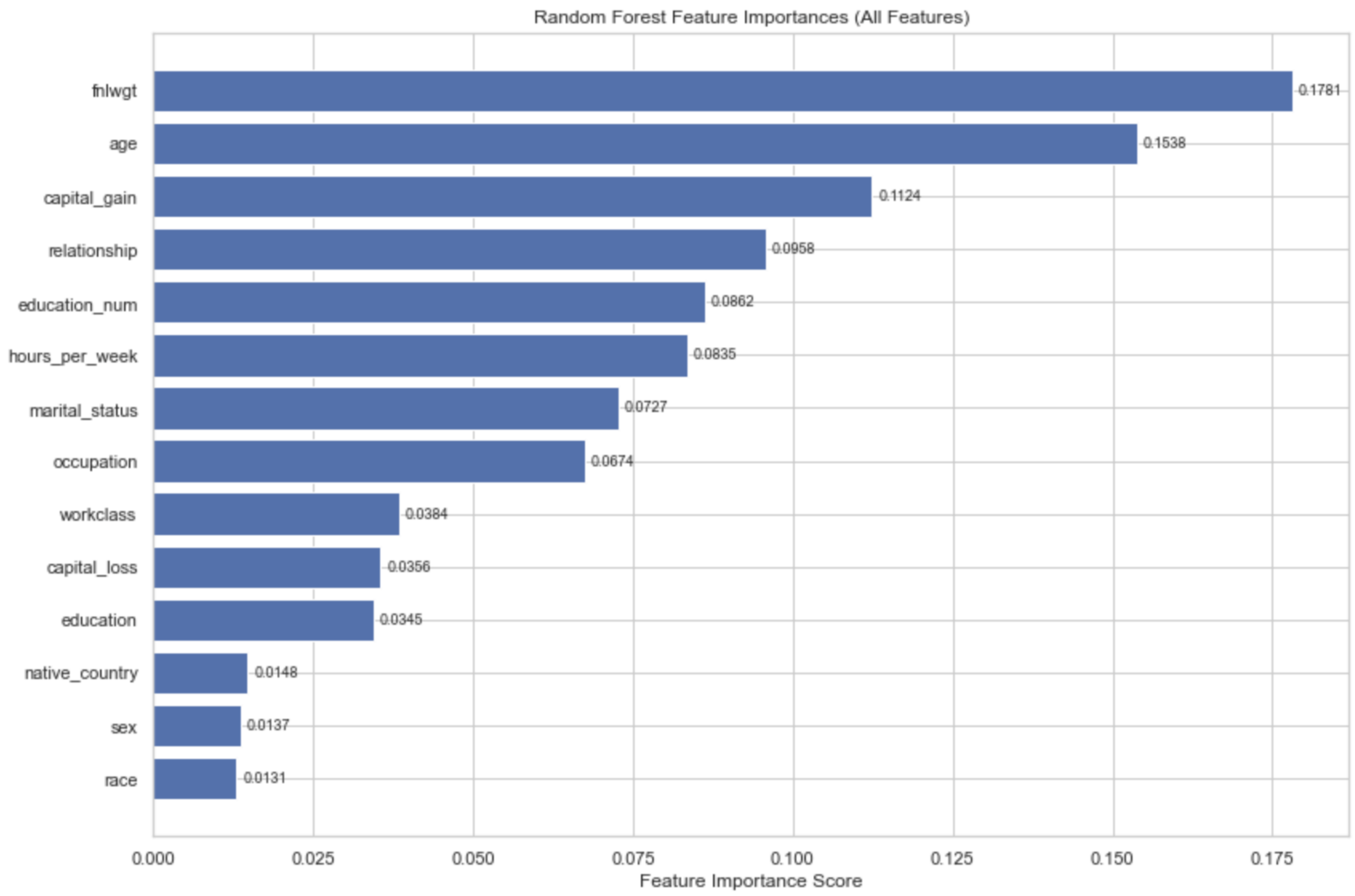
- Statistical tests between one feature and target column (income).

```
🔍 Kruskal-Wallis H Test Results:

age: H-statistic = 2752.7824, p-value = 0.0000
fnlwgt: H-statistic = 1.3173, p-value = 0.2511
education_num: H-statistic = 4196.1020, p-value = 0.0000
capital_gain: H-statistic = 2919.4135, p-value = 0.0000
capital_loss: H-statistic = 724.5646, p-value = 0.0000
hours_per_week: H-statistic = 2831.7150, p-value = 0.0000
```

```
🔍 Chi-Square Test Results:

         Feature  Chi2 Statistic  Degrees of Freedom        p-value
4   relationship     7883.639398                   5   0.000000e+00
2 marital_status     7652.648366                   6   0.000000e+00
1      education     5255.731301                  15   0.000000e+00
3     occupation     4480.064759                  13   0.000000e+00
6            sex     1778.554180                   1   0.000000e+00
0      workclass      798.120226                   7  4.722120e-168
5           race      395.351340                   4   2.809907e-84
7 native_country      355.470540                  40   3.310812e-52
```

FEATURE IMPORTANCE FROM RANDOM FOREST

- To assess how each feature contributes to prediction performance when considering in combination with other



DROPPED FEATURES

- Considered to drop: fnlwgt, sex, race, and native_country
- Actual dropped: fnlwgt, sex, and race

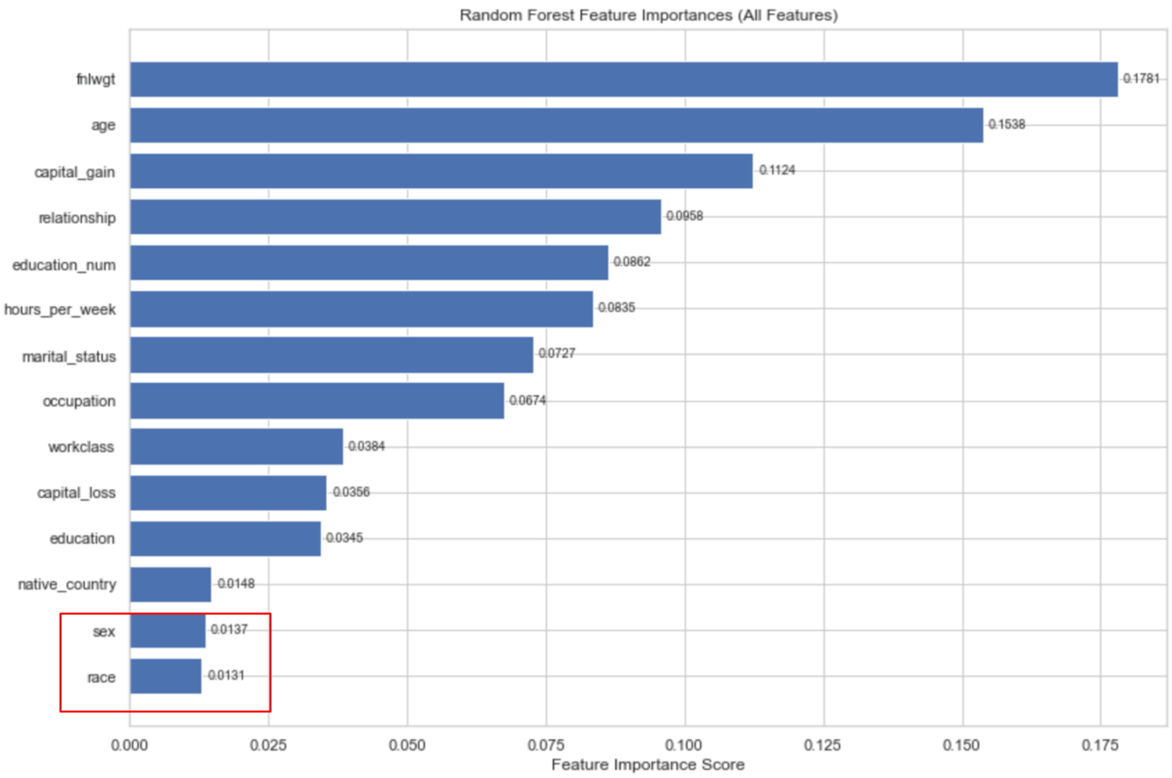
Explanation on Dropped Features

🔍 Chi-Square Test Results:

	Feature	Chi2 Statistic	Degrees of Freedom	p-value
4	relationship	7883.639398	5	0.000000e+00
2	marital_status	7652.648366	6	0.000000e+00
1	education	5255.731301	15	0.000000e+00
3	occupation	4480.064759	13	0.000000e+00
6	sex	1778.554180	1	0.000000e+00
0	workclass	798.120226	7	4.722120e-168
5	race	395.351340	4	2.809907e-84
7	native_country	355.470540	40	3.310812e-52

- Sex and race are highly correlated with income column as their statistical values are so significant.

Note: correlation does not imply causation



- Sex and race have low importance in combine effect.
- Dropping sex and race don’t make noticeable difference in model performance.
- Dropping native_country effect on predicting minority class (based on experiment).

Explanation on Dropped Features

Final Weight

Fnlwgt reflects how many people in the US population the record represents based on the sampling design of the survey.

Since it doesn't contain any inherent information, about a person's characteristics (age, education, occupation etc), it doesn't help a model learn patterns about income.

The feature, though having the highest feature importance is also statistically insignificant.

Including it might introduce bias or cause the model to overfit.

MODELS' HYPERPARAMETER TUNING

Trained four models: GBM, AdaBoost, Random Forest, ANN

GBM

```
gbm_params = {
    "n_estimators": [100, 200],
    "learning_rate": [0.01, 0.1],
    "max_depth": [3, 5],
    "min_samples_split": [2, 5],
    "min_samples_leaf": [1, 2, 5]
}
```

```
# GridSearchCV
gbm_grid = GridSearchCV(gbm, gbm_params, cv=5, n_jobs=-1, verbose=1)
gbm_grid.fit(X_train, y_train)
```

- Tree-based models have similar hyperparameters.
- Tune hyperparameter with GridSearchCV.
- Control Overfitting with CrossValidation.

ANN

```
model = Sequential([
    Dense(50, activation='relu', kernel_initializer=HeNormal(), input_shape=(X_train.shape[1],)),
    BatchNormalization(),
    Dropout(dropout_rate),

    Dense(25, activation='relu', kernel_initializer=HeNormal()),
    BatchNormalization(),
    Dropout(dropout_rate),

    Dense(1, activation='sigmoid', kernel_initializer=GlorotUniform())
])
```

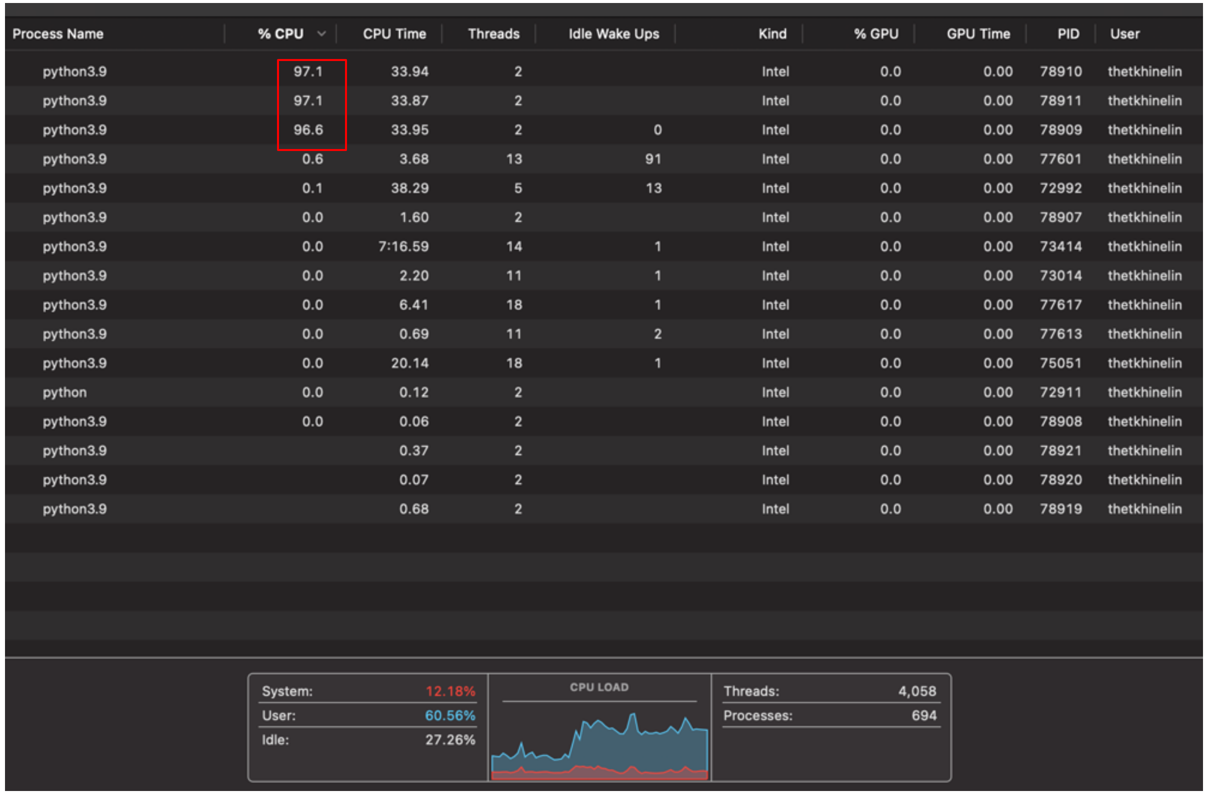
```
ann_params = {
    'epochs': [50, 100],
    'batch_size': [16, 32],
    'optimizer': ['adam', 'rmsprop'],
    'learning_rate': [0.001],
    'dropout_rate': [0.1, 0.3]
}
```

```
# GridSearchCV
ann_grid = GridSearchCV(estimator=model, param_grid=ann_params, n_jobs=-1, cv=3, verbose=1)
ann_grid.fit(X_train, y_train, callbacks=[early_stop], validation_split=0.2)
```

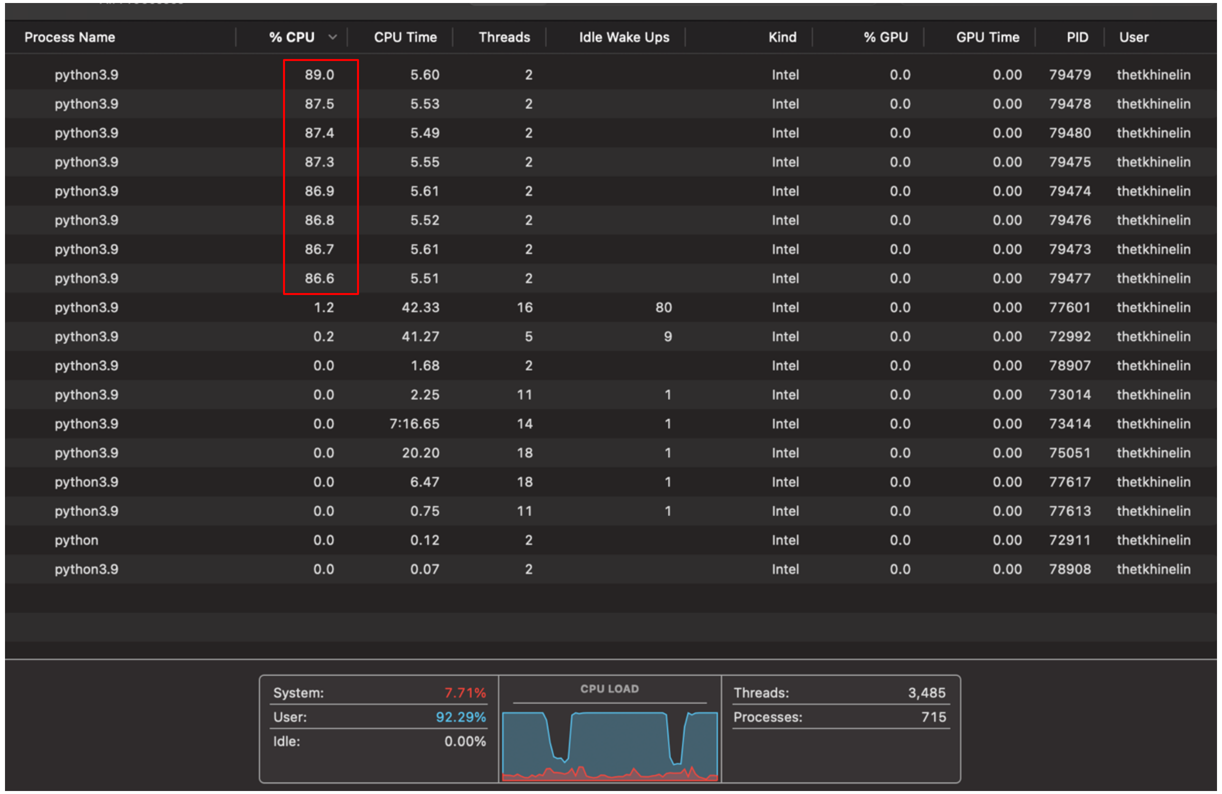
- Need to create ANN structure manually.
- Tune hyperparameter with GridSearchCV.
- Control Overfitting with Dropout, BatchNormalization, and CrossValidation.

CPU BASED PARALLELISATION WITH GBM

- GridSearchCV's n_jobs parameter allows us to do parallelization by using CPU cores.



Process Name	% CPU	CPU Time	Threads	Idle Wake Ups	Kind	% GPU	GPU Time	PID	User
python3.9	97.1	33.94	2		Intel	0.0	0.00	78910	thetkhinelin
python3.9	97.1	33.87	2		Intel	0.0	0.00	78911	thetkhinelin
python3.9	96.6	33.95	2	0	Intel	0.0	0.00	78909	thetkhinelin
python3.9	0.6	3.68	13	91	Intel	0.0	0.00	77601	thetkhinelin
python3.9	0.1	38.29	5	13	Intel	0.0	0.00	72992	thetkhinelin
python3.9	0.0	1.60	2		Intel	0.0	0.00	78907	thetkhinelin
python3.9	0.0	7:16.59	14	1	Intel	0.0	0.00	73414	thetkhinelin
python3.9	0.0	2.20	11	1	Intel	0.0	0.00	73014	thetkhinelin
python3.9	0.0	6.41	18	1	Intel	0.0	0.00	77617	thetkhinelin
python3.9	0.0	0.69	11	2	Intel	0.0	0.00	77613	thetkhinelin
python3.9	0.0	20.14	18	1	Intel	0.0	0.00	75051	thetkhinelin
python	0.0	0.12	2		Intel	0.0	0.00	72911	thetkhinelin
python3.9	0.0	0.06	2		Intel	0.0	0.00	78908	thetkhinelin
python3.9		0.37	2		Intel	0.0	0.00	78921	thetkhinelin
python3.9		0.07	2		Intel	0.0	0.00	78920	thetkhinelin
python3.9		0.68	2		Intel	0.0	0.00	78919	thetkhinelin



Process Name	% CPU	CPU Time	Threads	Idle Wake Ups	Kind	% GPU	GPU Time	PID	User
python3.9	89.0	5.60	2		Intel	0.0	0.00	79479	thetkhinelin
python3.9	87.5	5.53	2		Intel	0.0	0.00	79478	thetkhinelin
python3.9	87.4	5.49	2		Intel	0.0	0.00	79480	thetkhinelin
python3.9	87.3	5.55	2		Intel	0.0	0.00	79475	thetkhinelin
python3.9	86.9	5.61	2		Intel	0.0	0.00	79474	thetkhinelin
python3.9	86.8	5.52	2		Intel	0.0	0.00	79476	thetkhinelin
python3.9	86.7	5.61	2		Intel	0.0	0.00	79473	thetkhinelin
python3.9	86.6	5.51	2		Intel	0.0	0.00	79477	thetkhinelin
python3.9	1.2	42.33	16	80	Intel	0.0	0.00	77601	thetkhinelin
python3.9	0.2	41.27	5	9	Intel	0.0	0.00	72992	thetkhinelin
python3.9	0.0	1.68	2		Intel	0.0	0.00	78907	thetkhinelin
python3.9	0.0	2.25	11	1	Intel	0.0	0.00	73014	thetkhinelin
python3.9	0.0	7:16.65	14	1	Intel	0.0	0.00	73414	thetkhinelin
python3.9	0.0	20.20	18	1	Intel	0.0	0.00	75051	thetkhinelin
python3.9	0.0	6.47	18	1	Intel	0.0	0.00	77617	thetkhinelin
python3.9	0.0	0.75	11	1	Intel	0.0	0.00	77613	thetkhinelin
python	0.0	0.12	2		Intel	0.0	0.00	72911	thetkhinelin
python3.9	0.0	0.07	2		Intel	0.0	0.00	78908	thetkhinelin

- With 3-cores
- Training Time 593.85 seconds (9.9 minutes)

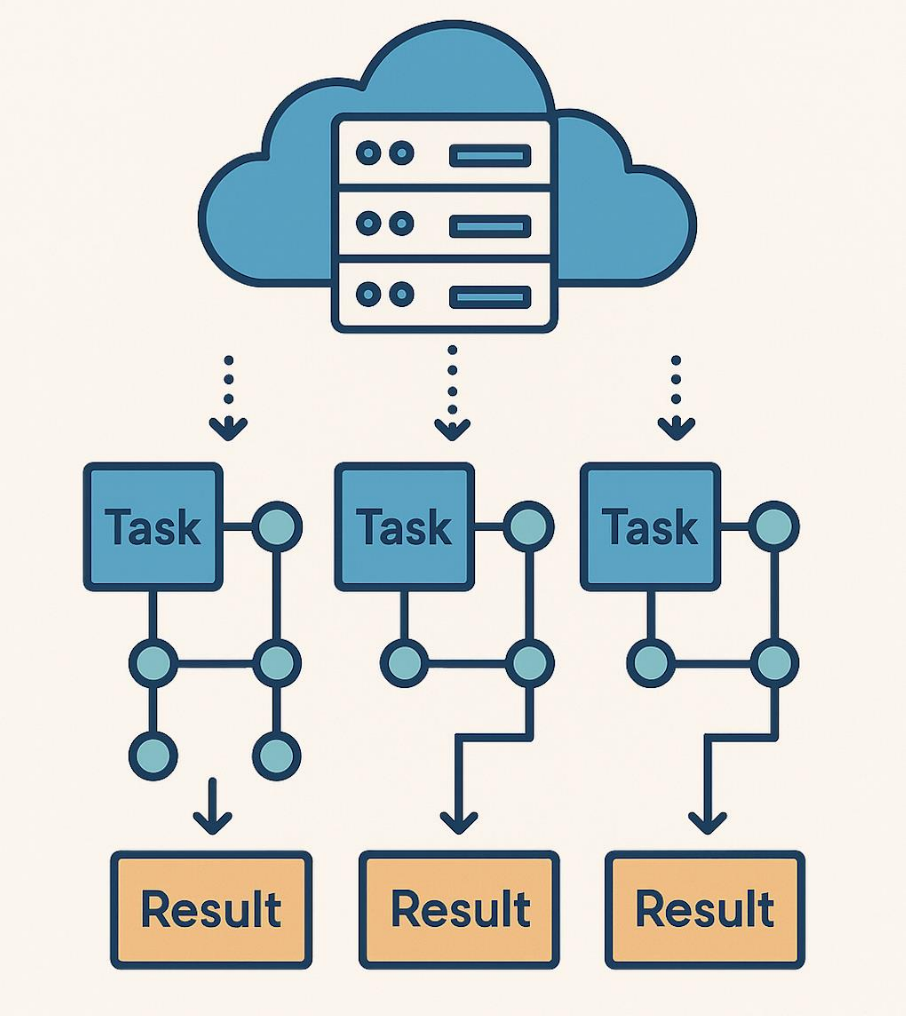
- With 8-cores
- Training Time 402.33 seconds (6.71 minutes)

CLOUD BASED PARALLELIZATION

- GCP, AWS and Azure support tensorflow based parallelization methods

Why didn't we use here?

- Useful only for huge datasets, which cannot be applied in our case
- Time taken to cloud parallelize is slower in this case.



OPTIMIZING FOR IMBALANCE (BEST THRESHOLD)

Why Adjust Threshold?
- Dataset is imbalanced, so Accuracy alone can be misleading
- Need to boost performance on the minority class (>$50K)

How?
- Use AUC-PR Curve
- Select threshold with highest F1 Score
- Evaluate each model again with optimized thresholds

BEST MODEL SELECTION

Key Metrics for Model Choice:

- Accuracy → Performance on both classes
- F1 Score → Performance on minority class (> $50K)

GBM chosen as the best model:

- Highest Accuracy: 0.8776
- Highest F1 Score: 0.7330
- Balances both majority and minority class performance effectively

📋 Model Evaluation Summary for best threshold:

	Model	Accuracy	Balanced Accuracy	F1 Score	AUC-PR
0	GBM	0.8776	0.8178	0.7330	0.8333
1	AdaBoost	0.8626	0.8194	0.7193	0.8176
2	Random Forest	0.8600	0.8103	0.7093	0.8045
3	ANN	0.8492	0.8146	0.7034	0.7797

GBM Classification Report

📊 Classification Report with best threshold:

	precision	recall	f1-score	support
0	0.91	0.93	0.92	7310
1	0.77	0.70	0.73	2295
accuracy			0.88	9605
macro avg	0.84	0.82	0.83	9605
weighted avg	0.87	0.88	0.88	9605

Summary

1. Explore the missing values pattern
2. Impute the missing values and validate the imputed dataset
3. Select most relevant features
4. Train four ML models
5. Evaluate and select the most generalized model

THANK YOU