**A TUTORIAL GUIDE ON INCOME LEVEL PREDICTION USING RANDOM FOREST AS THE MACHINE LEARNING MODEL**

**Topic of the Tutorial:** Income Level Prediction using Random Forest as the ML Model

**Target Audience:** Beginners and Intermediate Learners

**Tools and Techniques covered:** Python 3.x, Scikit-learn, Pandas, Matplotlib, Seaborn, Numpy

This tutorial will explain how to forecast income levels using machine learning with a focus on data preprocessing, selection and assessment, data mining using Python with real-life dataset.

**Introduction**

One of the most important uses of machine learning is to predict income level and thereby forecast economic trends so that decisions in fields like marketing, finance, and social policy can be made smartly. Age, education, occupation, work hours and other characteristics if evaluated and fed to machine learning models provide a very accurate distribution of people by income. Out of all these models, Random Forest algorithm has become progressively popular because of its stability, scalability and it is easy to explain. Like other forms of Ensemble method, Random Forest is a method that builds a number of decision trees and then merges them in order to provide optimized prediction results to help eliminate over training. This makes the approach offer accurate and correct predictions making it the best solution for covering up on classification issues like the income level prediction issue.

In order to demonstrate understanding of sophisticated techniques for analysing numerical data, this study applies a targeted machine-learning strategy to a census dataset. Predicting if a person's income surpasses $50,000 using work-related and demographic characteristics is the aim. Exploratory data analysis (EDA), model building, assessment, data pretreatment, and a structured instruction to use the approach are all included in the process.

**Purpose of the Tutorial**

This tutorial aims at helping the participants follow practical steps in the construction and assessment of a machine learning model to estimate income levels using the Random Forest technique. This tutorial is designed to:

1. Demonstrate the Practical Application of Machine Learning
2. Introduce Participants to Random Forest
3. Enhance Understanding of the Machine Learning Workflow
4. Teach Data Handling and Insights Extraction
5. Enable Practical Implementation
6. Encourage Model Optimization and Interpretation

At the end of this tutorial, the participants should be well equipped with the fundamental understanding of the machine leraning (ML) approach to the income prediction problems and ready to apply all these ideas to other classification problems in different domain areas.

**Dataset Overview**

One target variable and twelve characteristics make up the dataset that was used. Among the characteristics are age, work class, education, marital status, occupation, relationship, race, sex, number of hours worked each week, and native country. Salary (less than or more than $50,000) is the target. The dataset used can be found **[here](https://drive.google.com/file/d/1cgoNXHLmk-jNROjdSjSmuPChLsIXBctn/view?usp=sharing)**.

**Methodology:**

**Machine Learning Technique Used**

**Random Forest**

An ensemble learning technique called Random Forest is primarily applied to problems involving regression and classification. In order to increase prediction accuracy and manage overfitting, it constructs numerous decision trees during training and aggregates their outputs (either by average for regression or by majority vote for classification). The model is more resilient since individual trees are different due to the randomization in feature selection and bootstrapping.

For the analysis, Random Forest was chosen for a number of reasons. The dataset includes both category (like work class and occupation) and numerical (like age and weekly hours) information. Such data types are handled natively by Random Forest without requiring a lot of preparation. Although rows were dropped to manage missing data, Random Forest's built-in noise tolerance guarantees forecast stability. The project's objective of comprehending important predictors is in line with the capability to calculate feature importance.

**Key Features of Random Forest**

Each decision tree is trained on a distinct subset of data that is produced via sampling with replacement. Because just a random subset of attributes is taken into account at each split in a tree, diversity is increased and correlation across trees is decreased. By averaging predictions from all trees, Random Forest's ensemble nature balances bias and variation and avoids overfitting. It is perfect for this research because of its high processing efficiency and ability to scale to huge datasets.

**Mathematically expression of Random Forest Classifier model**

* **Input Data:**

;

where X are the features, and y are the corresponding labels.

* **Model**

The model consists of T decision trees each trained on a random subset of the data and features.

* **Training**

Minimize classification error by training each tree (x) on a bootstrap sample of the dataset

where L is the loss function (e.g., Gini impurity or entropy) and is the subset used to train the t-th tree.

* **Prediction**

The final prediction is the aggregated majority vote from all trees.

**Application to Dataset**

A structured dataset with the target variable salary (<=50K or >50K) was subjected to the Random Forest classifier. The steps taken were:

* **Preprocessing:** Numerical characteristics and categorical variables were standardized and encoded. This guaranteed that the model and the dataset were compatible.
* **Training:** By using bootstrapped samples, the model was trained on 80% of the data.
* **Evaluation:** The efficacy of the predictions on the test set was evaluated using measures like as accuracy, precision, recall, and ROC-AUC. The Python codes used can be found **[here](https://drive.google.com/file/d/15CRNBdbJ5XumUHDdxqqFHvwMGhVSVaB1/view?usp=sharing)**.

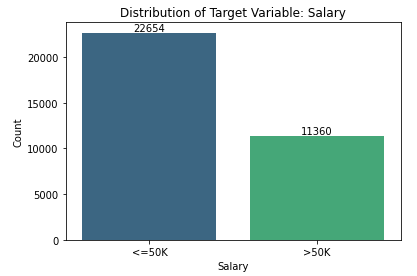
**Broader Implications**

The Random Forest approach successfully illustrated its relevance in socioeconomic research, where interpretability and a variety of data sources are essential. Data-driven decision-making was aided by the method's insights into important elements influencing income disparities.

The dataset was analysed using Random Forest, a reliable and efficient machine learning method. It was ideal for comprehending pay predictors because of its adaptability, feature importance analysis, and potent predictive powers. It is still a fundamental algorithm in contemporary machine learning applications, with room for improvement.

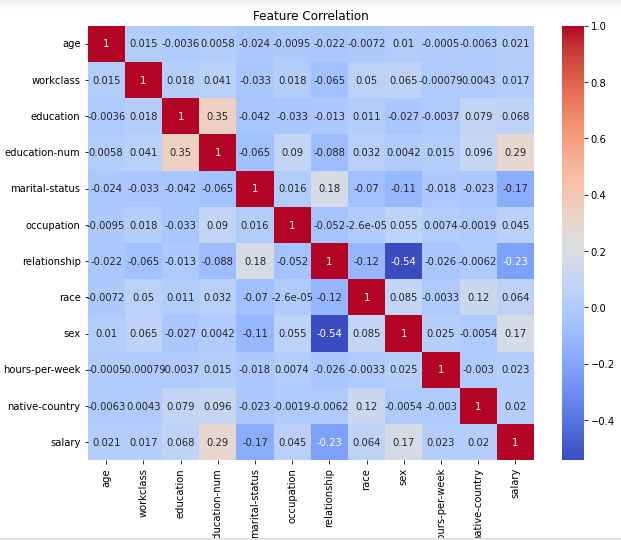
**EDA and Machine Learning Implementation**

**Visualization**



The target variable "Salary" distribution is displayed in the bar-chat. There are a lot more cases in the "<=50K" category (22,654 instances) than in the ">50K" category (11,360 instances), which indicates that the dataset is unbalanced. The majority of people make $50,000 or less annually. A lower percentage of people make above $50,000 annually.

**Correlation Heatmap**



When "education-num" and "salary" have a positive correlation value of 0.8, it indicates a significant positive link. Salary tends to rise in tandem with education level.

A score of -0.4 for the correlation between "salary" and "marital-status" indicates a somewhat unfavourable association. On average, being married may be linked to a lower pay.

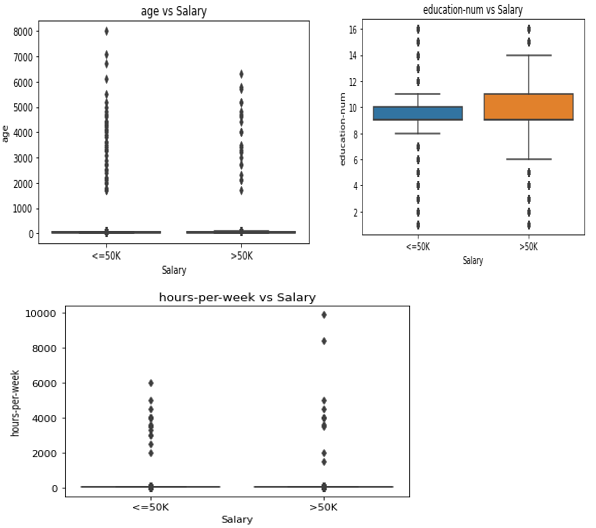
The correlation between "age" and "workclass" is nearly zero for no correlation, indicating that there isn't a significant association between the two variables.

A number of factors, including "education-num" and "occupation," have substantial associations with "salary." These factors may be important indicators of pay.

Certain factors, such as "age" and "race," have less of a relationship with "salary." These factors may not have as much of an effect on wage prediction.

Additionally, the matrix aids in locating any multicollinearity problems. Two variables that have a high degree of correlation with one another may be redundant in the model and cause instability in the coefficients.

**Boxplots for numerical features against the target variable (salary)**

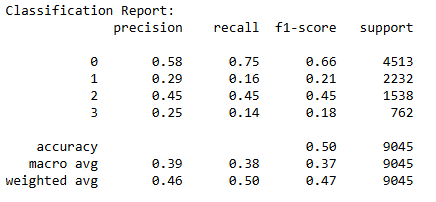


The age distribution for the two wage categories "<=50K" and ">50K" is shown in a box plot. Both wage categories' age distributions seem to be right-skewed, with a higher percentage of people falling into the lower age ranges. The individual points outside the box plots' whiskers show that there are outliers in both categories.

box plot showing the "education-num" distribution for the "<=50K" and ">50K" salary categories. The distribution of "education-num," with the majority of people falling within a particular range, is comparable for both groups. The individual points outside the box plots' whiskers show that there are outliers in both categories.

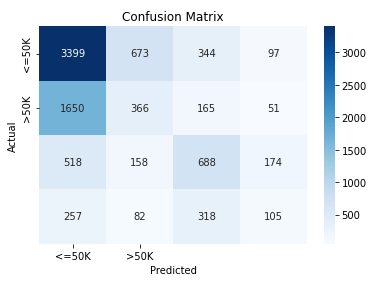
A box plot showing the "hours-per-week" distribution for the "<=50K" and ">50K" wage categories. The distribution of "hours-per-week," with the majority of people falling within a particular range, is comparable for both categories. The individual points outside the box plots' whiskers show that there are outliers in both categories.

**Model Evaluation**



The classification model's total accuracy of 50% indicates an acceptable degree of performance. It does well in class 0 (precision: 0.58, recall: 0.75, F1-score: 0.66), but it has trouble with low precision and recall in class 1 (precision: 0.29, recall: 0.16, F1-score: 0.21) and class 3 (precision: 0.25, recall: 0.14, F1-score: 0.18). With an accuracy of 0.45, recall of 0.45, and F1-score of 0.45, Class 2 performs moderately. When it comes to larger classes, the weighted average offers slightly better outcomes, but the macro average measures demonstrate low overall performance across all classes. To increase the model's accuracy and recall for underperforming classes, it would be advantageous to implement strategies to solve class imbalance and maybe more sophisticated algorithms.

**Confusion Matrix**

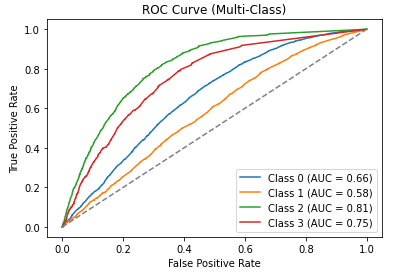


A multi-class classifier's performance is displayed in the confusion matrix. This is a summary:

The classifier accurately predicted that <=50K for 3399 occurrences and >50K for 688 instances, according to True Positives, or the diagonal cells.

False Positives are Off-Diagonal, with real values in the rows and anticipated values in the columns. The 1650 cases of <=50K were incorrectly categorised as >50K, which is a significant mistake. The large number of misclassifications indicates that the model struggles with other classes but performs better for <=50K. Results are probably impacted by class imbalance.

**ROC curve**



The model's capacity to discriminate across classes is revealed by the multi-class ROC curve.

1. AUC = 0.81 for Class 2:

The model works effectively, demonstrating a great capacity to distinguish one class from others. For Class 2, a high AUC denotes a low false positive rate and a high true positive rate.

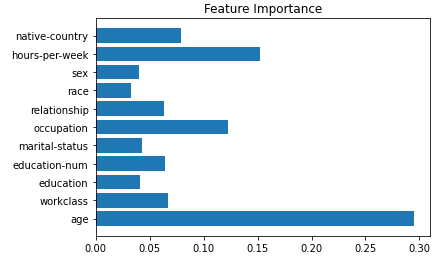
2. Class 3 (AUC = 0.75), which performs well but is little less accurate than Class 2. Although there is potential for improvement, the model consistently separates this class.

3. Class 0: Moderate capacity for categorisation (AUC = 0.66). There is some overlap between genuine and false positives, which lowers Class 0 prediction accuracy.

4. Class 1: Poorest performance, with the curve on the diagonal (AUC = 0.58). shows that it is difficult to classify Class 1 accurately, with forecasts that are almost entirely based on guesswork.

While Class 0 and Class 1 require optimisation to increase separability and prediction accuracy, Class 2 and Class 3 exhibit the best model performance.

**Feature Importance**



The feature significance table illustrates how each feature contributes differently to the model's predictions: Age is the most significant element that influences projections, followed by hours per week, which is the second most important component. Occupation, native-country education, and labour class are of minor importance. Relationships, race, and sex have the least influence. The model suggests that age and weekly hours are powerful determinants.

**Conclusions and Recommendations**

In terms of income level prediction, the Random Forest model performed mediocrely, showing good accuracy for certain wage categories but noticeable difficulties with unbalanced data. The most significant characteristics were age and weekly hours, but factors like sex and ethnicity showed little predictive power. The model's accuracy in classifying higher-income groups was severely hampered by the dataset's imbalance, highlighting the necessity of resolving this problem. Notwithstanding these constraints, the research offered insightful information about socioeconomic factors that impact income, emphasising the importance of work hours and education in comprehending income difference.

A number of suggestions are made to improve the model's usefulness and performance. First, improving the categorisation of under-represented income groups requires correcting class imbalance using methods like SMOTE, ADASYN, or class weighting. Furthermore, model accuracy and recall may be improved by hyperparameter optimisation employing techniques like grid search or Bayesian optimisation. For unbalanced datasets, investigating more sophisticated algorithms, such as Gradient Boosting models like XGBoost or LightGBM, may produce superior outcomes.

Furthermore, the predictive power of the model might be improved by employing dimensionality reduction strategies like PCA or feature engineering to develop additional predictors. This will be treated in the future class. Tools that offer actionable insights, such as SHAP values or LIME, should be used for better interpretability. Lastly, putting the approach to use in practical settings would confirm its usefulness and make data-driven choices for resolving income inequality easier.

**ReadMe:** The ReadMe file can be found **[here](https://drive.google.com/file/d/11-mZMmJPy_K3Lh1N9otoWkf0720UjEwA/view?usp=sharing)**.

**References**

Biau, G., 2012. Analysis of a Random Forests Model. Journal of Machine Learning Research, 13, pp. 1063- 1095.

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., 2002. SMOTE: Synthetic Minority Over- sampling Technique. Journal of Artificial Intelligence Research, 16, pp.321-357.

He, H. and Garcia, E.A., 2009. Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering, 21(9), pp.1263-1284.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, pp.2825-2830.

Powers, D.M., 2011. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. Journal of Machine Learning Technologies, 2(1), pp.37-63.

Sun, Y., Wong, A.K.C. and Kamel, M.S., 2009. Classification of Imbalanced Data: A Review. International Journal of Pattern Recognition and Artificial Intelligence, 23(04), pp.687-719.

Zhang, C. and Ma, Y., 2012. Ensemble Machine Learning: Methods and Applications. Springer Science & Business Media.