

**Artificial Intelligence & Intelligent Systems**

**10204250  
K/618/5660**

**Section (4)**

**The foundations of AI and the development of an AI system – Part 2**

**Submitted to**

Dr. Rami Ibrahim

**Submitted on**

June 20th, 2023

**Submitted by**

Marwan Tareq Shafiq Al Farah

**Student ID**

21110011

**Spring 2022 - 2023**

**Table of Content**

[***Report*** 3](#_Toc137931097)

[**Introduction** 3](#_Toc137931098)

[**Materials** 3](#_Toc137931099)

[**** **Dataset’s Source/Description** 4](#_Toc137931100)

[**** **Method of Data Collection** 4](#_Toc137931101)

[**** **Dataset Attributes** 4](#_Toc137931102)

[**** **Dataset Size** 5](#_Toc137931103)

[**** **Usage of the Dataset** 5](#_Toc137931104)

[**** **Data Pre-processing** 5](#_Toc137931105)

[**** **Exploratory Data Analysis (EDA)** 6](#_Toc137931106)

[**Methods** 9](#_Toc137931107)

[ **The Choice and Justification of Models Used** 9](#_Toc137931108)

[ **Models’ Pipeline Charts** 11](#_Toc137931109)

[ **Technical Implementation of the Models** 12](#_Toc137931110)

[ **Leveraging Models for Enhance Organizational Performance** 16](#_Toc137931111)

[ **Measuring the Success of our Models** 17](#_Toc137931112)

[ **Enhancing Model Performance** 18](#_Toc137931113)

[**Results and Discussion** 19](#_Toc137931114)

[ **Results Analysis** 19](#_Toc137931115)

[ **Project Implementation in Jordan** 22](#_Toc137931116)

[ **Future Enhancements and Improvements** 23](#_Toc137931117)

[ **My Role in Project Building and Improvement** 25](#_Toc137931118)

[**Conclusion** 26](#_Toc137931119)

[***References*** 27](#_Toc137931120)

# ***Report***

**Introduction:**

In this age of information, high-speed communication, and digital technology, making decisions based on large sets of data is increasingly becoming a normal and accepted part in a large variety of sectors in today’s society. A key area where data analytics plays a significant role is in predicting individual income levels based on several socio-demographic factors. Accurate and reliable income forecasting can be a key asset for a wide range of stakeholders, including law-makers shaping tax reforms and social welfare policies, businesses that want to better understand their customer base, and non-profits organizations aiming to target aid where it is needed the most.

This project aims to solve the problem of income predictions of whether an individual’s yearly income is over $50,000 according to the U.S. census data. This problem is complex as income levels are influenced by plenty of factors such as age, education, occupation, marital status, etc... In order to tackle this problem, we created a machine learning model that predicts income using both Artificial Neural Networks (ANN) & Random Forest (RF) classifiers. By comparing the performance of these two techniques, we want to enhance the income predictions, optimize it, and provide a stronger and more reliable model.

Firstly, instead of limiting the analysis to a single machine learning approach, it provides an in-depth comparison of two different yet powerful techniques, RF and ANN, thus adding a comparative perspective to the problem of income prediction. Moreover, the use of an ensemble learning method offers a comprehensive approach to understand the strengths and weaknesses of these different types of models in handling complex predictive tasks.

As for its impact, a more accurate and reliable income prediction model could benefit a wide range of users and organizations. For instance, governments could refine their social programs and financial policies, marketers could target their products to specific audiences using these insights, and researchers could use this model as a foundation for their work on exploring socio-economic issues and income-inequality. As such, by enhancing the accuracy of income predictions, this project has the potential to make a significant impact on both organizational decision-making and societal understanding of income dynamics.

**Materials:** (1994 Census bureau database, 2016)

This section provides an overview of the project’s dataset. It contains details about the dataset’s origin and popularity, the method of data collection, dataset attributes, size, and usage, data pre-processing techniques, and the exploratory data analysis performed. This section provides a brief overview of the dataset and its preparation for analysis.

* **Dataset’s Source/Description:** The dataset used in this project, adult.csv, was obtained from Kaggle, a well-known data science and machine learning platform. It was obtained from the Kaggle dataset repository “UCI Machine Learning Repository: Adult Census Income” (<https://www.kaggle.com/datasets/uciml/adult-census-income>). The dataset was contributed by the UCI Machine Learning Repository and has gained popularity in the data science community due to its application to income prediction and socioeconomic analysis.
* **Method of Data Collection:** The data was collected within the adult.csv dataset was originally gathered from the 1994 Census Bureau database. It is a subsample of reasonably clean records extracted under specific conditions: individuals over the age of 16, with an annual income greater than $100, a final weight (fnlwgt) greater than one, and working more than 0 hours per week. This dataset, derived from the Census Bureau’s Current Population Survey (CPS), provides valuable insights into the socioeconomic characteristics of individuals in the United States.
* **Dataset Attributes:** The adult.csv dataset consists of 15 columns, each representing a different attribute related to people’s socioeconomic profiles and demographics. These characteristics provide a comprehensive view of the various factors that may influence a person’s income level.
* **Age:** Denotes the individual’s age.
* **Workclass:** Indicates the individual’s employment status or type, such as private, self-employed, government, and so on.
* **Final Weight (fnlwgt):** This is the weight assigned to each observation in the dataset, calculated to ensure that the population is accurately represented. This weight takes into account the sampling design and aids in drawing conclusions about the entire population.
* **Education:** The highest level of education attained by the individual, such as a high school diploma, a bachelor’s degree, and so on.
* **Education Level (education.num):** A numerical representation of a person’s educational attainment. This attribute gives a numerical value to education level, which makes it easier to use numerical data in machine learning algorithms.
* **Marital Status:** Indicates the individual’s marital status, such as married, divorced, widowed, etc.
* **Occupation:** Represents the individual’s occupation or job, such as managerial, technical, clerical, and so on.
* **Relationship:** A person’s role in the household or family unit, such as spouse, child, or own child.
* **Race:** Represents the individual’s race or ethnicity.
* **Gender (Sex):** Indicates the individual’s gender or sex.
* **Capital Gains:** Reflects the individual’s capital gains, which represent the increase in the value of assets or investments.
* **Capital Losses:** Represents the individual’s capital losses, indicating a decrease in the value of assets or investments.
* **Hours Worked per Week:** The number of hours worked per week by the individual.
* **Country of Origin (Native Country):** Represents the individual’s country of origin or nationality.
* **Income:** The target variable indicating whether or not an individual’s annual income exceeds $50,000. The classification or prediction task for income levels is determined by this attribute, which serves as the primary focus of the analysis.

These features provide a comprehensive picture of people’s socioeconomic profiles, demographics, and factors that influence their income levels. They serve as the foundation for comprehending the relationships between various variables and their impact on the income prediction task.

* **Dataset Size:** The adult.csv dataset has 32561 rows and 15 columns, making it a large and representative dataset for income forecasting. Each row represents a person’s data, and each column represents a specific attribute or characteristic.
* **Usage of the Dataset:** The adult.csv dataset has received a lot of attention and use in the data science community since it was made available on Kaggle. According to its Kaggle repository page, the dataset has been viewed 568,147 times and downloaded 49,818 times. This high level of participation demonstrates the dataset’s popularity and widespread adoption among data scientists, researchers, and practitioners.
* **Data Pre-processing:** Prior to model training and analysis, the dataset underwent several pre-processing steps to ensure data quality and compatibility with machine learning algorithms.

First, missing values denoted by “?” were replaced with NaN values to ensure consistency and ease of handling missing data. Rows with missing values were then removed from the dataset to eliminate incomplete or unreliable data.

The categorical variables were then encoded in order to convert them into numerical representations suitable for machine learning algorithms. Each categorical column was transformed using scikit-learn’s LabelEncoder, which assigned a unique numerical label to each distinct category in the column. This step allows machine learning models to effectively process and learn from categorical data.

To ensure fair comparison and efficient learning within machine learning models, all variables were standardized, with the exception of the target variable “income”. The columns are changed during standardization to have a mean of 0 and a standard deviation of 1. By normalizing the numerical features, the effects of different scales are less noticeable, and they become more comparable.

Additionally, an analysis of the ‘education’ column and its corresponding ‘education.num’ column was done during the pre-processing stage. Identifying any mismatched values between the two columns was the goal. It was discovered that every unique value in “education” was matched by the same unique value in “education.num” by creating a mapping between the unique values in “education” and their corresponding unique values in “education.num” using the groupby function. As a result, the “education” column was deemed unnecessary and removed from the dataset. By removing unnecessary data from the dataset, this step allowed machine learning models to only train and analyze on features that were actually relevant and not redundant.

Additionally, oversampling and train-test splitting were incorporated into the pre-processing pipeline to address class imbalance and assess the model’s performance. The imbalanced-learn library’s RandomOverSampler was used to oversample the minority class. Using a sampling strategy of “minority,” this technique generated synthetic samples to balance the distribution of the two classes.

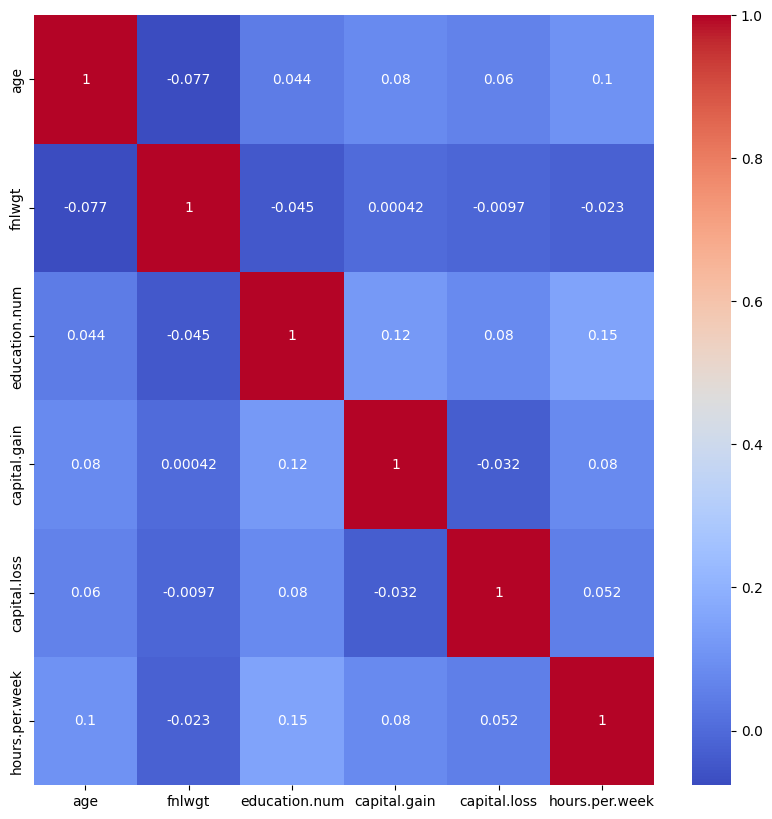
The train\_test\_split function from scikit-learn was then used to divide the dataset into training and testing sets. In the split, 30% of the data were designated for testing, and the remaining 70% were designated for training. The random\_state parameter was set to a different value in each iteration (the number of iterations) to ensure repeatability and consistency across iterations.

The fit\_resample method of the RandomOverSampler object was used to oversample the training set after each iteration. By creating synthetic samples for the minority class, this corrected the distribution of classes. The oversampled data was then used to update the X\_train and y\_train.

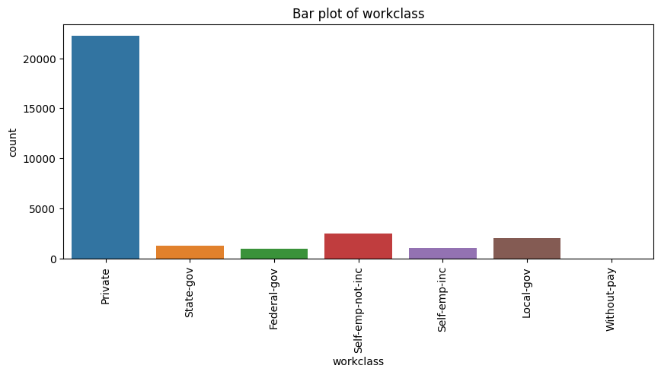
The dataset was prepared for effective machine learning analysis through these extensive pre-processing steps, which included missing value handling, categorical variable encoding, numeric variable standardization, redundant information elimination, oversampling, and train-test splitting. These steps ensured that the data was of high quality, that it was compatible with machine learning algorithms, that variables were compared fairly, and that class imbalance was addressed. The resulting pre-processed dataset served as a solid foundation for meaningful insights, accurate predictions, and robust model performance evaluation.

* **Exploratory Data Analysis (EDA):** The EDA performed on the adult census income dataset provided valuable insights into the socioeconomic characteristics and demographics of individuals related to income prediction. The analysis included a variety of visualizations, such as a correlation heatmap, histograms, bar plots, and a pairplot. While the correlation heatmap and bar plots revealed interesting patterns and distributions, the pairplot and histograms did not yield significant insights in this specific dataset.

The correlation heatmap revealed connections between numerical variables in the dataset. It found positive correlations between age, education level, and weekly hours worked with capital gain, implying that older people with higher education levels have higher capital gains and work more hours. These findings can be used to guide feature selection and modeling decisions.



The bar plots visualized categorical variable distributions and provided socioeconomic context. They revealed common categories like “Private” workclass, “HS-grad” education level, “Husband” relationship, “United States” native.country, “White” race, “Male” gender, and “Married-civ-spouse” marital status. These distributions shed light on the dataset’s composition and representation of various categories. But most importantly was the income bar plot, which showed us that the dataset isn’t evenly distributed as the class “<=50k” is much more present in the dataset in comparison to the class “>50k”, which could be problematic when predicting these values as the dataset is biased towards one value more than the other.

A picture containing text, screenshot, diagram, plot

Description automatically generated

A picture containing text, screenshot, diagram, plot

Description automatically generatedA picture containing text, screenshot, diagram, plot

Description automatically generated

A bar graph with different colored squares

Description automatically generated with low confidenceA picture containing text, screenshot, diagram, plot

Description automatically generated

A picture containing screenshot, text, diagram, rectangle

Description automatically generatedA picture containing text, screenshot, diagram, plot

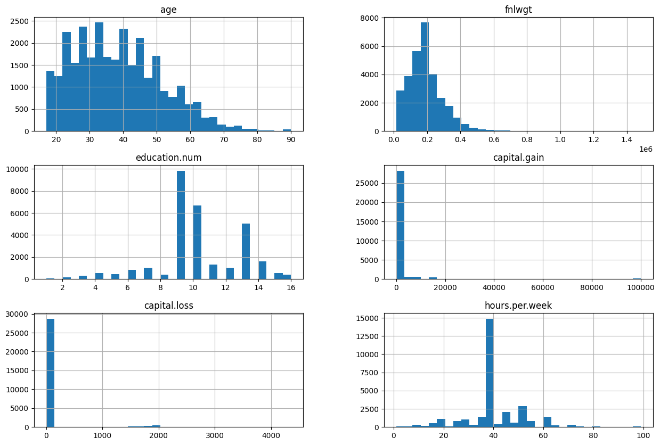
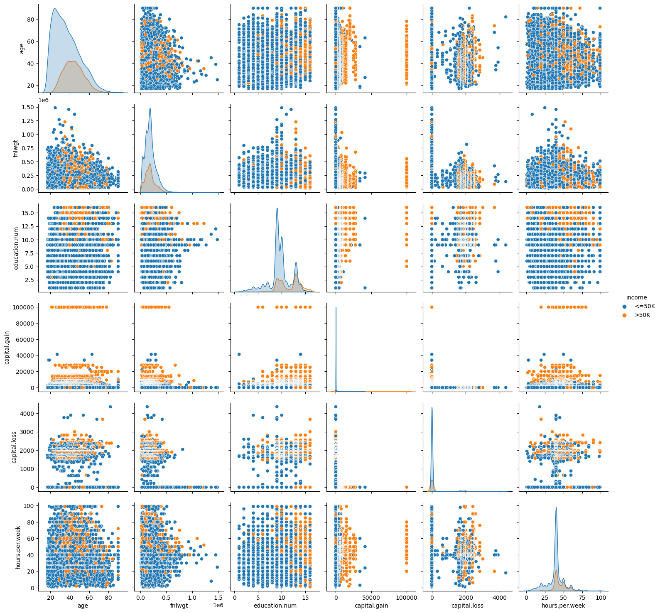
Description automatically generated

A picture containing text, screenshot, rectangle, diagram

Description automatically generatedA picture containing text, screenshot, plot, diagram

Description automatically generated

While the pairplot and histograms did not yield significant insights in this case, they are useful tools in EDA for investigating relationships, clusters, outliers, and data anomalies. These visualizations did not reveal any distinct patterns or clusters in the context of the adult.csv dataset. The focus should, however, be on the insights derived from the correlation heatmap and bar plots, which provided significant information about the dataset and its attributes.



Overall, the EDA was carried out on adults.csv dataset provided a solid foundation for further modeling and analysis. The correlation heatmap and bar plots revealed important attributes, relationships, and distributions that are critical for comprehending the dataset’s potential implications for income prediction. While the pairplot and histograms did not provide significant insights in this case, they are still useful tools for uncovering hidden patterns and anomalies in other datasets.

**Methods:**

* **The Choice and Justification of Models Used:**

The Random Forest classifier and the Artificial Neural Network (ANN) are the two modeling approaches we chosen for this project. Choosing these two models was based on our wish to explore the different features of machine learning, with regards to interpretability and complexity.

* **Artificial Neural Networks (ANN):**

Because of their ability to model complex, non-linear relationships between predictors and response variables, ANNs have grown in popularity. ANNs can capture subtle intricacies in data by constructing a network of interconnected “neurons” that is similar to the structure of the human brain and leveraging a vast number of parameters. As such, ANNs are a good fit to problems in which the relationship between predictors and the target variable is believed to be intricate and non-linear.

For that reason, in this project, we used an ANN with four layers (an input layer, two hidden layers, and an output layer). The network’s architecture was chosen to balance complexity and computational efficiency. The chosen model combines ‘relu’ and ‘sigmoid’ activation functions. The selection of these activation functions allows the model to capture non-linearity in the data while also producing output that is easily interpretable as probabilities (in the case of the ‘sigmoid’ function).

The Keras library was used to implement the ANN model, which allows for highly customizable model design and includes useful features such as model compilation and training, batch processing, and callbacks for tracking model performance.

* **Random Forest Classifier:**

The Random Forest classifier was chosen as the project’s second model. Random Forest is an effective ensemble method that combines the predictions of multiple decision trees to produce a final prediction. The model is well-known for its sturdiness and versatility, as it is suitable for both regression and classification tasks.

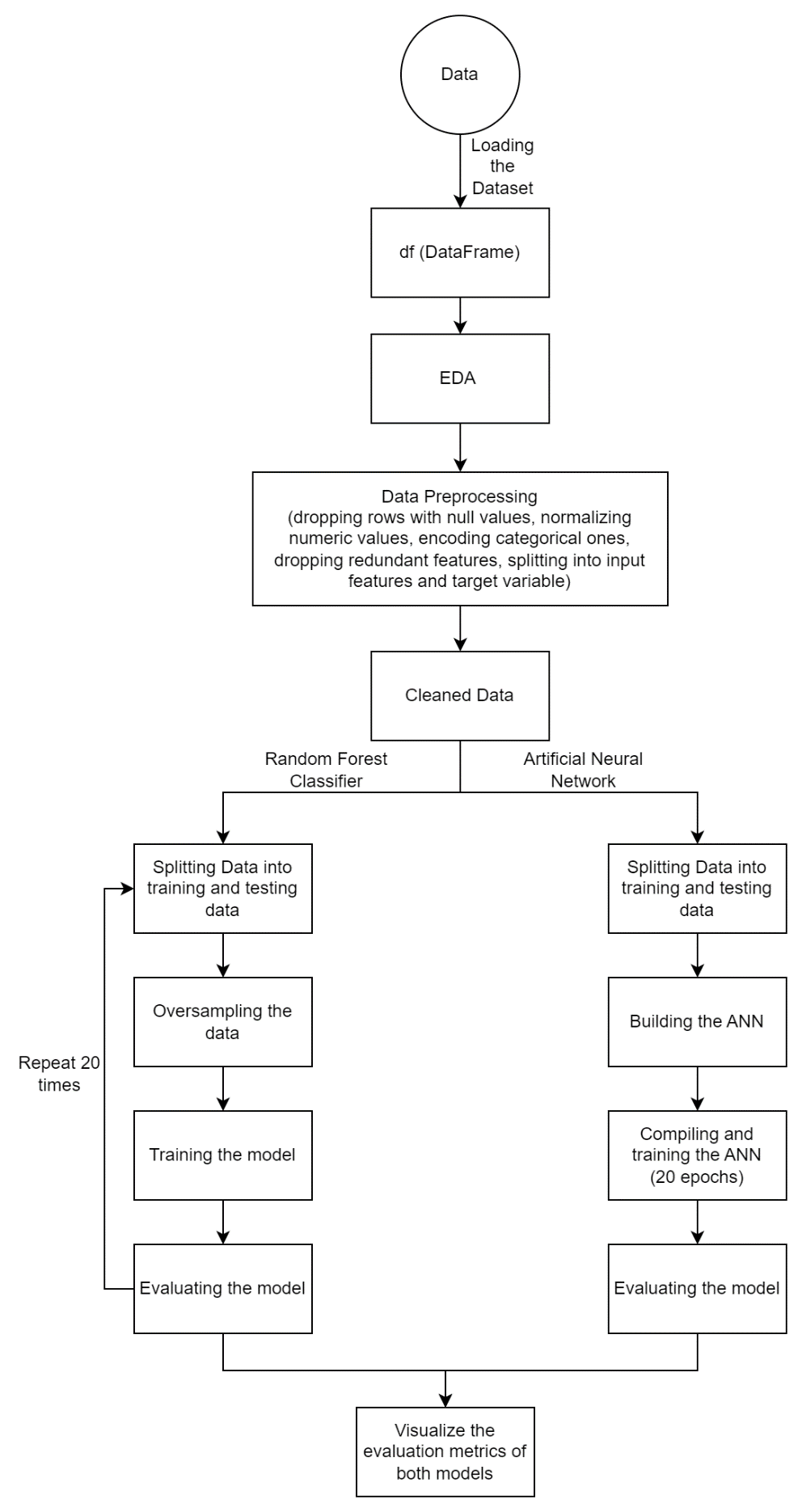
Random Forest has the advantage of providing a good balance between predictive performance and interpretability along with the fact that it avoids overfitting. Each tree in the forest can be visualized and interpreted, providing insights into the factors that influence predictions. In contrast, the ANN’s interpretability is more difficult due to the model’s complexity.

Furthermore, because of their ensemble nature, Random Forests are resistant to overfitting and can handle both numerical and categorical data without extensive preprocessing. Random Forest also has the advantage of returning feature importance measures, which can provide valuable insights into the model’s most influential predictors.

As for the Random Forest implementation, we used the scikit-learn library, which is an easy-to-use and well-rounded Python machine learning tool that provides a general scoop of operations such as model training, prediction, evaluation, and hyperparameter tuning.

Finally, the using of these two models allows for a comprehensive examination of our dataset. The ANN captures complex, non-linear relationships, whereas the Random Forest is a more interpretable, but still powerful, alternative.

* **Models’ Pipeline Charts:**

****

* **Technical Implementation of the Models:**

This section discusses the technical implementation of two models: the Artificial Neural Network (ANN) and the Random Forest. We will discuss data splitting, model architecture, compilation, training, evaluation, and storing metrics for the ANN model. Initialization, handling imbalanced datasets, training, evaluation, and storing metrics are all part of the Random Forest model implementation. These procedures ensure that models are well-designed, that training is effective, and that predictions are accurate.

* **Artificial Neural Network (ANN):**

The Artificial Neural Network (ANN) is a fundamental pillar of deep learning that can be used to perform a variety of complex tasks. In this section, we will go over the complete, step-by-step technical implementation of an ANN model using Python’s powerful Keras library for our project.

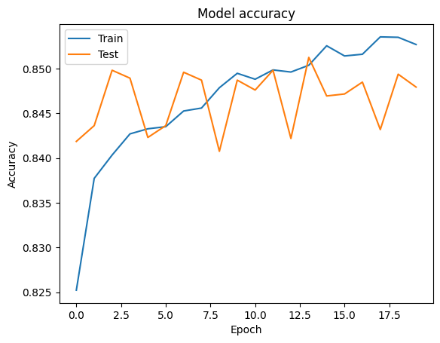
* **Data Splitting:** Before we begin designing the model, we must first divide our dataset into a training set and a test set. To accomplish this, we use the scikit-learn library’s train\_test\_split function. To evaluate our model on unseen data, we must set aside a portion of the data (in this case, 30%) as a test set (X\_test, y\_test). The remaining 70% is used for model training (X\_train, y\_train). This split ensures that the model can generalize and not just memorize the training data.
* **Model’s Architecture:** The architecture of the ANN model serves as the foundation for the learning that will take place. To begin the model design process, we create a Sequential model. The Sequential model is a linear stack of layers that can be added one by one until the network architecture is complete. This option provides an easy way to create models with exactly one input tensor and one output tensor for each layer.
* **Model’s Layers:** After the model’s initialization, we add several layers:

1. **The First Layer (The Input Layer**): The first layer is a dense (fully connected) layer with 64 neurons or nodes and a’relu’ (Rectified Linear Unit) activation function. The ‘input\_dim’ parameter corresponds to the number of features in our training dataset. It is only required for the first layer; subsequent layers can infer the input dimensions based on the previous layer. The activation function ‘relu’ adds nonlinearity to the model, allowing it to learn from complex patterns. ‘relu’ also helps to mitigate the vanishing gradients problem, which is encountered when training deep neural networks.
2. **The Second Layer (The First Hidden Layer):** Our network’s second layer is another dense layer, this time with 128 neurons. The ‘sigmoid’ activation function is used in this layer. A sigmoid function converts any input value into a range of 0 to 1, making it an excellent choice for binary classification problems and allowing the output to be interpreted as a probability.
3. **The Third Layer (The Second Hidden Layer):** The third layer is a dense layer with 32 neurons that use the ‘relu’ activation function, continuing the model’s pattern of alternating non-linearities.
4. **The Fourth Layer (The Output Layer):** Finally, we will add the output layer. Given that our task is a binary classification problem, this layer is dense with only one neuron. It, like the second layer, employs a’sigmoid’ activation function to compress outputs between 0 and 1.

* **Model Compilation:** Once the model’s architecture has been defined, the model must be compiled. Several critical components must be defined during the compilation:

1. **Loss Function:** For binary classification problems, the ‘binary\_crossentropy’ loss function is appropriate. This function computes the cross-entropy loss between true and predicted labels as a model performance metric.
2. **Optimizer:** The ‘adam’ optimizer is employed. Adam is an acronym for “Adaptive Moment Estimation,” a method for calculating adaptive learning rates for each parameter. In practice, it has been demonstrated to be effective across a wide range of deep learning models.
3. **Metrics:** During the training phase, we use ‘accuracy’ as the metric to monitor. The proportion of correct predictions over total predictions is calculated as accuracy.

* **Early Stopping:** We use the EarlyStopping callback to prevent overfitting by monitoring the model’s validation loss (‘val\_loss’). If the validation loss does not decrease for five consecutive ‘epochs’ (an epoch is an iteration over the entire dataset), training is terminated, and the model from the epoch with the lowest validation loss is retained.

**A picture containing text, diagram, plot, line

Description automatically generated**

* **Model Training:** The model is then trained on our training data using the fit function. The model learns to plot the inputs (features) to the outputs (target variable) over a number of epochs (20 in this case) and a batch size of 20. The number of samples generated in the network at a time is referred to as the batch size. The PredictionHistory callback records the model’s predictions on the test set at the end of each epoch during the training process. This generates a trajectory that shows how the model’s predictions changed during training.
* **Model Evaluation:** The evaluation phase begins after training. During this stage, we first convert the recorded predictions into an array and reshape them for compatibility. We then compute four key metrics for each epoch’s predictions: accuracy, precision, recall, and F1 score. These metrics, computed on the test set, assist us in understanding how well our model performs on unknown data, providing insights into its predictive power and reliability.
* **Storing Metrics:** The metrics calculated for each iteration are saved in the arrANN array for later analysis. This array can then be used to generate statistics such as mean and standard deviation, as well as plots for visual evaluation of the model’s performance.

To summarize, from data splitting and model architecture design to compilation, training, and evaluation, the technical implementation of our ANN model requires careful planning and strategic design decisions at every stage. The entire process is designed to create a model that learns quickly and generalizes well to new data, ultimately assisting us in meeting our project’s goal of accurately predicting income categories.

* **Random Forest:**

Random Forest is a versatile and widely used machine learning model known for its high predictive accuracy, resistance to overfitting, and ability to handle a large number of input variables. It works by constructing a large number of decision trees and then displaying the class that is the mode of the classes (classification) or the mean prediction (regression) of the individual trees. In this section, we’ll go over the Random Forest Classification model’s technical implementation step by step.

* **Model Initialization:** The Random Forest Classifier from the scikit-learn library is initialized first. We create a new Random Forest model with default parameters besides the number of estimators as it was set to 20, by calling RandomForestClassifier(n\_estimators = 20). Within the parentheses, we can fine-tune various hyperparameters such as the number of trees (n\_estimators), the maximum depth of the tree (max\_depth), and others if necessary.
* **Oversampling:** We then instantiate the RandomOverSampler class from the imbalanced-learn library, which is designed to handle imbalanced datasets. In many real-world scenarios, there is often a significant disparity in the number of observations in each class, which can lead to biases in the model. If we didn’t employ the RandomOverSampler, our model may overemphasize the majority class (<= 50k) while ignoring the minority class (> 50k), therefore, the RandomOverSampler is configured with the ‘minority’ strategy in this project, indicating that we want to resample the minority class to have an equal number of samples as the majority class. This class balance ensures that our model does not overlook the minority class.
* **Training and Evaluation:** Once the Random Forest classifier and oversampler are ready, we start a loop to perform multiple training and evaluation iterations. The variable n\_iters stores the number of iterations (20 iterations). We did the following for each iteration:

1. **Splitting the Data:** Using the train\_test\_split function, we divided the entire dataset into a training set and a test set. The test size is set at 30%, which means that 30% of the data is reserved for testing the model’s performance, while the remaining 70% is used for training. To ensure that the split is reproducible, the random state is set to the current iteration number.
2. **Oversampling:** The fit\_resample function is then called on the training data to perform minority class over-sampling.
3. **Train the Random Forest classifier:** Next, we use the Random Forest classifier’s fit method, passing in the oversampled training data. This method trains the model the relationship between input features (X\_train) and the target variable (y\_train).
4. **Make predictions and evaluate:** Using the trained model, we make predictions on the test data. For each iteration, the predicted values are compared to the actual values to compute the accuracy, precision, recall, and F1 score. These metrics provide detailed information about the model’s performance.

* **Storing Metrics:** The metrics calculated for each iteration are saved in the arrRF array for later analysis. This array can then be used to generate statistics such as mean and standard deviation, as well as plots for visual evaluation of the model’s performance.

To summarize, the technical implementation of the Random Forest model is a deliberate process that carefully handles the issue of data imbalance, efficiently trains the model, and thoroughly evaluates its performance across multiple iterations. The process is designed to produce a model with high predictive accuracy and excellent generalization capabilities.

In conclusion, this part gives a thorough explanation of the technological implementation of two models, the Random Forest and the Artificial Neural Network (ANN). The procedures described here, such as data splitting, model architecture design, compilation, training, assessment, and storing metrics, guarantee that the models are reliable, well-trained, and able to predict outcomes accurately. By following these steps, we can create models that effectively learn from data, generalize well to previously unseen instances, and aid in the achievement of project goals.

* **Leveraging Models for Enhance Organizational Performance:**

The Artificial Neural Network (ANN) and Random Forest (RF) classifiers used in this project to predict individual income levels are not isolated entities. When these models are used in tandem, they can provide a more robust and comprehensive predictive capability. This is due to the ensemble learning principle, which states that a combination of learning models can often outperform individual models.

Based on its deep learning principles, the ANN excels at capturing complex, nonlinear relationships within data. RF, on the other hand, is known for its ability to handle overfitting and provide estimates of feature importance. When both of these models work in tandem, the organization benefits from the strengths of both models, resulting in more accurate income predictions.

Incorporating income predictions from both models into strategic planning can provide a more complete picture of the target demographic. Because it incorporates insights from both the ANN and RF models, incorporating income predictions into customer segmentation strategies can improve personalized marketing approaches.

In terms of resource allocation, the ANN and RF models’ synergistic operation can help organizations identify key demographics based on a more accurate prediction of income levels. As a result, resources are optimally allocated to the most profitable market segments, enhancing the effectiveness of marketing and sales initiatives.

Additionally, incorporating these models into existing Customer Relationship Management (CRM) systems can enrich CRM data. With the combined predictive power of ANN and RF, the system can not only track customer interactions but also more accurately predict their income levels.

Combining these models in socioeconomic analysis can result in more insightful policy development for government organizations. In the financial services industry, the combination of ANN and RF models can improve decision-making processes, improve creditworthiness assessments, and tailor insurance policies.

It is crucial to keep in mind that effective use of these models necessitates both a strong data infrastructure and a clear grasp of the objectives of the company. While the models are quite accurate on their own, they may produce even more accurate forecasts when paired with human judgment and taking the greater business context into account.

Organizations can gain a competitive advantage, make more informed decisions, and ultimately drive growth and success by strategically integrating and leveraging the collaborative strengths of these models.

* **Measuring the Success of our Models:** (Koo, 2018; Harikrishnan, 2019; Tigerschiold, 2019)

We used four key metrics to assess the performance of our models, namely the Artificial Neural Network (ANN) and Random Forest (RF) Classifiers: accuracy, precision, recall, and F1 score. Each of these metrics provides a unique perspective on the model’s performance, allowing us to evaluate its dependability from various perspectives. Here’s a detailed breakdown of each metric and the reasoning behind its selection:

**Accuracy:** It is the ratio of accurate (both positive and negative) predicts to all predictions. Mathematically, it may be expressed as (True Positives + True Negatives) / (True Positives + False Positives + False Negatives + False Negatives). Regardless of the exact result, accuracy gives us a rough indication of how frequently the model is accurate.

**Precision:** Precision measures the accuracy of the model’s positive predictions. It is defined as the proportion of true positive predictions to all positive predictions (True Positives / (True Positives + False Positives)). Precision is an important metric when the cost of a false positive is high. For example, if we are targeting high-income individuals with a luxury marketing campaign, we want to be extremely confident in our positive (high-income) predictions, so precision is an important metric.

**Recall:** The model’s ability to correctly detect positive instances is measured by recall, also known as sensitivity or true positive rate. It is defined as the proportion of true positive predictions to all actual positives (True Positives / (True Positives + False Negatives)). When the cost of missing a positive instance is high, recall becomes an important metric. For example, if a bank uses this model to predict who will default on a loan (a ‘positive’ prediction in this context), a false negative (failure to predict a default) could be costly, emphasizing the importance of high recall.

**F1 Score:** The harmonic mean of precision and recall is used to calculate the F1 score. It strikes a balance between the two metrics, rewarding models with comparable precision and recall. It is especially useful when the data distribution is unbalanced. For example, if we are dealing with a small percentage of high-income people in a large population, the F1 score would be a more relevant metric than accuracy.

We can ensure a thorough evaluation of our models by employing these four metrics. Accuracy provides a quick summary of overall performance, whereas precision, recall, and the F1 score provide more nuanced views of the model’s performance, which is especially important given that income prediction frequently involves imbalanced datasets.

* **Enhancing Model Performance:**

We used an Artificial Neural Network (ANN) and a Random Forest Classifier (RF) to implement our predictive models for the adult census income dataset. We made several changes based on iterative testing and evaluation to improve and optimize these models’ performance. This procedure, which we will describe in detail below, enabled us to significantly improve our model performance in terms of accuracy, precision, recall, and F1 score.

* **Optimization of an Artificial Neural Network (ANN):**

ANNs are complex models that necessitate extensive tuning and experimentation. Several strategies were implemented to improve our ANN model, with a focus on refining the network architecture and employing appropriate optimization techniques:

* **Network Architecture:** To optimize our ANN, we experimented with different configurations of hidden layers and neurons within each layer. We eventually settled on a model with two hidden layers: the input layer contains 64 neurons, the first hidden layer contains 128 neurons, and the second hidden layer contains 32 neurons. These layers employ the activation functions relu and sigmoid to improve the network’s learning capability.
* **Number of Epochs:** has a significant impact on the performance of an ANN. We were able to see how the model learned over time by training it for 20 epochs at first. Underfitting can occur when there are too few epochs, while overfitting occurs when there are too many.
* **Batch Size:** Another important factor in ANN optimization is batch size. We discovered that a batch size of 20 produced favorable results. A batch size of 20 means that the model processed 20 training examples before updating the weights during each training iteration. This batch size strikes a good balance between computational efficiency and convergence speed, allowing the model to generalize well without consuming too much memory.
* **Early Stopping:** We used an early stopping mechanism to prevent overfitting and unnecessary computation. This technique monitors a specified metric, in our case, the validation loss, and stops the training process if this metric does not improve after a certain number of epochs. We limited our patience to five epochs.

After conducting all of the above optimization strategies, it led to a notable increase in all evaluation metrics, as it increased the ANN’s accuracy from around 0.763 to 0.840, along with an increase in the model’s precision, recall, and f1 score.

* **Optimization of Random Forest Classifiers (RF)**

We used the following strategies to improve the RF performance:

* **Oversampling:** We used an oversampling strategy to balance the dataset due to the imbalance in the classes (>50k, <=50k) within our target variable. We specifically used RandomOverSampler to increase the representation of the minority class, thereby improving the model’s ability to predict less frequent outcomes.
* **Multiple Training Sets:** To ensure the robustness of our RF model, we trained it with 20 different train-test splits. This procedure exposes the model to a wide range of data scenarios, reducing the risk of overfitting to a specific data partition.
* **Number of Estimators:** We decreased the number of estimators of the model from the default value of 100 to 20, thus making the running time of the model shorter, and without affecting the model’s performance significantly.

After conducting all of the above optimization strategies, it led to an increase in all evaluation metrics, as it increased the RF’s accuracy from around 0.818 to 0.849, along with an increase in the model’s precision, recall, and f1 score. While it didn’t make a significant difference, but it did enhance the model’s overall performance.

We were able to significantly improve the performance of our ANN and RF models by implementing these strategies and monitoring how these strategies affected the models’ accuracy, precision, recall, and f1 score. Our process emphasizes the significance of ongoing model refinement as well as the value of rigorous testing and evaluation protocols. We optimized our models using these metrics to provide accurate and reliable income predictions based on the adult census dataset.

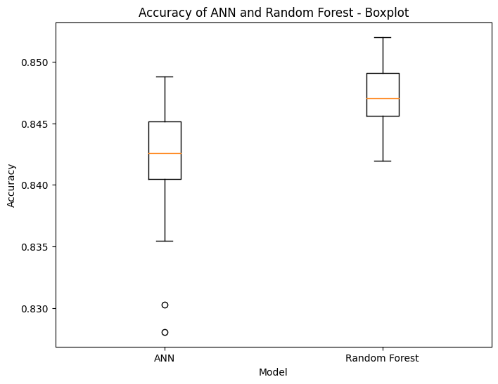
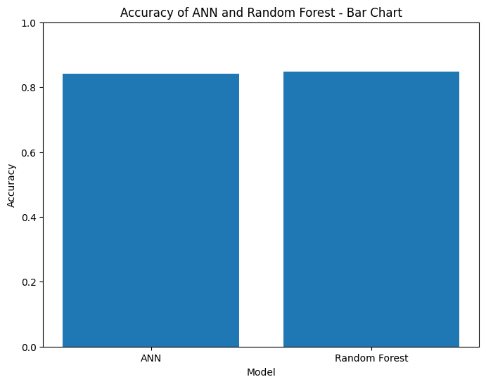
**Results and Discussion:**

This sections provides a thorough examination of the performance of the Random Forest Classifier (RF) and Artificial Neural Network (ANN) models on the Adult Census dataset. The results are evaluated using bar charts and box plots and visualized using accuracy, precision, recall, and F1 score. The section delves deeper into the project’s implementation in Jordan, taking into account contextual differences, dataset limitations, and the need for a localized approach. Future enhancements and improvements are also discussed, with strategies for collecting localized data, model adaptation, advanced feature engineering, and continuous evaluation proposed to address the limitations encountered.

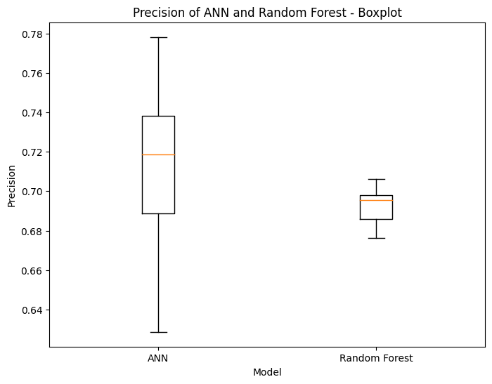
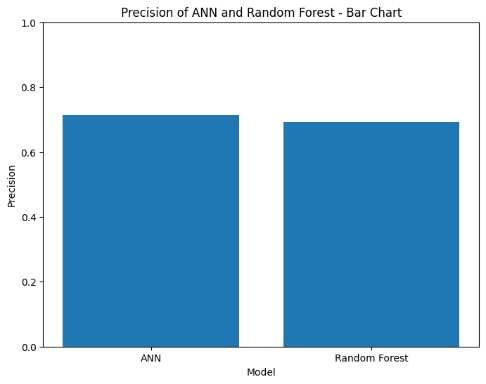
* **Results Analysis:**

This section provides a thorough examination of the Random Forest Classifier (RF) and Artificial Neural Network (ANN) models’ performance on the Adult Census dataset. The assessment is based on four key performance metrics: accuracy, precision, recall, and F1 Score. To highlight the comparative performance of the two models, the results are visualized using bar charts and box plots.

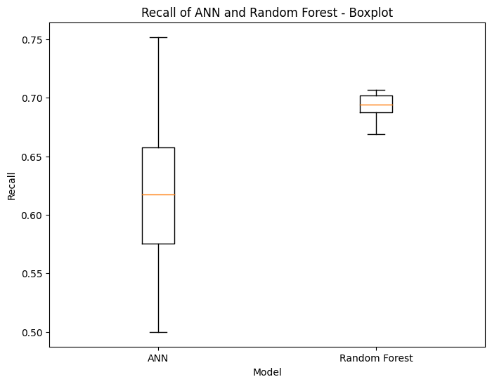
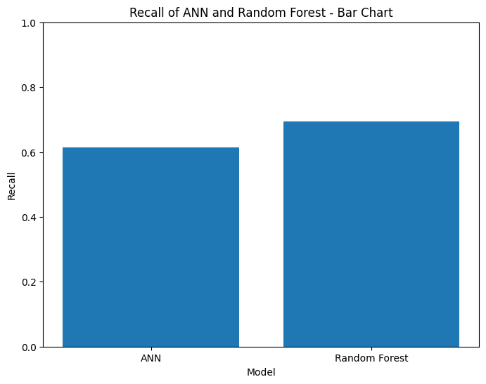
**Accuracy:** On the Adult Census dataset, both RF and ANN achieved high accuracy scores. When compared to ANN, RF had a slightly higher mean accuracy 0.849, while the ANN had a mean accuracy of 0.840. The bar chart shows the mean accuracy for each model, highlighting the slight advantage of RF over ANN. The box plot analysis adds more information, revealing that the distribution of accuracy scores for RF is narrower than that of ANN, implying more consistent performance across different data samples.



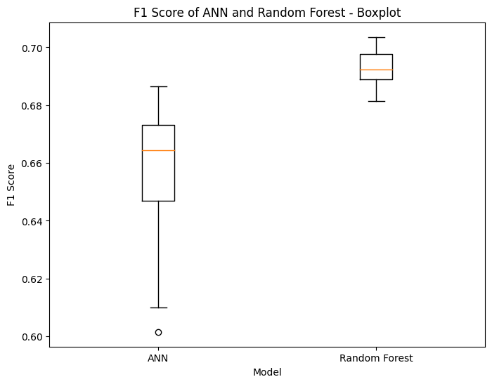
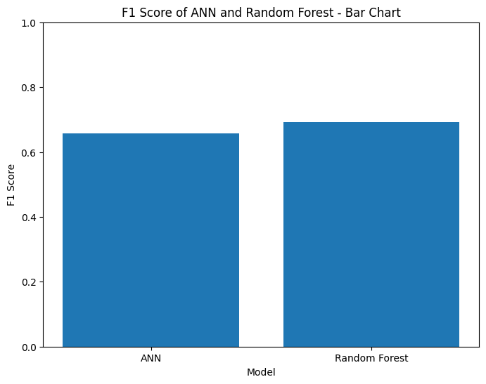
**Precision:** Precision measures the models’ ability to correctly classify positive instances. Precision scores were comparable for RF and ANN. The mean precision of RF was 0.696, while the mean precision of ANN was slightly higher at 0.710. The bar chart compares the mean precision of the models quickly, highlighting the slight advantage of ANN over RF. The box plots show a more detailed picture, indicating that RF has a more concentrated distribution of precision scores, implying consistent performance across different data subsets. ANN, on the other hand, has a wider spread, implying that precision varies depending on the subset of data.



**Recall:** Recall assesses the models’ ability to capture positive instances from the dataset. The mean recall scores of RF and ANN are comparable. The mean recall for RF was 0.699, while the mean recall for ANN was slightly higher at 0.707. The bar chart clearly displays the mean recall values, demonstrating ANN’s marginal advantage over RF. According to the box plot analysis, RF has a narrower distribution of recall scores, indicating more consistent performance across different data subsets. ANN, on the other hand, has a wider distribution, implying that recall varies depending on the subset of data.



**F1 Score:** The F1 Score is a measure of overall model performance that combines precision and recall into a single metric. The mean F1 scores for RF and ANN are comparable. The average F1 score for RF was 0.700, while the average F1 score for ANN was 0.712. The bar chart allows for a clear comparison of the mean F1 scores, emphasizing ANN’s slight advantage over RF. The box plots depict the distributions in greater detail, indicating that RF has a more concentrated distribution of F1 scores, implying consistent performance across different subsets of data. ANN, on the other hand, has a wider spread, indicating variations in F1 scores for different subsets of the data.



Several findings emerge from a comparison of RF and ANN on the Adult Census dataset. Both models perform well across all metrics, indicating their effectiveness in the classification task. RF has a slight advantage in terms of mean accuracy, precision, recall, and F1 score. The visualizations, which include bar charts and box plots, provide a clear and concise representation of the results, assisting in the interpretation and understanding of the comparative performance.

Furthermore, the box plots provide additional insight into the consistency of the models’ performance across different data subsets. RF has narrower distributions for accuracy, precision, recall, and F1 score, indicating more consistent performance across different data samples. ANN, on the other hand, has wider spreads, indicating that performance varies depending on the subset of data.

These findings imply that RF may be better suited for producing consistent and reliable results on the Adult Census dataset. However, other factors such as computational complexity, interpretability, and feature importance must be considered when selecting the best model for the given task.

In conclusion, both RF and ANN perform well on the Adult Census dataset, with RF slightly outperforming ANN in terms of accuracy, precision, recall, and F1 score. The visualizations provide valuable insights into the models’ comparative performance, making it easier to choose the best model for similar classification tasks.

* **Project Implementation in Jordan:**

The adult census dataset used in this project primarily represents data collected from US citizens in 1994. As a result, it is important to give the possibility of completing this project in Jordan significant thought as economic inequalities, cultural distinctions, and the chronological distance between the dataset and the present (2023) are a few things that need to be taken into account.

**Data Relevance and Contextual Differences:** When implementing the project in Jordan, the primary concern is the dataset’s relevance. The US and Jordan have very different economic conditions, labor market dynamics, and sociocultural factors. As a result, the predictive model trained on the US dataset may fail to capture the unique characteristics of the Jordanian context.

**Limitations of the US Dataset:**Due to the dataset’s US-centric focus, which might make it less representative of Jordan’s population, it has several disadvantages as there may be differences between the two nations in terms of the economy, economic distribution, level of education, and other relevant aspects. As a result, applying the model directly to Jordan can produce unreliable results and erroneous forecasts.

**Time Difference:** The amount of time that has passed between the dataset was collected in 1994 and the present is a crucial additional component to take into account. Both the economic, social, and demographic environments in the US and Jordan have seen significant change over time. These transformations can have an impact on income patterns, potentially reducing the model’s applicability to the current situation.

**Adaptation and a Localized Approach:** A localized approach is required to address these challenges. It would entail adapting the model to Jordan’s specific context by incorporating relevant data, updating variables, and taking into account the country’s unique economic and social dynamics. Collecting up-to-date Jordanian population data is critical for developing a more accurate and reliable income prediction model.

**Evaluation Metrics and Practical Applicability:** While the models’ evaluation metrics received high scores, it is important to note that these metrics were based on performance in the US context. The models’ high accuracy, precision, recall, and F1 scores reflect their ability to predict income levels for the US population. However, due to the aforementioned disparities, it does not guarantee the same level of performance in the Jordanian context.

**Collaboration and Stakeholder Engagement:** Collaboration with local experts, policymakers, and stakeholders is instrumental in implementing the project in Jordan and collecting the correct data required for this project. Their expertise, insights, and understanding of the local context can inform the adaptation process, ensuring that the model is attuned to the specific needs and challenges faced by Jordan. Collaborative efforts can refine the model’s predictions and enhance its applicability.

Given the limitations of the US dataset, contextual disparities, time gap, and the need for a localized approach, implementing the project in Jordan requires careful thought. To create an accurate and reliable income prediction model, relevant and up-to-date Jordanian data must be gathered. Collaboration with local experts and stakeholders is critical for adapting the model and ensuring alignment with Jordan’s distinct characteristics and challenges.

It may be possible to develop a contextually relevant income prediction model for Jordan by addressing these considerations, conducting extensive research, and leveraging localized expertise. However, extensive adaptation, data collection, and stakeholder engagement are required to ensure the model’s practical applicability and reliability in the Jordanian context.

Once the challenges have been addressed, the project’s implementation in Jordan could provide significant benefits. It would allow for more accurate income forecasting, which would aid in policymaking and economic planning. Furthermore, it would promote data-driven decision making and strengthen local data analysis capacities. While assisting in socioeconomic development, this initiative would also boost Jordan’s knowledge economy and strengthen stakeholder networks.

* **Future Enhancements and Improvements:**

This section discusses the limitations encountered during project implementation and looks into potential areas for future improvements. We can ensure the project’s continuous development and refinement by acknowledging these limitations and considering future improvements.

* **Limitations:**
  + - **Dataset Relevance:** The project’s reliance on the adult census dataset, which primarily represents data from US citizens, is one of its primary limitations. This limitation limits the models’ generalizability to other contexts, such as Jordan. The economic disparities, cultural differences, and time differences between the United States and Jordan make direct application of the models to the Jordanian population difficult.
    - **Time Gap:** Because the adult census dataset was collected in 1994, there is a time lag between the training data and the present. Socioeconomic conditions, job markets, and income patterns have undoubtedly changed over time, potentially making the model’s predictions less accurate in the current environment.
    - **Data Availability:** The availability and quality of data are critical factors influencing the success of any machine learning project. In the case of this project, data availability constraints may have influenced the accuracy and reliability of the models. Obtaining more detailed and up-to-date data on the target population could improve the project’s outcomes.
* **Future Enhancements:**
* **Collecting Localized Data:** To address dataset limitations, future enhancements could include collecting localized data that is specifically tailored to the Jordanian context. This would entail gathering detailed socioeconomic data, taking cultural factors into account, and ensuring the representation of the relevant features and attributes that contribute to Jordan’s income levels.
* **Feature Engineering:** Investigating additional features and conducting extensive feature engineering could improve the predictive capabilities of the models. Incorporating domain knowledge and identifying new variables that have a significant impact on income levels in Jordan can improve the models’ accuracy and robustness.
* **Model Adaptation:** Retraining models with localized datasets and incorporating domain-specific knowledge can lead to more accurate predictions in the Jordanian context. Fine-tuning the model architectures, optimizing hyperparameters, and experimenting with different algorithms tailored specifically for the Jordanian population may improve performance.
* **External Factors:** Taking into account external factors that affect Jordanian income levels, such as macroeconomic indicators, policy changes, and social dynamics, could help to improve the project. Incorporating these factors as extra features or developing separate models to capture their influence can improve prediction accuracy and relevance.
* **Continuous Evaluation and Updating:** The models must be evaluated and updated on a regular basis to ensure their continued relevance and accuracy. Monitoring model performance with new data, incorporating feedback from users and stakeholders, and implementing a feedback loop can aid in identifying areas for improvement and driving continuous improvements.

While the project has provided valuable insights and preliminary predictions, it is critical to acknowledge the limitations and plan for future improvements. Addressing constraints such as dataset relevance, time gap, and data availability, as well as implementing future enhancements such as localized data collection, feature engineering, model adaptation, external factor consideration, and continuous evaluation, can lead to more robust and accurate predictions tailored to the Jordanian context. By implementing these changes, the project will be able to evolve and provide more meaningful and reliable insights into Jordanian income forecasting.

* **My Role in Project Building and Improvement:**

In this section, I will describe my role in the project’s building and improvement, as well as outline specific plans and future directions for the project’s enhancement.

* **Building of the Project:**

My role as a key contributor to this project included several critical aspects:

* + **Data** **Preprocessing:** I was in charge of data preprocessing tasks such as missing value handling, variable transformation and standardization, and exploratory data analysis. This step ensured the quality of the dataset and prepared it for model training.
  + **Model Development:** I was heavily involved in the development of the machine learning models. Selecting appropriate algorithms, designing model architectures, compiling and training the models, and evaluating their performance using relevant metrics were all part of the process.
  + **Evaluation and Analysis:** I evaluated the models thoroughly, analyzing their performance using metrics such as accuracy, precision, recall, and F1 score. This analysis revealed the models’ strengths and weaknesses, guiding decision-making regarding model selection and improvement strategies.
* **Project Enhancement:**

Moving forward, several specific plans and future directions are being developed to improve the project:

* + **Jordanian-Specific Dataset Collection:** Recognizing the limitations of the current dataset, a critical step in improving the project is to collect a more representative and localized dataset. This would entail gathering data that reflects the target population’s economic, social, and demographic characteristics, allowing for more accurate and contextually relevant predictions.
  + **Model Adaptation:** Using the localized dataset, I intend to adapt and fine-tune the models to address the limitations of their applicability to the Jordanian context. Retraining the models and incorporating domain-specific knowledge will be required in order to improve their predictive capabilities and ensure their relevance to Jordan’s economic landscape.
  + **Advanced Feature Engineering:** I plan to investigate more advanced feature engineering techniques in order to capture additional nuances and factors that influence Jordanian income levels. This could include incorporating external data sources, utilizing natural language processing techniques, or investigating advanced feature selection methods to improve the accuracy and predictive power of the models.
  + **Model Optimization and Ensemble Techniques:** I intend to optimize hyperparameters, fine-tune model architectures, and investigate ensemble techniques such as model stacking and boosting to improve model performance. These approaches can improve the robustness of the models, reduce overfitting, and improve their ability to handle complex data relationships.
  + **Continuous Model Evaluation and Updates:** A framework for continuous model evaluation and updates must be established. This includes tracking the performance of the models, incorporating new data to track their accuracy and relevance, and implementing a feedback loop that takes into account user feedback and stakeholder insights. This iterative approach will ensure that the project is constantly improved and adjusted to changing conditions.

My contributions to the development and enhancement of this project included data preprocessing, model development, feature engineering, and evaluation. In the future, I intend to improve the project by gathering a localized dataset, adapting the models, utilizing advanced feature engineering techniques, optimizing the models, and establishing a framework for continuous evaluation and updates. These steps will help to improve the project’s accuracy, relevance, and applicability in the Jordanian context, resulting in more valuable insights into income prediction.

**Conclusion:**

In conclusion, this report provided a thorough examination of income prediction on the Adult Census dataset using the Random Forest Classifier (RF) and Artificial Neural Network (ANN) models. The dataset, preprocessing steps, and model development process were all described in the material and methods section. The section on results and discussion thoroughly examined the performance of RF and ANN based on accuracy, precision, recall, and F1 score, with visualizations assisting in the comparison. The project’s implementation in Jordan was discussed, emphasizing the importance of a tailored approach and data collection tailored to the Jordanian context. Localized data collection, model adaptation, advanced feature engineering, and continuous evaluation were proposed as future enhancements and improvements. Overall, this report provides useful insights into income prediction, emphasizing the importance of contextual considerations and the need for ongoing refinement to improve the models’ accuracy and applicability.

# ***References***

1994 Census bureau database (2016) *Adult Census Income | Kaggle, Kaggle.Com.* Available at: <https://www.kaggle.com/c/adult-census-income>

Harikrishnan, N.B. (2019) *Confusion Matrix, Accuracy, Precision, Recall, F1 Score | by Harikrishnan N B | Analytics Vidhya | Medium, Medium.* Available at: <https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd>

Koo, P.S. (2018) *Accuracy, Precision, Recall or F1? | by Koo Ping Shung | Towards Data Science, Towards Data Science.* Available at: <https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9>

Tigerschiold, T. (2019) *What is Accuracy, Precision, Recall and F1 Score?, Labelf.* Available at: <https://www.labelf.ai/blog/what-is-accuracy-precision-recall-and-f1-score>