Machine Learning Project Report: Predictive Analytics On Adults Income - Vian Sebastian Bromokusumo

Project Domain

Background

According to a survey by the Central Statistics Agency of Indonesia published on November 6, 2023, the unemployment rate in Indonesia is 5.32 percent [1].Although this is a decrease from previous years, it remains an issue that needs to be resolved. The results of the study by Pramudjasi and Juliansyah, 2019 in the FEB Unmul Journal [2] stated that the population has a significant positive effect on the unemployment rate. This is also confirmed in the research of Sari and Pangestuty, 2022 in [3] stating that population growth has an impact on the increase in the Open Unemployment Rate. Various surveys and findings from this academic journal are the background for the Author in trying to analyze what factors have a strong relationship to unemployment, employment, especially factors that are related to high income.

This project focuses on the analysis of the "Adults Income" dataset with the aim of identifying and predicting factors that have a high correlation, or even contribute to employment, especially the amount of income of a person. In this case, the target feature of this dataset is whether a person's annual income is <50,000 dollars or >= 50,000 dollars per year, so this project will solve the classification case. Although this dataset is taken from the 1996 census data, this dataset still has significant relevance today, due to the general nature of the variables in this dataset and the absence of drastic changes in the components that determine incomedanya perubahan yang drastis pada komponen-komponen penentu penghasilan.

According to Jepchumba from Microsoft [4], *machine learning* , is a technique that uses advanced mathematics and statistics to recognize patterns in data that do not exist explicitly, and can predict according to the results of these patterns. With the variety of factors (variables) involved in this project, machine learning is the best solution. Identification and prediction of factors will be done by applying data analysis techniques such as Exploratory Data Analysis (EDA) and using Machine Learning algorithms such as Random Forest, K-Nearest Neighbors, and Boosting .

This project is a small tool to help solve the problem of difficulty in finding work, by analyzing the dynamics of high annual income factors. The results of this project are expected to help the Government and individuals in productive age as additional insight related to this problem, and help these parties to develop society and themselves to improve the quality of life.

Business Understanding

Stakeholder dan sasaran:

1. Government As the highest level organization in a country, the government can make good policies and changes, in order to improve the quality of life of its people. One way is to create/improve the system in the country to encourage the advancement of its human resources.
2. Individual At the individual level, it is hoped that the results of this project will provide insight into important factors that can improve the quality of life, through work and employment with high annual

incomes.

Problem Statements

1. Of the various features, which has the most influence on income?
2. With certain characteristics, can income be predicted?

*income refers to annual income.

Predictive Modelling Goals

1. Knowing the features that have a high relationship to income.
2. Able to predict income with an accuracy above 90%.

Solution Statements (Metodologi)

1. The target feature in this dataset is a boolean variable between >50k and <= 50k, so this is a Classification prediction case.
2. Perform Exploratory Data Analysis to obtain meaningful information in the data and understand the dynamics of the features.
3. Testing the differences between missing values handling techniques, and their impact on the accuracy of machine learning models.
4. Create a machine learning model that can predict income with an accuracy above 90%
5. Using Accuracy, Precision, Recall, F1-Score, and Confusion Matrix to evaluate model performance.

Data Understanding

Dataset: https://archive.ics.uci.edu/dataset/2/adult

Dataset Overview

The description of the variables in the dataset is as follows:

1. age (int64) : sample age
2. workclass (object) : sample job status
3. fnlwgt (int64) : sample ID
4. education (object) : last education sample
5. educational-num (int64) : sample's last education in numbers
6. marital-status (object) : sample marital status
7. occupation (object) : sample job/industry
8. relationship (object) : sample relationship status
9. race (object) : sample race
10. gender (object) : sample gender
11. capital-gain (int64) : capital gain
12. capital-loss (int64) : capital loss
13. hours-per-week (int64) : sample working hours in a week
14. native-country (object) : country of origin/grandfather of the sample
15. income (object) : sample income

Detailed description of this dataset is as follows.

1. There are 48,842 data samples
2. There are two different data types, namely 'object' (categorical), and 'int64' (numeric).
3. Of the 15 variable columns, there are 9 categorical columns and 6 numeric columns.
4. The division is 14 feature columns and 1 target column.
5. In this dataset, missing values, marked with a '?', constitute approximately 7% of the entire dataset.

Description of the data distribution can be seen in the Univariate Analysis and Multivariate Analysis sections below.

Univariate Analysis

Figure 1.0: Univariate analysis of work class vs income

Description of the workclass distribution can be seen in the following table.

Table 1.0: Workclass Distribution

workclass	sample count	percentage
Private	33906	69.4
Self-emp-not-inc	3862	7.9
Local-gov	3136	6.4
?	2799	5.7
State-gov	1981	4.1
Self-emp-inc	1695	3.5
Federal-gov	1432	2.9
Without-pay	21	0.0
Never-worked	10	0.0

Looking at table 1.0, it can be seen that the mode category of the workclass is Private, with a difference of more than 60%.

Figure 1.1: Univariate analysis educationClass vs income

The description of the educationClass distribution can be seen in the following table.

Table 1.1: Distribution of educationClass

educationClass	sample count	percentage

educationClass	sample count	percentage
HS-grad	15784	32.3
Some-college	10878	22.3
Bachelors	8025	16.4
Masters	2657	5.4
Assoc-voc	2061	4.2
11th	1812	3.7
Assoc-acdm	1601	3.3
10th	1389	2.8
7th-8th	955	2.0
Prof-school	834	1.7
9th	756	1.5
12th	657	1.3
Doctorate	594	1.2
5th-6th	509	1.0
1st-4th	247	0.5
Preschool	83	0.2

In table 1.1, it can be seen that the mode category falls on HS-grad (high school graduate), but the distribution looks more balanced than the Workclass category, with a Right-skewed (positive) distribution trend .

Figure 1.2: Univariate analysis of status vs income

Description of the status distribution can be seen in the following table.

Table 1.2: Status distribution

status	sample count	percentage
Married-civ-spouse	22379	45.8
Never-married	16117	33.0
Divorced	6633	13.6
Separated	1530	3.1
Widowed	1518	3.1

status	sample count	percentage
Married-spouse-absent	628	1.3
Married-AF-spouse	37	0.1

Based on table 1.2, similar to educationClass, the distribution looks more balanced and not high in one category, with a Right-skewed distribution trend.

Figure 1.3: Univariate analysis of occupation vs income

Information on occupation distribution can be seen in the following table.

Table 1.3: Occupation distribution

occupation	sample count	percentage
Prof-specialty	6172	12.6
Craft-repair	6112	12.5
Exec-managerial	6086	12.5
Adm-clerical	5611	11.5
Sales	5504	11.3
Other-service	4923	10.1
Machine-op-inspct	3022	6.2
?	2809	5.8
Transport-moving	2355	4.8
Handlers-cleaners	2072	4.2
Farming-fishing	1490	3.1
Tech-support	1446	3.0
Protective-serv	983	2.0
Priv-house-serv	242	0.5
Armed-Forces	15	0.0

In table 1.3, the distribution is increasingly normal, with a mild Right-skewed distribution trend .

Figure 1.4: Univariate analysis relationship vs income

Description of the relationship distribution can be seen in the following table.

Table 1.4: Distribution of relationships

relationship	sample count	percentage
Husband	19716	40.4
Not-in-family	12583	25.8
Own-child	7581	15.5
Unmarried	5125	10.5
Wife	2331	4.8
Other-relative	1506	3.1

Then in table 1.4, it can be seen that the dataset trend is quite balanced, with the highest value of Husband. This result can imply that the distribution of Gender will be higher in men than women. This will be confirmed in the Univariate Analysis on the Gender feature.

Figure 1.5: Univariate analysis of race vs income

Description of the race distribution can be seen in the following table.

Table 1.5: Race distribution

race	sample count	percentage
White	41762	85.5
Black	4685	9.6
Asian-Pac-Islander	1519	3.1
Amer-Indian-Eskimo	470	1.0
Other	406	0.8

From table 1.5, it can be seen that the mode clearly falls in the White category, with a difference of up to 75.9%.

Figure 1.6: Univariate analysis of gender vs income

Information on gender distribution can be seen in the following table.

Table 1.6: Gender distribution

gender	sample count	percentage
Male	32650	66.8

gender	sample count	percentage
Female	16192	33.2

The results of table 1.6 confirm the theory of table 1.4, where the distribution of Male is greater than Female.

Figure 1.7: Univariate analysis native vs income

Description of the native distribution can be seen in the following table.

*only the top 15 values are displayed, due to too many unique values.

Table 1.7: Native distribution

native	sample count	percentage
USA	43832	89.7
Mexico	951	1.9
?	857	1.8
Philippines	295	0.6
Germany	206	0.4
PuertoRic	184	0.4
Canada	182	0.4
El-Salvador	155	0.3
India	151	0.3
Cuba	138	0.3
England	127	0.3
China	122	0.2
South	115	0.2
Jamaica	106	0.2
Italy	105	0.2

In table 1.7 it can be seen that the mode falls in the USA category, with a percentage difference of up to 87.8% with the next category.

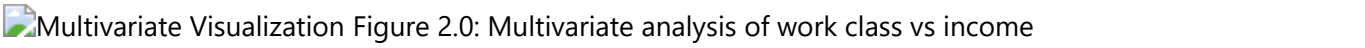
Figure 1.8: Univariate analysis of numerical features distribution

In Figure 1.8, it can be seen that the majority of numerical distributions have mode values with very high differences compared to the other values.

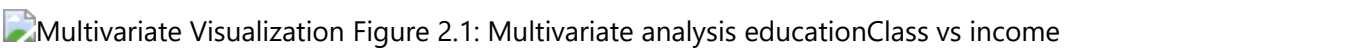
Analysis and interpretation of Univariate Analysis results:

1. Missing values are marked with '?'. The majority of the variable distributions are heavily biased towards the mode, with very large percentage differences. This implies that this dataset can be addressed for missing values using statistical methods, such as Mode Imputation .

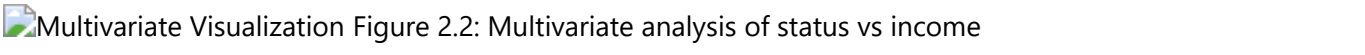
Multivariate Analysis


Multivariate Visualization Figure 2.0: Multivariate analysis of work class vs income

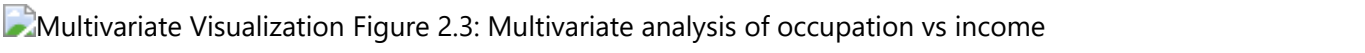
In workclass, it can be seen that all workclasses have income, except of course never-worked. On average, those who work for the government (-gov) have a higher income compared to other workclasses, but self-emp-inc has the highest income.


Multivariate Visualization Figure 2.1: Multivariate analysis educationClass vs income

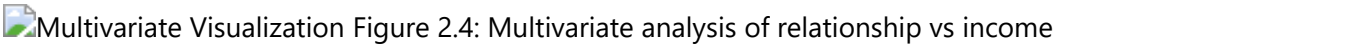
On educationClass, it can be clearly seen that there is a very large gap starting from the undergraduate level (bachelors), and even higher the higher the degree held (masters, doctorate, prof-school).


Multivariate Visualization Figure 2.2: Multivariate analysis of status vs income

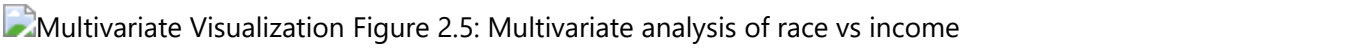
In terms of marital status, a trend can also be seen that data with stable marital status have higher incomes compared to those who are unmarried, divorced, separated, or divorced by death.


Multivariate Visualization Figure 2.3: Multivariate analysis of occupation vs income

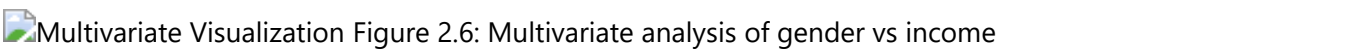
In occupation, the three highest-paying sectors are Specialty, Managerial, and Protective Service. In addition, there is no clear trend that can be observed.


Multivariate Visualization Figure 2.4: Multivariate analysis of relationship vs income

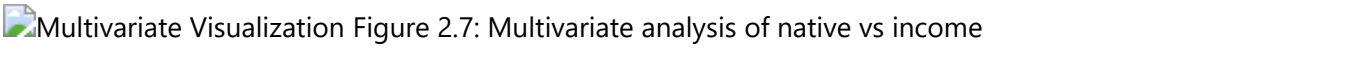
In relationships, it can be seen that it strengthens the results of observations of marital status, those with the status of husband and wife have the highest income from other data.


Multivariate Visualization Figure 2.5: Multivariate analysis of race vs income

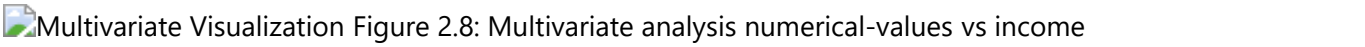
In racing, the two races with the highest income are White (white) and Asian-Pac (Asia-Pacific)


Multivariate Visualization Figure 2.6: Multivariate analysis of gender vs income

In terms of gender, it can be seen that men's income is higher than women's income.


Multivariate Visualization Figure 2.7: Multivariate analysis of native vs income

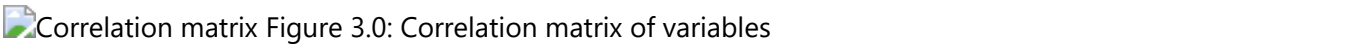
In native (country of origin/grandfather), it can be seen that the income trend is less observable, meaning that the country of origin/grandfather is not a strong factor that can influence income.


Multivariate Visualization Figure 2.8: Multivariate analysis numerical-values vs income

In the visualization of numeric features against income, it can be seen that there is less visible pattern in the influence of numeric features. This implies that the correlation value between numeric features and income is not very significant.

Analysis and interpretation of Multivariate Analysis results:

1. The results of the Multivariate Analysis show that the clearest trends are in educationClass, status, and relationship.
2. In Figure 2.1, which visualizes the educationClass analysis of income, it can be seen that there is a high jump starting from undergraduate education, and income increases as the level of education increases.
3. In Figures 2.2 and 2.4, status and relationship, it can be seen that samples in healthy marital relationships will have higher incomes compared to other statuses, and this is confirmed in the analysis of relationships.


Correlation matrix Figure 3.0: Correlation matrix of variables

The results of the correlation matrix above show that strong features such as educationClass are not too far from the correlation value with low values, such as capital-gain. Therefore, no more column drops will be performed.

Data Preparation

Sequentially, the data preparation processes that will be carried out are as follows:

1. Features Encoding
    - Features Encoding is the conversion of categorical features into numeric representation. The reasons for Features Encoding are as follows:
        - Although there are models that can process categorical data, the majority of existing models still require input in numeric form.
        - Continuing from the first point, sometimes there are ordinal relationships in the dataset. Representing categorical data as numeric can help preserve these relationships, so that they are implicitly understood by the model.
        - Machine learning models are models based on mathematics and statistics, so processing them in numerical form will speed up the training and testing process.
    - The Features Encoding techniques that will be used in this project are:
        1. Ordinal Encoding
        2. One Hot Encoding
2. Train Test Split
    - Train Test Split is a standard procedure in Machine Learning, where the dataset will be divided into training data (train) and test. This is done to ensure that the trained model can solve problems on real data that it has never encountered.
3. KNN Imputation

    - KNN Imputation is one of the processes that will be tested in this project, where missing values will be filled by looking at the similarity of other features in the same row, and inputting new values. KNN Imputation can cause Data Leakage, so it needs to be considered as a transformation process, rather than pre-processing. That is why KNN Imputation is done after the Train Test Split process.

4. Standardization

    - Standardization is also a standard procedure in performing Machine Learning. According to Google Machine Learning Developers[5], standardization is a method for changing the numerical values in data to a very similar scale, to improve the performance and stability of the model during train and test.

After the explanation regarding the project above, here is the flow that of the Data Preparation stage.

1. Features Encoding.

- The input at this stage is three DataFrames, with the following descriptions:
    - One original DataFrame, there are still missing values, and
    - Two identical DataFrames where Dropping has been performed. On the original DataFrame and one Dropped DataFrame, Ordinal Encoding is performed, and One Hot Encoding on the remaining DataFrame. Thus, the output of this stage is three DataFrames,
        - Dropped One Hot Encoded,
        - Dropped Ordinal Encoded, dan
        - Data Asli Ordinal Encoded.

2. Train Test Split

- Train Test Split is performed on each DataFrame, separating them into x and y.

3. KNN Imputation

- KNN Imputation is only performed on the Original Ordinal Encoded DataFrame. KNN Imputation is performed after Train Test Split to prevent Data Leakage.

4. Standarisasi

- At this stage standardization is performed on all x_train and x_test on all DataFrames, preparing the data for train and test.

Model Development

The Model Development process that the Author will carry out can be divided into several stages:

1. DataFrame initialization: prepare dataframes to store the model training results.

2. Pipeline Prep: prepare pipelines to simplify model training.

3. Model Training: train the model.

4. Visualization

Model building and training will be done simultaneously using pipelines.

Model Explanation, pros and cons

In this project, the author uses four algorithms, including Random Forest, K-Nearest Neighbor, XGBoost, and AdaBoostt.

1. **Random Forest**

- Random Forest is a Supervised Learning Based Algorithm, and is an implementation of ensemble learning. The Ensemble model itself is a group of models that work together to improve prediction performance. Random Forest is an ensemble (collection) of many decision tree models , combined with the Bagging technique , where each decision tree will produce a prediction, Bagging in the case of classification will take the most predictions on the entire tree as the final prediction. This is very suitable for this project because this property makes Random Forest not susceptible to bias.

- In this project, the parameters for the Random Forest algorithm used are:

    - n_estimator = 50
        - n_estimator is a parameter that determines the number of trees (decision trees) to be used in the Random Forest algorithm. The author used 50 for the experiment and the accuracy obtained was quite good. Since the accuracy of the Random Forest algorithm is not a priority goal of the project, 50 trees are considered sufficient
    - max_depth = 16
        - max_depth is a parameter that determines the depth of the tree, more precisely how much the tree can be divided to perform computation/observation. Similar to the first parameter, the accuracy is already good, so max_depth = 16 is considered sufficient.
    - random_state = 55
        - random_state is a parameter to set the random generator to ensure that every training/testing process is consistent.
    - n_jobs = -1
        - n_jobs is a parameter used to set the number of jobs running in parallel. n_jobs = -1 means all processes run in parallel.

- From the brief explanation regarding the Random Forest algorithm above, some of its advantages and disadvantages are as follows:

    - Pros:

    1. Can be used for both classification and regression.
    2. Can be used on categorical and numeric data.
    3. Can work without scaling and transformation.
    4. Can perform feature selection implicitly, and is robust to outliers
    5. Can work on both linear and non-linear problems, is robust to bias, and is accurate.

    - Cons:

    1. Computationally expensive, especially on large datasets.
    2. Not flexible, not much to be adjusted (less effective for tuning)

- **K-Nearest Neighbor**

    - KNN works by using feature similarities to predict the value of each new data. Mathematically, the KNN algorithm calculates the (Euclidean) distance between each data point and chooses a

classification based on the majority of nearest neighbors. When viewed from the complexity of the algorithm, KNN can be categorized as a simpler algorithm. KNN is a frequently used algorithm, including in this project, because of its simplicity and no assumptions. However, it should be noted that the curse of Dimensionality is not very effective on datasets with very many features.

- In this project, the parameters used by the KNN model are:
    - n_neighbors = 10
        - n_neighbors is a parameter to set how many neighbors will be considered for the Euclidean distance calculation process. The author chose 10 neighbors, because there are several features that have very weak correlations, so it is expected that these weak features are not prioritized in the training process.
- From the brief explanation regarding the KNN algorithm above, some of its advantages and disadvantages are as follows:
    - Pros:
    1. Simple, intuitive and easy to use.
    2. Very effective on multi-class problems.
    3. Easy to tune because the distance can be adjusted between Euclidean, Hamming, Manhattan, Minkowski, etc
    4. No assumptions.
    - Lack:
    1. Curse of Dimensionality, weak against datasets with large dimensions.
    2. KNN is not the fastest algorithm.
    3. Scaling and Transformation are mandatory.
    4. Weak against outliers, missing values, and imbalanced datasets.
- **XGBoost**
    - eXtreme Gradient Boosting (XGBoost) is an ensemble learning and boosting-based model, and is derived from the Gradient Boosting Decision Tree framework. Boosting itself is a process that creates and combines weak learner models iteratively to produce a strong learner model. The way boosting works is by building models sequentially with a focus on previously incorrect data. In the case of XGBoost, or Gradient Boosting in general, the iteration of model creation and combination is done based on the gradient of error or what is often called the Gradient Loss Function. Fundamentally, the difference between XGBoost and AdaBoost is the loss function used.
- From the brief explanation regarding the XGBoost algorithm above, some of its advantages and disadvantages are as follows:
    - Pros:
    1. Fast.
    2. Very resistant to outliers
    3. Flexible, able to adapt to high data variations.
    4. Built-in regularization.
    5. High accuracy.

    - Cons:
    1. Computationally expensive.
    2. Complex.
    3. Similar to Random Forest, XGBoost is less amenable to hyperparameter tuning.

- **AdaBoost**

    - Adaptive Boosting (AdaBoost) is a Supervised Learning Based Algorithm that implements ensemble learning and boosting. Similar to XGBoost, AdaBoost works by creating a number of weak learners based on decision trees, then creating subsequent models with the wrong answers from the previous model. Unlike XGBoost, AdaBoost uses Exponential Loss Function.

- In this project, the parameters for the AdaBoost algorithm used are:

    - n_estimators = 50
        - Similar to the Random Forest algorithm, AdaBoost accepts a n_estimators parameter to set the number of decision trees to be generated. The author used 50 as an experiment, and the results obtained were sufficient.
    - random_state = 123
        - random_state is a parameter to set the random generator to ensure that every training/testing process is consistent.

- From the brief explanation regarding the AdaBoost algorithm above, some of its advantages and disadvantages are as follows:

    - Pros:
    1. High potential, significantly improving model performance.
    2. Computationally inexpensive.
    3. Flexible, can be used in various cases.
    4. Can be used with other models.
    - Cons:
    1. High variance.
    2. Less effective in linear problems.
    3. More susceptible to outliers.

Evaluation

In this project, some evaluation metrics used are as follows. Before going into further explanation of the metrics, it is necessary to understand that::

- TN = True Negative, negative data that is predicted to be negative (true)
- TP = True Positive, positive data that is predicted to be negative (true)
- FN = False Negative, negative data that is predicted positive (wrong)
- FP = False positive, positive data that is predicted to be negative (wrong)

First, Accuracy, which can be calculated using the formula:

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

Accuracy represents the number of correctly predicted data divided by the total number of data. Ideally, accuracy gives an idea of how well a model can predict data, but the downside of this metric is that it is less fair if the dataset used is unbalanced .

Next is precision, which can be calculated using the formula:

$$Precision = \frac{TP}{FP + TP}$$

Precision is the ratio of true positive predictions (TP) to total positive predictions. The higher the precision, the fewer the number of false positive predictions (FP).

Then there is the Recall metric, as follows.

$$Recall = \frac{TP}{TP + FN}$$

Recall measures the value of the correct positive predictions from the actual number of positives. The higher the recall value, the fewer the False Negatives (FN).

After calculating the Accuracy, Precision, and Recall metrics, we can find the F1 Score value using the following formula.

$$F1 Score = 2* \frac{Precision * Recall}{Precision + Recall}$$

F1 Score is a harmonic value that uses precision and Recall, meaning that a high F1 Score value has high Precision and Recall. **Model Training Results**

- Visualization  Train Acc Figure 4.0: Model training results

Table 2.0 Evaluation of training Accuracy, Precision, Recall, and F1 Score data Dropped One Hot Encoded

Model	Accuracy	Precision	Recall	F1-Score
KNN	0.864862	0.794974	0.612125	0.691669
RandomForest	0.877515	0.840669	0.623536	0.716003
XGBoost	0.88582	0.821095	0.689026	0.749285
AdaBoost	0.858154	0.770313	0.608653	0.680007

Table 2.1 Evaluation of training Accuracy, Precision, Recall, and F1 Score Dropped Ordinal Encoded data

Model	Accuracy	Precision	Recall	F1-Score
KNN	0.860758	0.787511	0.599424	0.680714
RandomForest	0.89845	0.868751	0.69488	0.772148
XGBoost	0.887442	0.8215	0.696864	0.754067
AdaBoost	0.855402	0.767718	0.596547	0.671394

Table 2.2 Evaluation of training Accuracy, Precision, Recall, and F1 Score of KNN data Imputed Ordinal Encoded

Model	Accuracy	Precision	Recall	F1-Score
KNN	0.86414	0.787736	0.589262	0.674195
RandomForest	0.900721	0.859779	0.697597	0.770243
XGBoost	0.891053	0.82002	0.696071	0.752979
AdaBoost	0.859909	0.775178	0.581347	0.664414

1. From the training results, the two best models are Random Forest and XGBoost. This is proven not only by the very high training accuracy, but also by the very high F1 Score. A high F1 Score is an indication that the model rarely makes mistakes in predicting true or false values. The explanation of the F1 Score will be explained in more depth at the Evaluation stage.
2. It can also be observed that from the One Hot Encoded and Ordinal Encoded drop datasets, there is a significant change (up to two percent increase) in training accuracy. This shows the importance of preserving ordinality in the dataset (if any), which in this case is mainly in the educationClass feature.
3. The final prediction test will use test data from test_ord, because this dataset implements Ordinal Encoding and has a higher level of integrity than the imputation dataset. Given that the number of this dataset is abundant from the beginning, the Drop method remains the best Data Handling method because of its integrity preservation.

Testing and Final Predictions

Looking at the results of the training results above, especially from test_dum and test_ord, it can be seen that there is an increase in accuracy from the two models with the highest accuracy, namely Random Forest and XGBoost. This confirms that it is important to store ordinality features (if any) which in this case are in the dataset, especially educationClass.

Consistently, the Random Forest and XGBoost models produced high accuracy and F1 Score, an indication of a superior model.

The final prediction test will use test data from test_ord because this dataset implements the Drop and Ordinal Encoding methods so that this dataset maintains its ordinality properties and has a higher level of integrity than the imputation dataset. Given that the number of this dataset is abundant from the beginning, the Drop method remains the best Data Handling method because of its integrity preservation.

The Testing method will be the same as the Training method, using Pipeline to increase Testing efficiency.

Testing Results

Table 3.0 Evaluation of test Accuracy, Precision, Recall, and F1 Score data Dropped One Hot Encoded

Model	Accuracy	Precision	Recall	F1-Score
KNN	0.858943	0.770925	0.619469	0.686948
RandomForest	0.928366	0.930556	0.770796	0.843175

Model	Accuracy	Precision	Recall	F1-Score
XGBoost	0.940084	0.909438	0.844248	0.875631
AdaBoost	0.86845	0.772126	0.671681	0.71841

Table 3.1 Evaluation of test Accuracy, Precision, Recall, and F1 Score data Dropped Ordinal Encoded

Model	Accuracy	Precision	Recall	F1-Score
KNN	0.853858	0.77686	0.582301	0.665655
RandomForest	0.954455	0.93097	0.883186	0.906449
XGBoost	0.941853	0.897342	0.866372	0.881585
AdaBoost	0.860491	0.759086	0.646903	0.698519

Table 3.2 Evaluation of training Accuracy, Precision, Recall, and F1 Score of KNN data Imputed Ordinal Encoded

Model	Accuracy	Precision	Recall	F1-Score
KNN	0.845855	0.762911	0.541216	0.63322
RandomForest	0.951894	0.917098	0.884263	0.900382
XGBoost	0.938588	0.895522	0.849292	0.871795
AdaBoost	0.854452	0.751025	0.610325	0.673404

A summary and visualization of the data handling methods, and their accuracy can be seen as follows.

 Test Acc Figure 5.0: Testing results

The test results that will be reviewed, and used for the final prediction, are from the test data with Dropped Encoded. This is because the Drop method is the best method in maintaining data integrity, where the results can be seen from the following Confusion Matrix visualization.

 Confusion Matrix Figure 5.1: Confusion matrix testing

Project Results and Conclusions

Based on the results of Data Understanding, Data Preparation, Model Development, and Evaluation, we can conclude the following things.

1. Answering Problem Statement 1: The features that have the most influence on income in this problem are educationClass, status, and relationship.
2. Answering Problem Statement 2: Yes, income can be predicted, and based on the Accuracy, Recall, Precision, and F1 Score values , the Random Forest model is the model with the best performance, achieving an accuracy of up to 95.4%, a precision of up to 93%, a recall of 88.3%, and an F1Score of 90.6%. This is one of the advantages of Random Forest, namely its nature of performing feature selection implicitly. Plus the properties inherited from ensemble learning , namely resistance to bias and overfitting.

3. There is an observable effect in the form of increased model performance if ordinality preservation (if present in the dataset) is performed.
4. KNN Imputation has a less significant impact compared to point number 3, but performance still increases.

Based on the findings above, it can be concluded that this project was successful and ran according to the Author's wishes, namely answering the Problem Statements and achieving the formulated Predictive Modeling Goals, and successfully observing the differences in ordinality and the influence of KNN Imputation on the dataset.

Referensi

[Dataset] Becker, Barry and Kohavi ,Ronny. (1996). Adult. UCI Machine Learning Repository. https://doi.org/10.24432/C5XW20.

[1] Badan Pusat Statistika Indonesia. (2023). Diakses dari https://www.bps.go.id/id/pressrelease/2023/11/06/2002/tingkat-pengangguran-terbuka--tpt--sebesar-5-32-persen-dan-rata-rata-upah-buruh-sebesar-3-18-juta-rupiah-per-bulan.html .

[2] R. Pramudjasi, Juliansyah, D. Lestari. (2019). Effect of population and education and wages on unemployment in paser regency. Journal of Faculty of Economics and Business Universitas Mulawarman. Diakses dari https://journal.feb.unmul.ac.id/index.php/KINERJA/article/download/5284/472.

[3] S.A.E. Sari, F.W.Pangestuty. (2020). Analisis Pengaruh jumlah Penduduk, Tingkat Pendidikan, dan Produk Domestik Regional Bruto Terhadap Tingkat Pengangguran Terbuka di Provinsi Jawa Timur Tahun 2017-2020. Journal of Developement Econoic and Social Studies. Diakses dari https://jdess.ub.ac.id/index.php/jdess/article/download/78/57/373.

[4] B. Jepchumba. (2020). Getting started with using Visual Machine Learning Tools for building your Machine Learning Models. Microsoft Community Hub. Diakses dari https://techcommunity.microsoft.com/t5/educator-developer-blog/getting-started-with-using-visual-machine-learning-tools-for/ba-p/3578397.

[5] Google Machine Learning Developers. Normalization. Diakses dari https://developers.google.com/machine-learning/data-prep/transform/normalization#:~:text=The%20goal%20of%20normalization%20is,training%20stability%20of%20the%20model.

Daftar Pustaka

[1] P. Schmitt, J. Mandel, M. Guedj. (2015). A Comparison of Six Methods for Missing Data Imputation. Journal of Biometrics and Biostatics. DOI: 10.472/2155-6180.1000224. Diakses dari https://www.hilarispublisher.com/open-access/a-comparison-of-six-methods-for-missing-data-imputation-2155-6180-1000224.pdf.

[2] Kleindessner, M., Awasthi, P., & Morgenstern, J. (2019). Fair k-Center Clustering for Data Summarization. International Conference on Machine Learning. Diakses dari https://www.semanticscholar.org/paper/Fair-k-Center-Clustering-for-Data-Summarization-Kleindessner-Awasthi/9c26bbf34bdab544a000038d628a8fb232d60cb6.

[3] Goutte, C., & Gaussier, E. (2005). A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. Advances in Information Retrieval, 345–359. Diakses dari doi:10.1007/978-3-540-31865-1_25.

[4] J. Brownlee. (2020). Machine Learing Mastery. Diakses dari https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/ .

[5] Dicoding Academy. Diakses dari https://www.dicoding.com/ .