Income Prediction Using CRISP-DM Methodology

1. Business Understanding

Objective:

The goal of this project is to predict whether an individual's income exceeds 50K (`>50K`) or is less than or equal to 50K (`<=50K`). This binary classification problem involves using demographic and employment-related data, such as age, education, occupation, and hours worked, to estimate an individual's income level. The outcome of this analysis can be used by companies, policymakers, and research institutions to better understand the socioeconomic factors affecting income distribution.

Justification for Dataset:

The **UCI Adult (Census Income)** dataset is used in this project, which includes demographic and employment data for individuals. This dataset is relevant for predicting income as it captures various factors, such as age, education, work class, and occupation. Understanding these relationships allows for better targeting of policies, social programs, and business strategies to address income disparities.

---

2. Data Understanding

Exploratory Data Analysis (EDA)

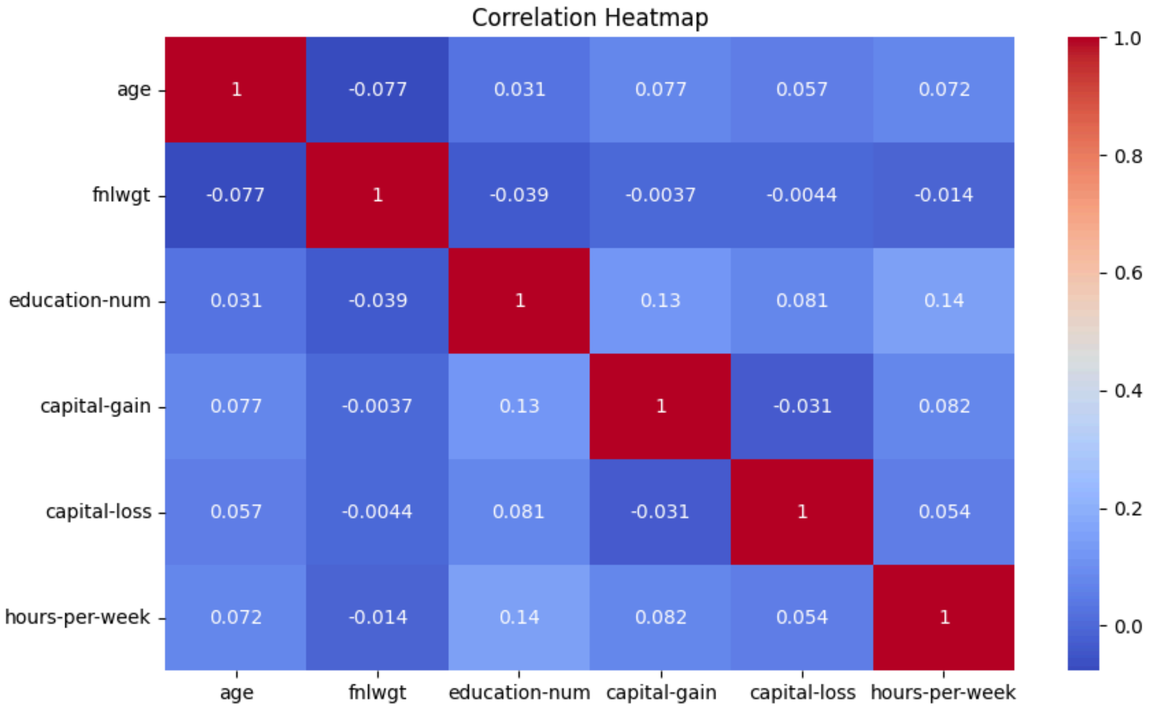
Dataset Overview:

The dataset consists of 14 columns, including both **numeric** and **categorical** features. Below is a summary of the columns and their types:

Column	Type	Description	Units	Missing Values
age	Integer	Age of the individual	N/A	No
workclass	Categorical	Type of employer	Private, etc.	Yes

fnlwgt	Integer	Census sampling weight	N/A	No
education	Categorical	Highest education level	Bachelors, etc.	No
education-num	Integer	Education level (numeric)	N/A	No
marital-status	Categorical	Marital status	Married, etc.	No
occupation	Categorical	Job type	Tech-support, etc.	Yes
relationship	Categorical	Relation to head of family	Wife, etc.	No
race	Categorical	Ethnicity	White, etc.	No
sex	Binary	Gender	Male, Female	No
capital-gain	Integer	Monetary gain	USD	No
capital-loss	Integer	Monetary loss	USD	No
hours-per-week	Integer	Work hours per week	Hours	No
native-country	Categorical	Country of origin	Various	Yes
income	Target (Binary)	Income class	>50K, <=50K	No

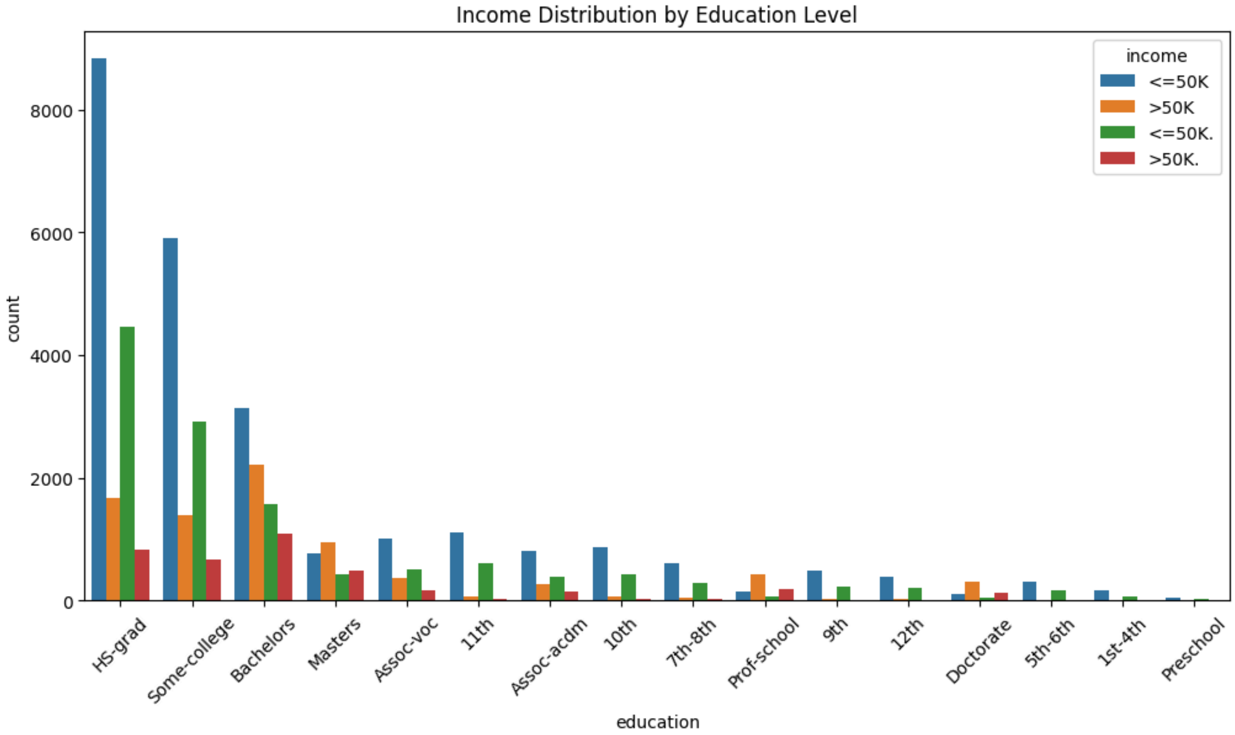
Correlation Analysis:



A **correlation matrix** was generated to assess relationships between numeric features. The following relationships were observed:

Feature Pair	Correlation	Interpretation
education-num ↔ hours-per-week	0.14	More educated individuals tend to work slightly more hours.
education-num ↔ capital-gain	0.13	Higher education often correlates with higher capital gains.
age ↔ capital-gain	0.077	Older individuals tend to have slightly higher capital gains.
capital-gain ↔ capital-loss	-0.031	Negligible inverse relationship between capital gain and loss.

Income Distribution by Education Level:



The **Income Distribution by Education Level** chart below presents the count of individuals earning `>50K` and `<=50K` for each education level.

(Chart Placeholder)

Key Findings:

- **Higher Education → Higher Income**: Individuals with higher education (Doctorate, Masters) have a higher proportion of earners making `>50K`.
- **Dominance of High School & Some College**: These are the most common education levels, but most individuals in these categories earn `<=50K`.
- **Steep Drop in Income with Less Education**: Individuals with education levels below 9th grade (e.g., 1st-4th, 5th-6th, Preschool) almost exclusively earn `<=50K`. These categories show nearly no high-income earners.
- **Bachelors Degree as a Turning Point**: The number of `>50K` earners significantly increases starting from the `Bachelors` level.

Actionable Notes for Modeling:

- Use `education-num` to preserve ordinal relationships.
- May group lower education levels into a single category like **Low-edu** to simplify modeling.
- Consider visualizing `education-num` vs. income using a **boxplot** or **violin plot** for additional insight.

---

3. Data Preparation

Data Cleaning and Transformation

Feature Removal:

- The **`fnlwgt`** feature was dropped as it does not offer predictive power for the classification problem.

Handling Missing Values:

- **Missing values** in the columns **workclass**, **occupation**, and **native-country** were handled by imputation, filling them with the label **"Unknown"** to retain all records.

Skewness Treatment:

- The **`capital-gain`** and **`capital-loss`** features were found to be heavily skewed. A **log transformation** using `log1p` was applied to these features to normalize their distributions, making them more suitable for modeling.

Categorical Encoding:

- **Label Encoding** was applied to all categorical features to convert them into numeric values, allowing them to be used in machine learning models.

Target Variable Encoding:

- The **`income`** column was transformed into a binary target variable:

  - `>50K` → 1
  - `<=50K` → 0

Redundant Feature Removal:

- The **`education`** column was dropped in favor of **`education-num`**, as the latter preserves the ordinal relationships and is numerically encoded.

Data Quality Issues Identified:

Feature	Issue	Action Taken
capital-gain	Highly skewed with extreme values.	Log-transformation applied.
capital-loss	Similar skew as capital-gain.	Log-transformation applied.
hours-per-week	Presence of extreme outliers (e.g., >80 hours).	Visual inspection and possible clipping.
fnlwgt	No predictive value for this classification.	Removed.
Missing values	Columns like `workclass`, `occupation`, `native-country` contain missing values.	Imputed with "Unknown".

4. Modeling (with SMOTE)

Objective:

Train multiple machine learning models to classify whether an individual's income exceeds $50K using census features. Evaluate each model using appropriate metrics and select the best-performing models for deployment.

Models Trained:

- Logistic Regression
- Decision Tree
- Random Forest
- K-Nearest Neighbors (KNN)
- Support Vector Machine (SVM)
- Gradient Boosting Classifier

Class Imbalance Handling:

To address class imbalance (~75% of individuals earn `<=50K`), **SMOTE** (Synthetic Minority Oversampling Technique) was applied to the training data. This helped create a more balanced representation of high-income individuals during training.

Evaluation Metrics Used:

- **Accuracy:** Correct predictions over total predictions
- **Precision:** Proportion of correctly predicted positives over all predicted positives
- **Recall:** Proportion of correctly predicted positives over actual positives
- **F1-Score:** Harmonic mean of precision and recall (balance between the two)

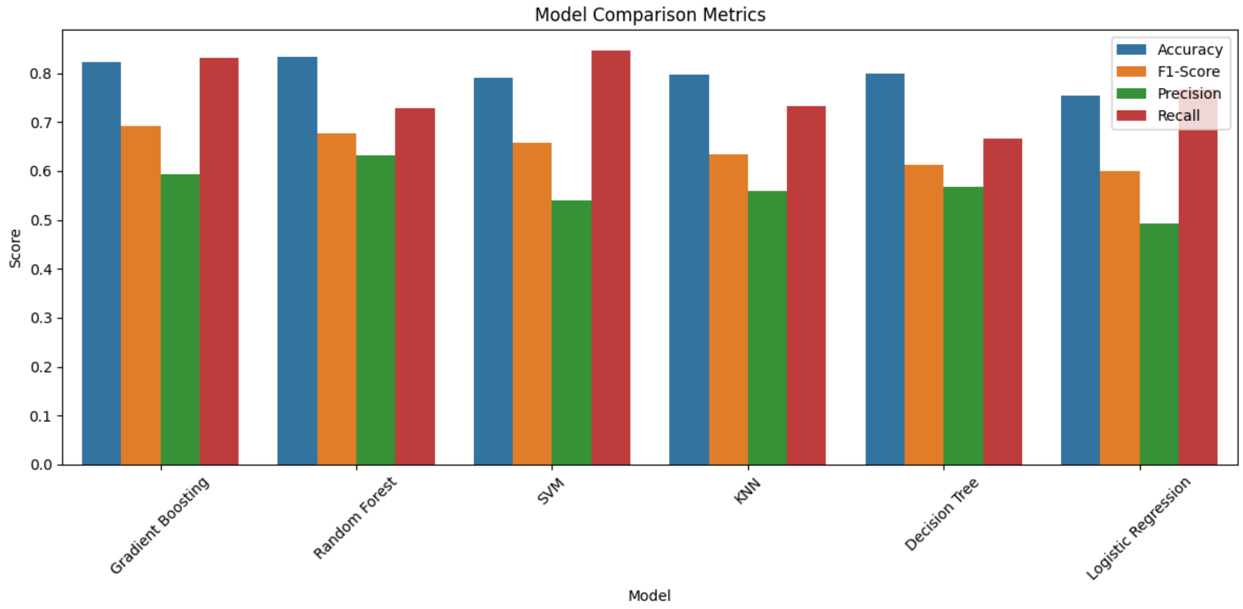
Evaluation Results After SMOTE:

Model	Accuracy	F1-Score	Precision	Recall
Gradient Boosting	0.823	0.692	0.593	0.832
Random Forest	0.834	0.677	0.632	0.729
SVM	0.790	0.659	0.539	0.846
KNN	0.798	0.634	0.559	0.732

Decision Tree	0.800	0.616	0.569	0.672
Logistic Regression	0.755	0.600	0.493	0.767

Model Comparison Metrics (SMOTE-based)

The **Model Comparison Metrics** chart below illustrates the performance of the various models across different evaluation metrics:



Observations:

- **Gradient Boosting** achieved the highest **F1-Score** (0.692) and strong recall (0.832), making it the best-performing model overall.
- **SVM** had the highest recall (0.846), which is useful when minimizing false negatives is critical.
- All models saw significant improvement in recall compared to results before SMOTE.
- **Precision** may slightly decrease when improving recall, a trade-off that needs consideration depending on the application.

Final Model Selection:

Model	Use Case
Gradient Boosting	✅ Final model for deployment (balanced, high F1)
Random Forest	🔄 Reliable backup / interpretable alternative
SVM	⚠️ High-recall use case (e.g., policy targeting)

6. Deployment

Objective:

The objective of the deployment phase is to propose how the model can be implemented in real-world applications.

Proposed Deployment Strategy:

1. **Model Application**: The **Gradient Boosting** model, selected for its high performance, can be deployed as an API in a production environment. The model will predict whether an individual's income exceeds 50K based on demographic features like education, age, and work experience.

2. **API Integration**:

   - An API endpoint could be created to receive input from a web application or other data sources. Users would input their demographic details (e.g., age, education level, hours worked), and the model would return a prediction of whether their income is likely to exceed 50K.

3. **Web Application**:

   - The model could be integrated into a web-based dashboard where companies or organizations can input employee or candidate data to evaluate income potential based on historical data.

4. **Monitoring and Maintenance**:

   - **Monitoring**: The deployed model should be regularly monitored for performance decay, especially if there are shifts in the data over time (e.g., changes in societal trends or economic conditions).

- **Model Retraining**: Periodic retraining of the model with fresh data should be scheduled to ensure that it adapts to any shifts in income patterns.

- **Data Drift Detection**: Implement data drift monitoring to ensure that the distribution of incoming data doesn't drastically differ from the training data.

Conclusion

This project successfully applied the **CRISP-DM methodology** to predict whether an individual's income exceeds 50K based on demographic and employment data. The entire process was executed step-by-step, with each phase contributing to the final goal of deploying a robust income prediction model.

Step 1: Business Understanding

The business objective was clearly defined as a binary classification problem—predicting whether an individual's income is greater than 50K. This problem holds substantial real-world relevance, particularly in economic research and policymaking, where understanding income distribution is crucial. The UCI Adult dataset, which provides demographic and employment-related features, was selected to build the model due to its comprehensive nature and the relevance of its features to the task at hand.

Step 2: Data Understanding

Through careful exploration of the dataset, key insights were drawn from exploratory data analysis (EDA), including the identification of crucial features such as education, age, and hours worked, which are closely tied to income. The distribution of these variables was understood, and correlations were examined to establish how these features interact with one another and the target variable. This step also highlighted the need for careful handling of missing values and skewed distributions in certain features.

Step 3: Data Preparation

In the data preparation phase, the dataset was cleaned, transformed, and encoded for machine learning. The preprocessing steps included handling missing values through imputation, addressing feature skewness with log transformations, and encoding categorical features to ensure compatibility with machine learning algorithms. Redundant and irrelevant features, such as the `fnlwgt` feature, were removed, ensuring that the final dataset was optimal for modeling. This careful data preparation set the foundation for effective model training and evaluation.

Step 4: Modeling

Multiple machine learning models were trained and evaluated using the prepared dataset. To address the class imbalance in the data, **SMOTE (Synthetic Minority Oversampling Technique)** was applied, which resulted in a more balanced training set. Several models, including **Logistic Regression**, **Random Forest**, **SVM**, **KNN**, **Decision Tree**, and **Gradient Boosting**, were trained, and their performance was evaluated using accuracy, precision, recall, and F1-score. This comprehensive approach allowed us to compare model performances and identify the best model for the task.

Step 5: Evaluation

After applying SMOTE, **Gradient Boosting** emerged as the best-performing model, achieving the highest F1-score and strong recall. **Random Forest** and **SVM** also performed well, with **SVM** showing the highest recall, which is particularly useful when minimizing false negatives is critical. The results showed significant improvements across all models compared to their pre-SMOTE performance, particularly in recall, which is essential in predicting income in scenarios where identifying high-income individuals is a priority.

Step 6: Deployment

The final step proposed the deployment strategy for the selected **Gradient Boosting** model. This model could be deployed as part of a web application or API, enabling real-time predictions of income based on user inputs. To ensure long-term performance, the system would incorporate monitoring and retraining processes to adapt to changes in the underlying data. This deployment would be valuable for applications in business strategy, policymaking, and social program targeting.

Final Thoughts

This project demonstrated the application of data mining techniques using the CRISP-DM methodology, leading to the successful development of a robust model for predicting income levels. By leveraging EDA, feature engineering, SMOTE for class balancing, and a thorough evaluation of multiple models, we were able to identify the **Gradient Boosting** model as the best candidate for real-world deployment. This model can be effectively used in various applications, from predicting income in recruitment processes to informing public policies aimed at reducing income disparity.