Project Report - Group 10

CSP571 Data Preparation and Analysis

Group Members

Sr. No.	Name	CWID
1.	Belthnagady Akash Vittaldas Narayana Pai	A20560317
2.	Merlin Santano Simoes	A20531255
3.	Harneet Kaur Dehiya	A20548613
4.	Sana Samad	A20543001
5.	Owaiz Majoriya	A20546104

Github URL: https://github.com/hdehiya/CSP571Project

Dataset

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

This dataset is a sample of the US population of adults. The demographic attributes in this dataset include **age**, **educational attainment**, and **occupation**, while financial variables pertain to **capital gains/losses** and **hours worked per week**.

This analysis will attempt to break down the adult population into meaningful segments or clusters based on the available features. By understanding the different features associated with various clusters, businesses can develop strategies and extend targeted offers to different customer groupings.

The dataset contains approximately **48,000 observations** and **14 features** for each individual. A few important variables in this dataset include:

- **age**: Age of the subject in years
- **education-num**: Number of years of education the person has completed
- **capital-gain**: Person's capital gains in US dollars
- **capital-loss**: Person's capital losses in US dollars
- **hours-per-week**: Number of hours the person works per week

Data Preprocessing

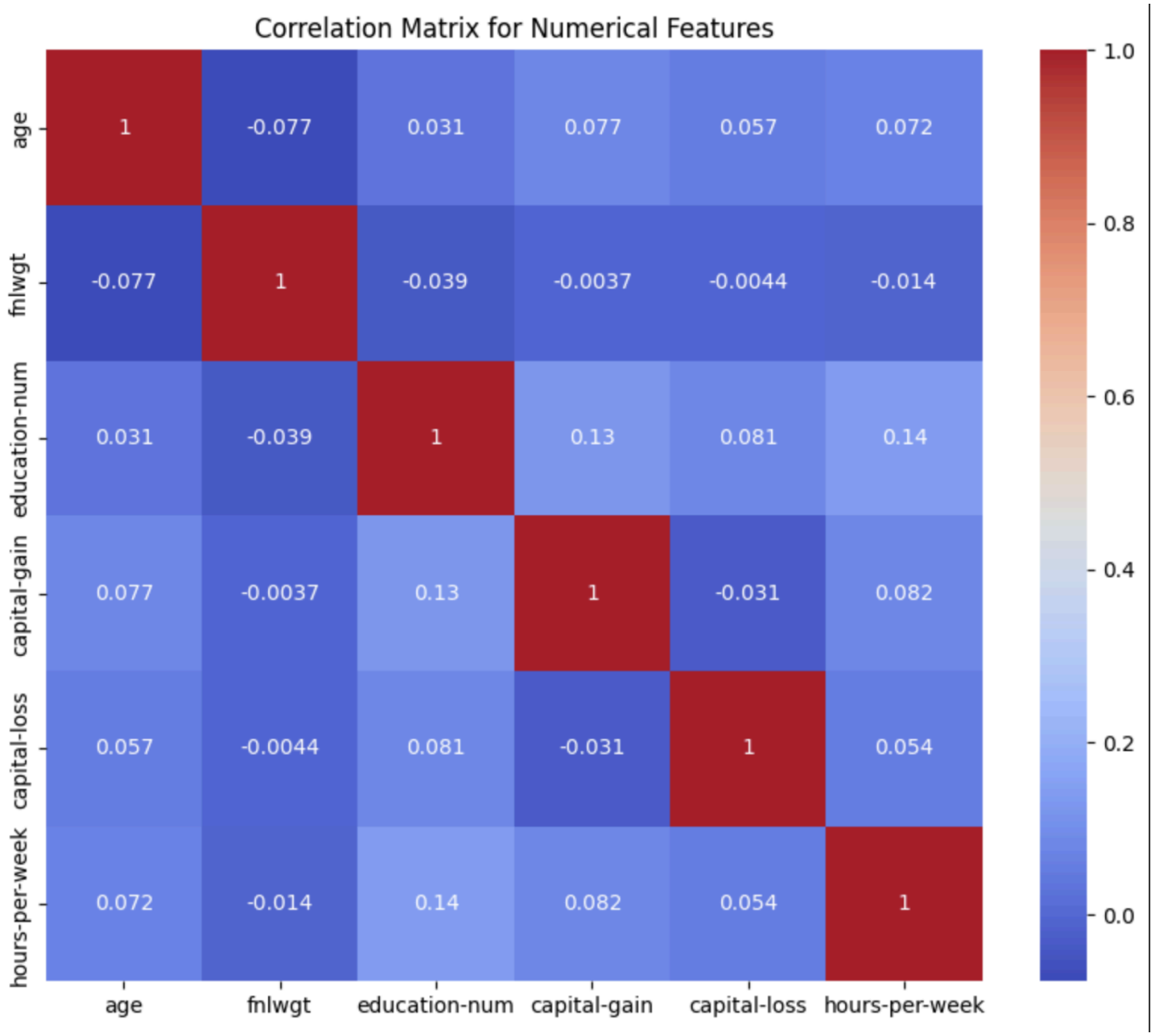
Preparing the dataset for K Means clustering involved the following steps:

- **Feature Identification**: Differentiated between categorical and numerical features.
- **Data Transformation**: Applied encoding for categorical variables and scaling for numerical features using preprocessing pipelines.
- **Integration**: Combined preprocessing steps into a **ColumnTransformer** to streamline the data transformation process.

Correlation Between Features

- Numerical Features: Correlation matrix with Pearson correlation for numeric features.

The heatmap below visualizes the Pearson correlation matrix for numerical features in the dataset, highlighting their pairwise relationships. Values closer to 1 indicate a strong positive correlation, while values closer to -1 reflect a strong negative correlation. Most numerical features in the dataset exhibit low correlations, suggesting minimal multicollinearity, with a few notable interactions like between education-num and hours-per-week. Such analyses help in understanding feature dependencies and guide model development.

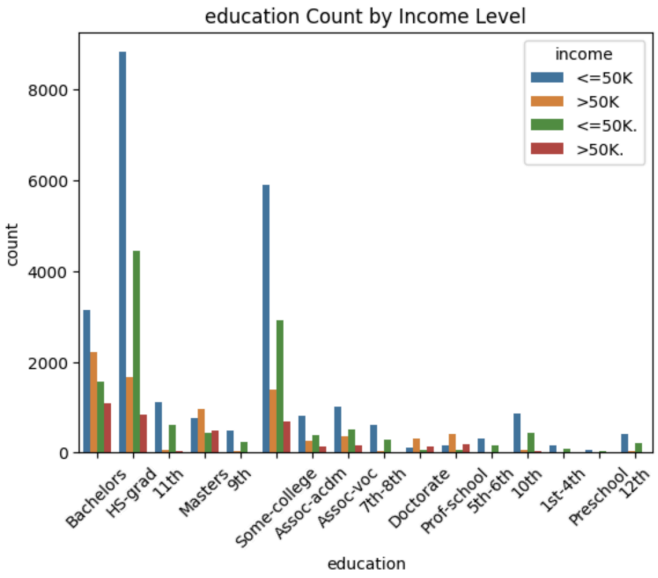


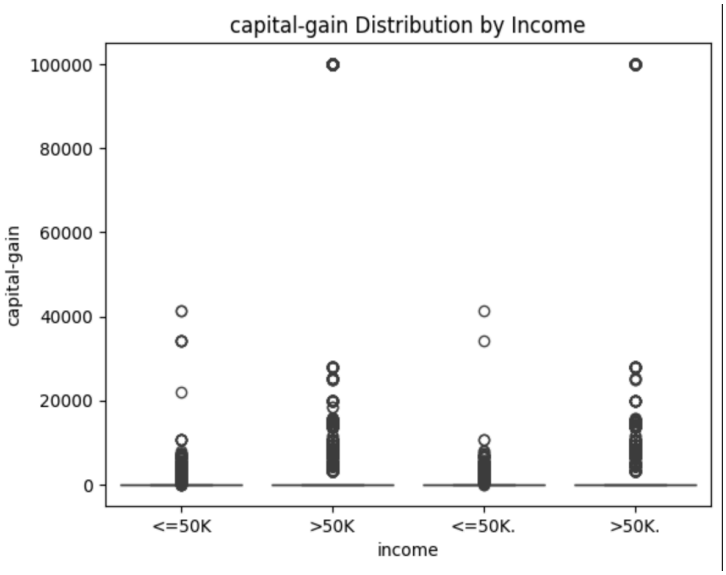
Feature Impact on Target

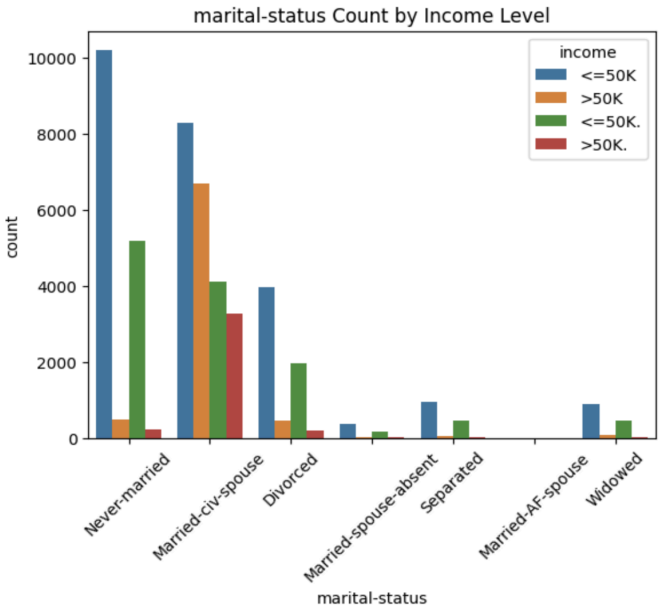
The visual analysis illustrates the impact of individual features on income classification.

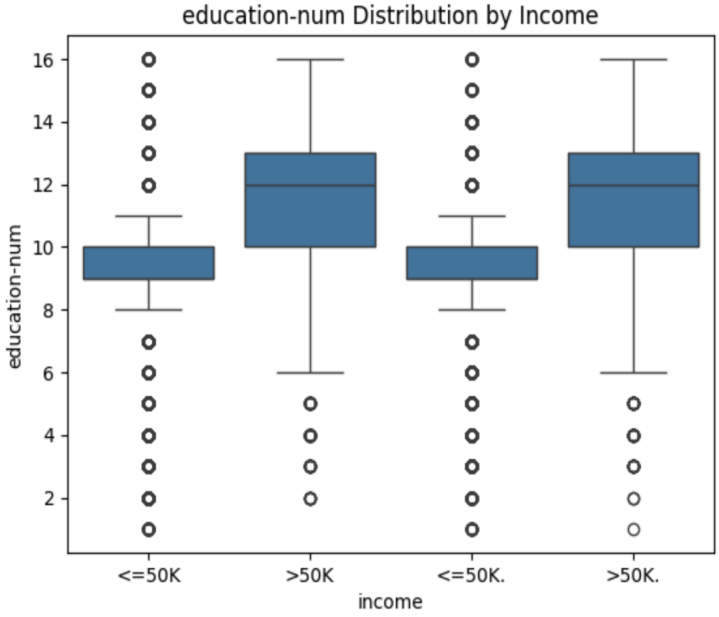
- Boxplots for numerical features, such as age, education-num, and hours-per-week, reveal clear differences in their distributions across income levels, highlighting their potential predictive power.
- Count plots for categorical features like workclass, education, and marital-status show significant variations in class proportions, providing valuable insights into the relationship between these features and income categories. These visualizations underscore the importance of both numerical and categorical variables in predicting income levels effectively.

- The visualizations highlight key insights into the relationship between features and income levels. Higher-income earners tend to be older, have higher education levels (education-num), work more hours, and show non-zero capital gains/losses. Categorical features like workclass, education, and marital-status also show clear patterns, with private-sector and self-employed roles, advanced degrees, and being married strongly associated with higher incomes. These distinctions underscore the predictive power of both numerical and categorical features, guiding effective feature selection and preprocessing for the income classification model.









Independence Assumptions

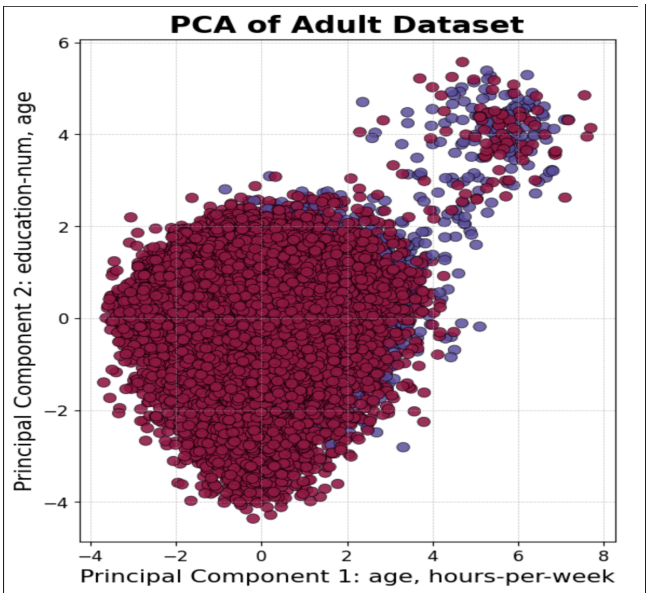
Statistical tests, such as the Chi-square test, indicate that categorical features like workclass, education, marital-status, and others are significantly dependent on the income target variable, with extremely low p-values providing strong evidence against the null hypothesis of independence. These findings emphasize the predictive relevance of these features, underscoring their necessity in the modeling process to ensure an accurate and effective analysis pipeline.

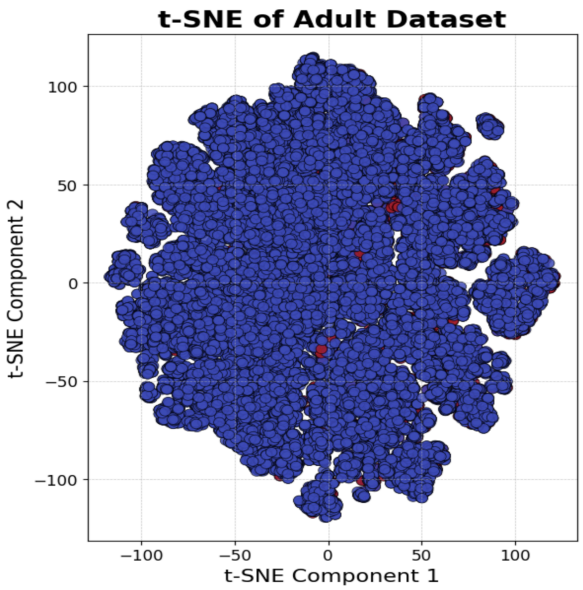
```
Feature: workclass, p-value: 0.0
workclass is dependent on the target.
Feature: education, p-value: 0.0
education is dependent on the target.
Feature: marital-status, p-value: 0.0
marital-status is dependent on the target.
Feature: occupation, p-value: 0.0
occupation is dependent on the target.
Feature: relationship, p-value: 0.0
relationship is dependent on the target.
Feature: race, p-value: 3.4347723295010175e-98
race is dependent on the target.
Feature: sex, p-value: 0.0
sex is dependent on the target.
Feature: native-country, p-value: 2.1525667073421703e-102
native-country is dependent on the target.
```

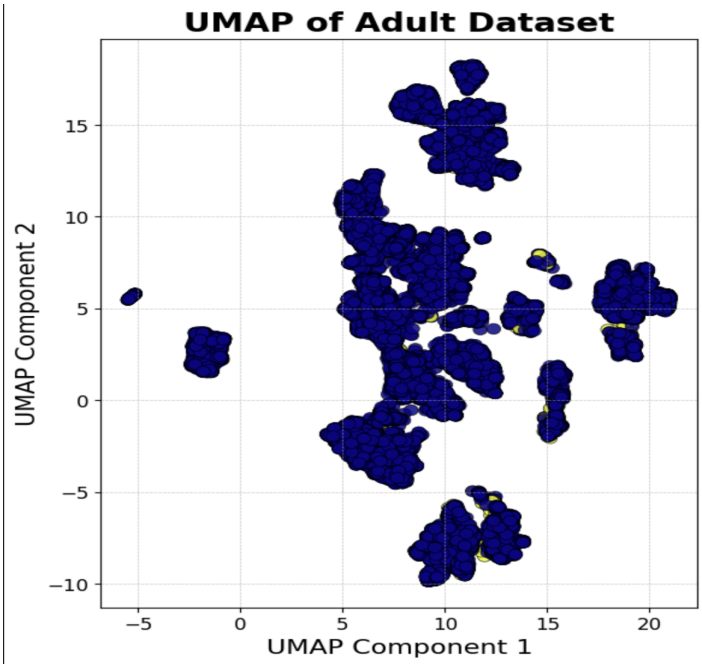
Dimensionality reduction techniques such as **PCA, t-SNE, and UMAP** were applied to the dataset to gain insights into feature relationships and class separability:

1. **PCA (Principal Component Analysis):** PCA reduces the dataset to two principal components, capturing the variance of features like age, hours-per-week, and education-num. The visualization shows a dense cluster of points with limited separation between income classes, indicating that the dataset's variance is not strongly aligned with linear relationships.
2. **t-SNE (t-Distributed Stochastic Neighbor Embedding):** t-SNE creates a non-linear embedding, preserving local relationships between data points. The resulting plot shows a scattered arrangement with subtle patterns, suggesting some separability between income groups but no distinct clusters.
3. **UMAP (Uniform Manifold Approximation and Projection):** UMAP provides a non-linear representation focusing on both global and local structures. The plot highlights more defined groupings, capturing fine-grained data relationships and potentially aiding in better class discrimination.

These techniques illustrate the complexity of the dataset, with PCA revealing limited linear separability, while t-SNE and UMAP highlight non-linear relationships critical for classification.







Random Forest

Random Forest was chosen for its robustness, ability to handle both numerical and categorical features, and effectiveness in managing class imbalances through techniques like class weighting. It is an ensemble learning method that constructs multiple decision trees during training and combines their outputs to improve accuracy and reduce overfitting, making it a reliable choice for classification tasks.

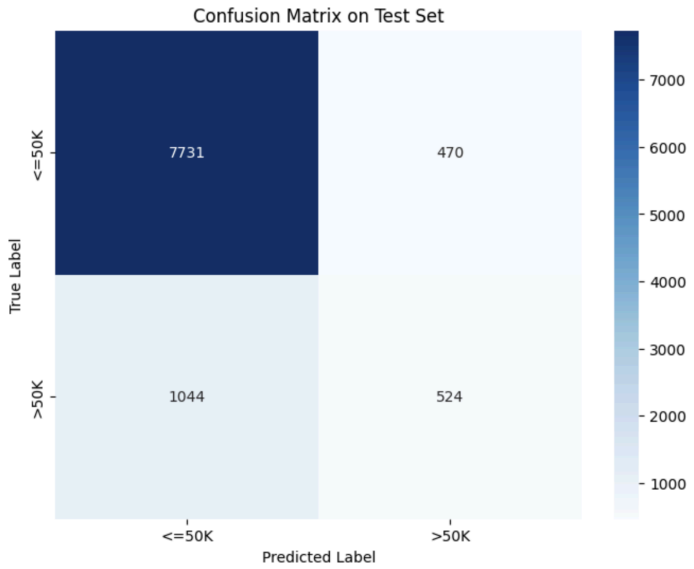
The Random Forest model was optimized using **hyperparameter tuning** with **RandomizedSearchCV** and evaluated using **10-fold cross-validation**.
The preprocessing pipeline included **scaling numerical features** and **one-hot encoding categorical features**.
The best hyperparameters, including **100 estimators** and **balanced class weight**, achieved a test accuracy of **84.5%**.
The **classification report** shows:

- Strong performance in predicting the majority class (<=50K) with **high precision, recall, and F1-score**.
- Struggles with the minority class (>50K), achieving **33% recall** and **41% F1-score**.

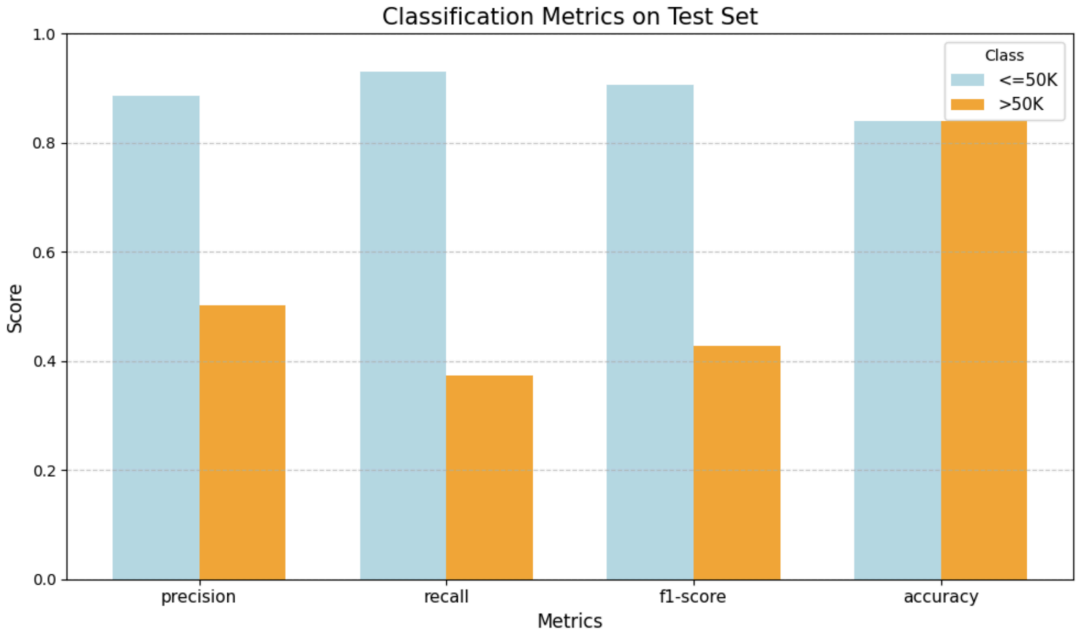
The **confusion matrix** confirms this imbalance, with a higher rate of **false negatives** for the minority class.

While the model demonstrates **robust overall accuracy**, further improvements could be achieved by applying:

- **Class balancing techniques** (e.g., oversampling, undersampling).
- **Feature engineering** to better capture patterns in the minority class.



The bar chart compares classification metrics (precision, recall, F1-score, and accuracy) for the two income classes (<=50K and >50K). While the model performs well for the majority class (<=50K), achieving high scores across all metrics, it struggles with the minority class (>50K), particularly in recall and F1-score, reflecting difficulty in capturing true positives for this group.



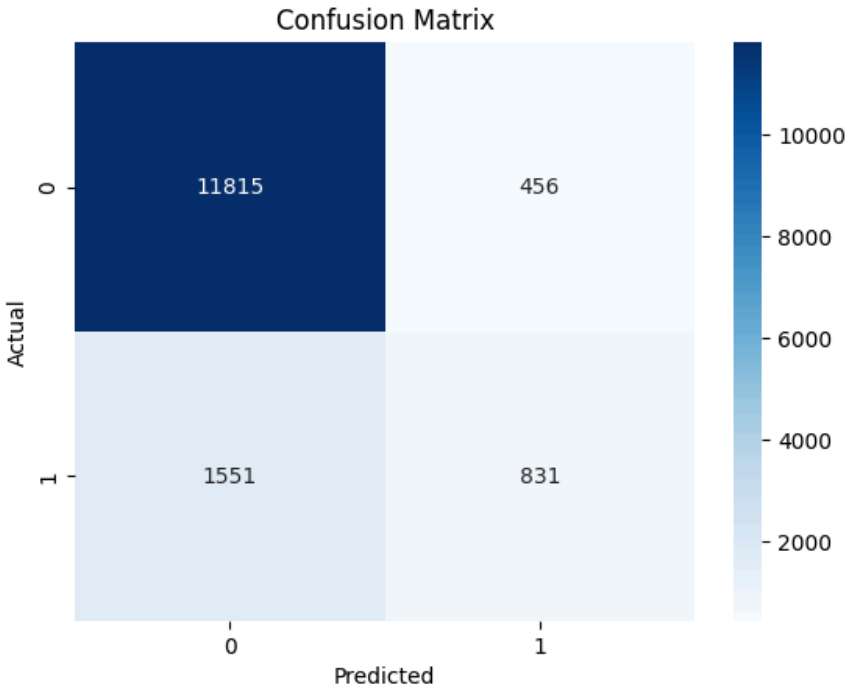
Gradient Boosting

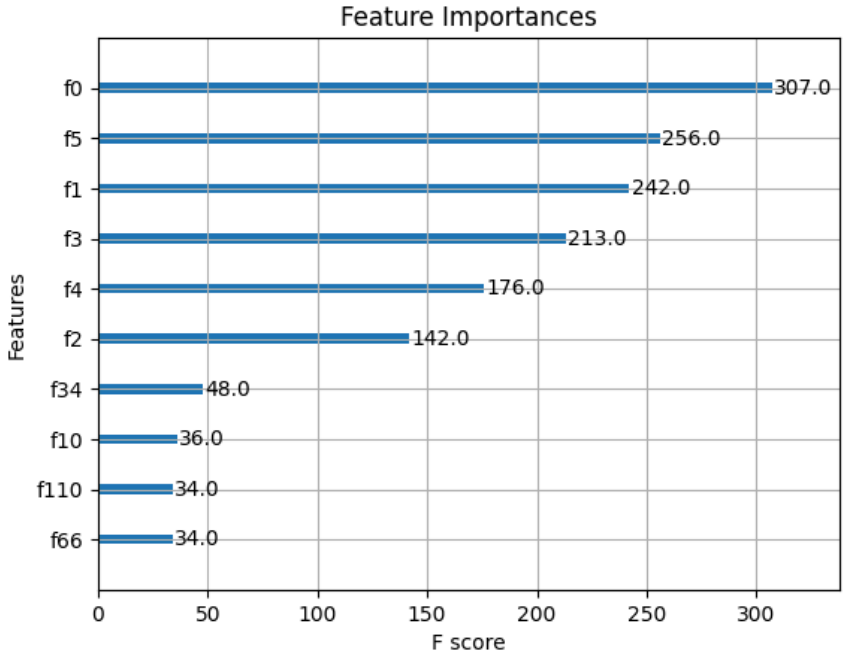
Gradient Boosting was chosen for its effectiveness in handling complex relationships within the data and its ability to boost the performance of weak learners. It is an ensemble learning method that sequentially builds models, each correcting errors from the previous ones, to minimize the overall loss. This iterative process allows Gradient Boosting to produce a robust model that excels in classification tasks.

The Gradient Boosting model was optimized using hyperparameter tuning with GridSearchCV and evaluated using 5-fold cross-validation.
The preprocessing pipeline included scaling numerical features and one-hot encoding categorical features.
The best hyperparameters, including a learning rate of 0.1, a maximum depth of 5, and 100 estimators, achieved a test accuracy of 86.3%.

The classification report shows:
Strong performance in predicting the majority class (<=50K) with high precision, recall, and F1-score.
Challenges with the minority class (>50K), achieving 35% recall and 45% F1-score.
The confusion matrix indicates a higher rate of false negatives for the minority class, reflecting the class imbalance issue.

```
Preprocessing data...
Training XGBoost model with hyperparameter tuning...
Fitting 5 folds for each of 27 candidates, totalling 135 fits

Best Parameters: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 100}
Best Cross-Validation Accuracy: 0.864

Test Accuracy: 0.863

Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.96      0.92     12271
           1       0.65      0.35      0.45      2382

    accuracy                           0.86     14653
   macro avg       0.76      0.66      0.69     14653
weighted avg       0.85      0.86      0.85     14653
```


Confusion Matrix
11815
456
1551
831
0
1
Actual
0
1
Predicted
10000
8000
6000
4000
2000


Feature Importances
f0
f5
f1
f3
f4
f2
f34
f10
f110
f66
307.0
256.0
242.0
213.0
176.0
142.0
48.0
36.0
34.0
34.0
Features
0
50
100
150
200
250
300
F score

Hierarchical Clustering

Analysis:

Determining Optimal Clusters: To identify the optimal number of clusters for hierarchical clustering, the following techniques have been employed:

1. Dendrogram Analysis:

The dendrogram provides a visual representation of the hierarchical clustering process, including the distances at which clusters merge. By investigating the largest vertical gaps-which reflect the largest merging distances, the optimal number of clusters was 3, representing a balance between cluster differentiation and interpretability.

2. Silhouette Analysis:

The silhouette score was calculated to judge the cohesion of the clusters and their separation. A range of 3-5 clusters showed good clustering, with 3 clusters being the clearest and most distinct groupings. This eventually led to a 3-cluster solution that was also quite consistent with the dendrogram and guaranteed well-defined groups to analyze.

Cluster Characteristics

The hierarchical clustering algorithm uncovered three clusters with the following key features:

Cluster 0 (Largest, 21,056 data points):

- Youngest average age.
- Lowest average fnlwgt(final weight).
- Moderate levels of education and income attributes.
- Primarily associated with entry-level or less-educated individuals.

Cluster 1 (Second largest, 19,824 data points):

- Oldest average age.
- Highest education levels and income-related attributes.
- Represents well-established individuals with significant career achievements.

Cluster 2 (Smallest, 3,345 data points):

- Middle-range average age.
- Moderate education levels.
- Associated with middle-income and mid-career individuals.

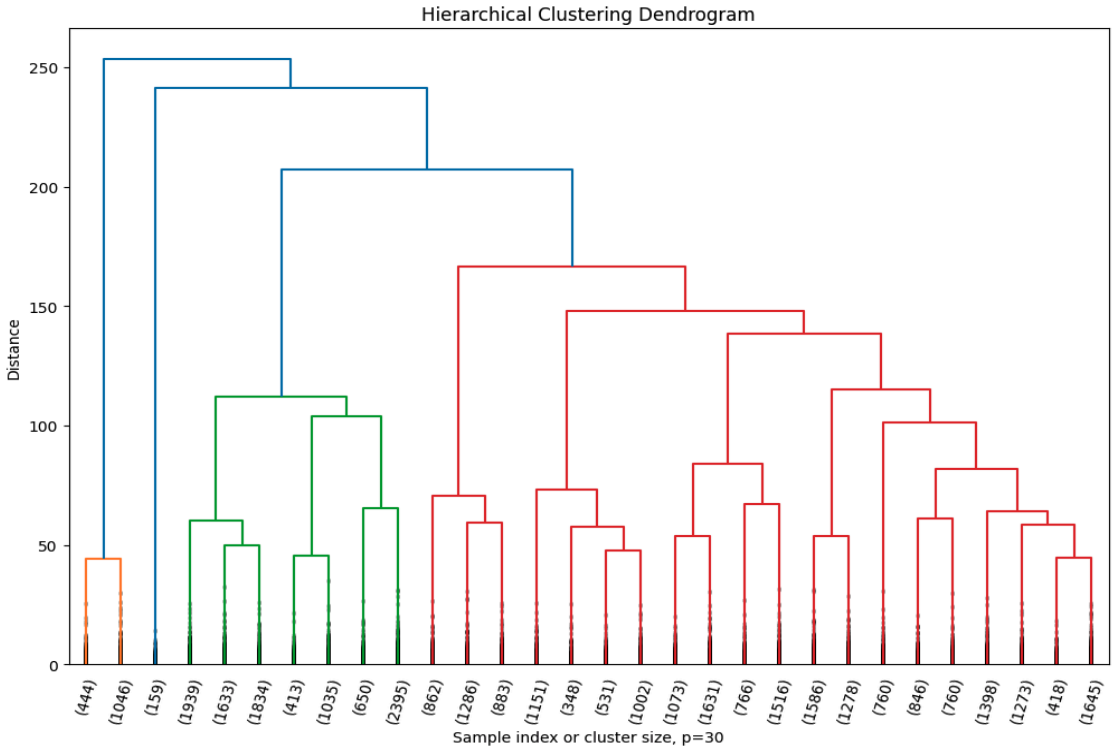
Cluster Validation

To validate the hierarchical clustering solution, the following metrics were used:

1. Silhouette Score: A score of 0.115 suggested a reasonable clustering structure, showing that all the data points within each cluster were well grouped with good separation from other clusters.

2. Calinski-Harabasz Score: A high value of 6,842.719 demonstrated the denser and well-separated clusters, establishing the correctness of choosing a 3-cluster model for the study.

Visualization:

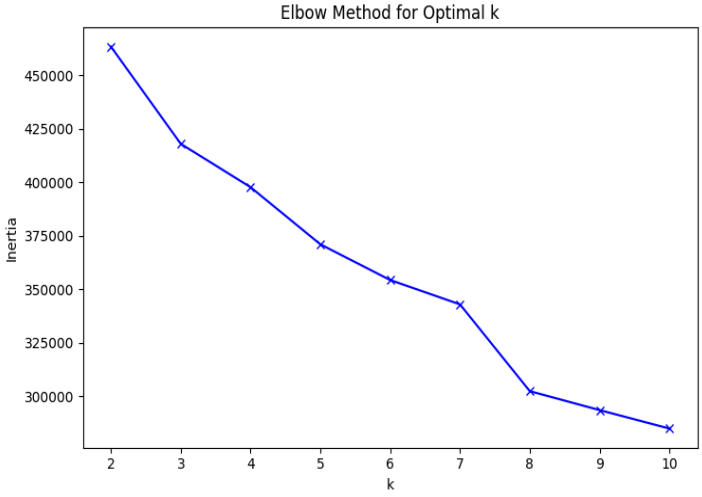


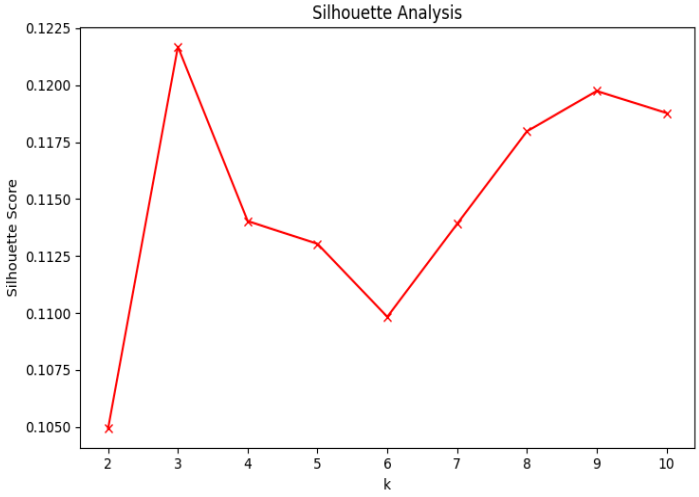
Finally, PCA was used to obtain a lower-dimensional representation that may effectively visualize the dataset. The obtained 2D plot clearly depicted separation between the three clusters and hence very distinctive group characteristics in each cluster, which establishes the interpretability and structure present in the hierarchical clustering solution.

K-Means Clustering Analysis

Determining Optimal Clusters

To determine the appropriate number of clusters, two key techniques were employed:





1. **Elbow Method**:
    - Examined the "elbow" in the within-cluster sum of squares (WCSS) curve.
    - Suggested an optimal range of **3-5 clusters** based on diminishing WCSS reductions.
2. **Silhouette Analysis**:
    - Evaluated the cohesion and separation of clusters using the silhouette score.
    - Indicated that **3-5 clusters** provided strong clustering performance.

Ultimately, a **3-cluster solution** was selected, balancing clarity and interpretability in customer segmentation.

Cluster Characteristics

The K-Means algorithm identified three clusters with the following key attributes:

- **Cluster 0** *(Largest, 21,444 data points)*:
    - Youngest average age.
    - Lowest average fnlwgt (final weight).
    - Lowest average education level.
    - Lowest average capital gain.
- **Cluster 1** *(Second largest, 25,261 data points)*:
    - Oldest average age.

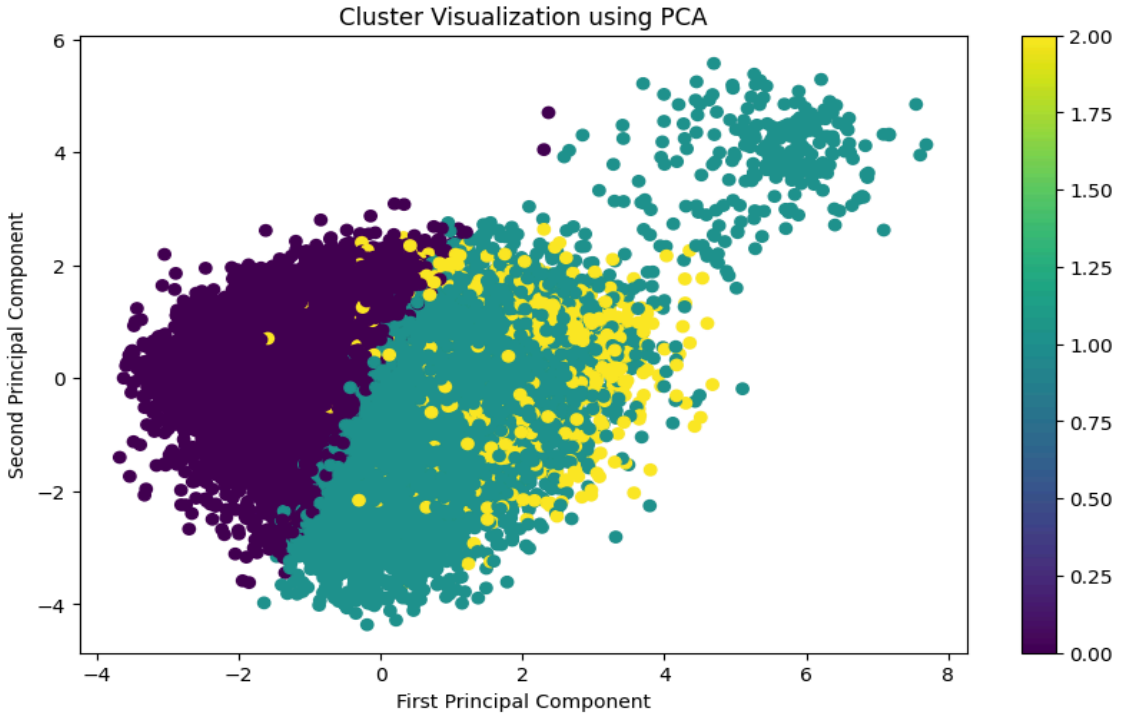
    - Highest average fnlwgt.
    - Highest average education level.
    - Highest average capital gain.
- **Cluster 2** *(Smallest, 2,237 data points)*:
    - Middle-range average age.
    - Lowest average fnlwgt.
    - Middle-range education level.
    - Lowest average capital gain.

Cluster Validation

Two metrics were used to validate the clustering solution:

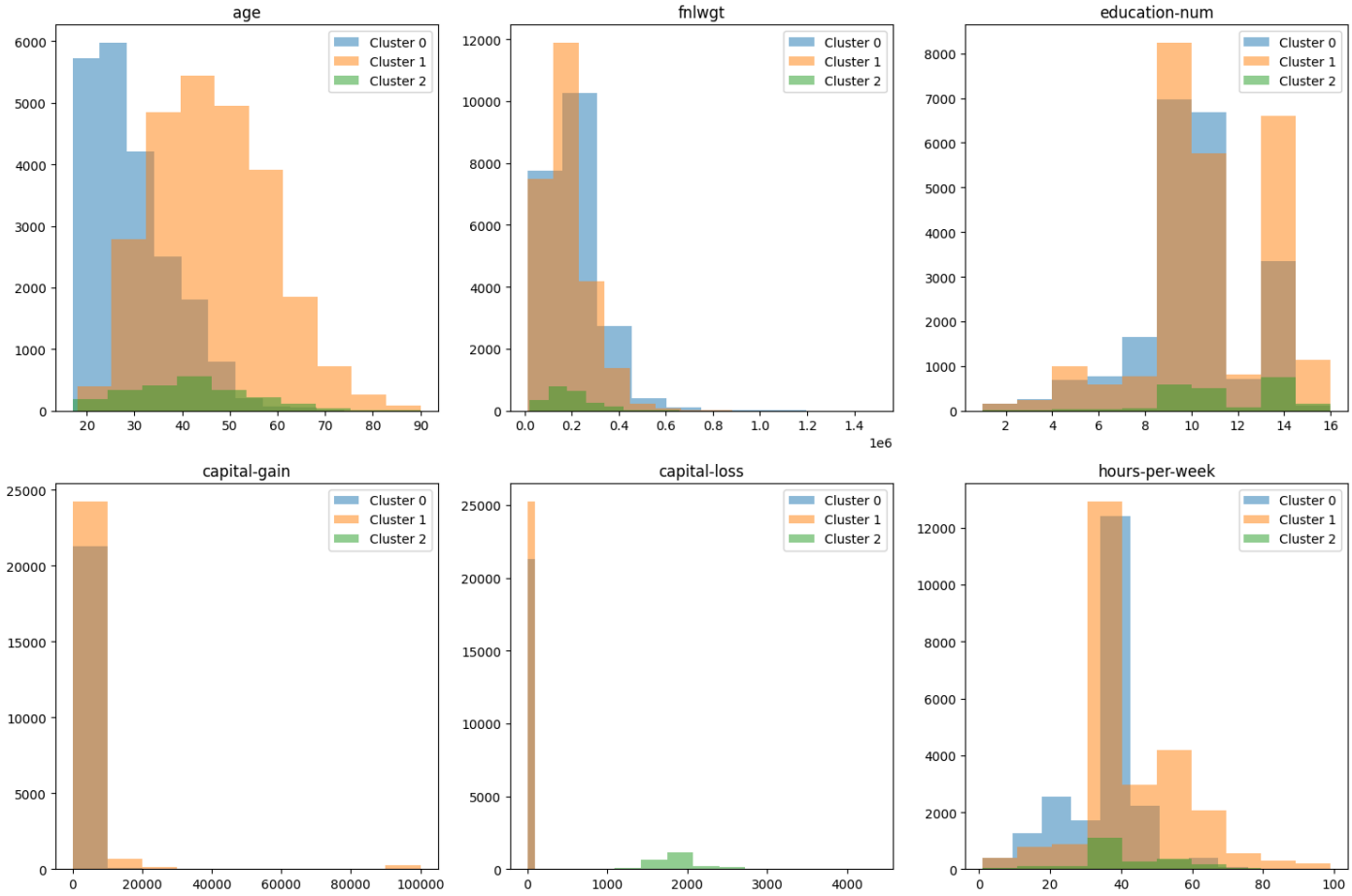
1. **Silhouette Score**:
    - A score of **0.122** indicated a reasonably structured clustering.
2. **Calinski-Harabasz Score**:
    - A score of **5561.468** confirmed the density and separation of clusters, supporting the validity of the 3-cluster solution.

Visualization



To visualize the clustering results, **Principal Component Analysis (PCA)** was applied to reduce dimensionality. A 2D plot of the clusters demonstrated clear separation and distinct characteristics among the three groups.

Feature Distribution



The K-Means clustering analysis identified three well-defined customer segments, each with distinct demographic and financial traits. These clusters offer actionable insights to:

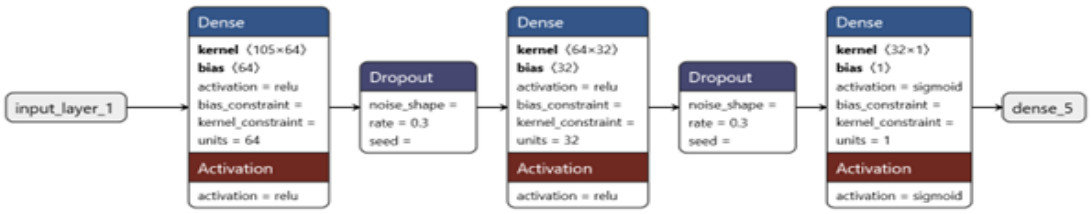
- Develop targeted marketing strategies.
- Optimize product offerings.
- Enhance the overall customer experience.

Future steps could include analyzing the relationships between cluster characteristics and outcome variables to derive deeper business insights.

Neural Network Model

The Neural Network implementation was performed using TensorFlow / Keras libraries. The objective was to predict whether an individual's income exceeds $50,000 based on various demographic and work-related attributes.

The neural network was designed as a feedforward sequential model with the following structure:

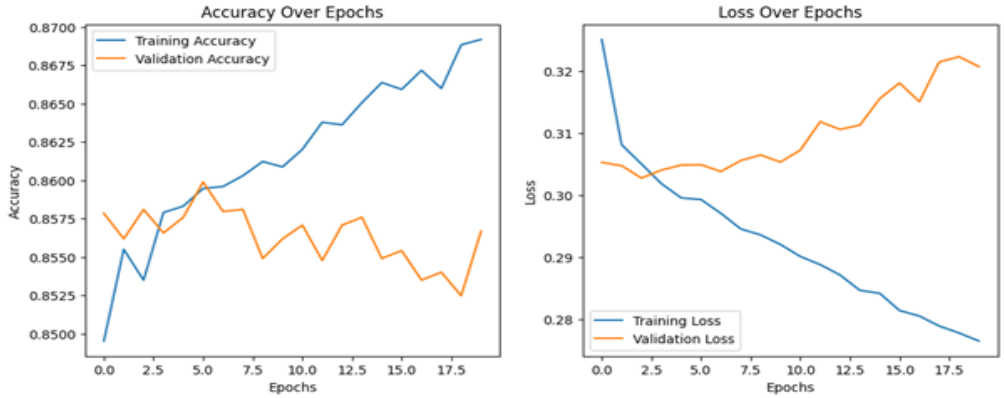


1. **Input Layer:**
    - Accepts data of shape matching the feature dimension of the dataset after preprocessing.
    - **Input Dimension:** Number of encoded features.
2. **Hidden Layers:**
    - **First Hidden Layer:**
        - Neurons: 64
        - Activation: ReLU (Rectified Linear Unit)
        - Dropout Rate: 30% (to prevent overfitting)
    - **Second Hidden Layer:**
        - Neurons: 32
        - Activation: ReLU
        - Dropout Rate: 30%
3. **Output Layer:**
    - Neurons: 1
    - Activation: Sigmoid (for binary classification)

Compilation and Training

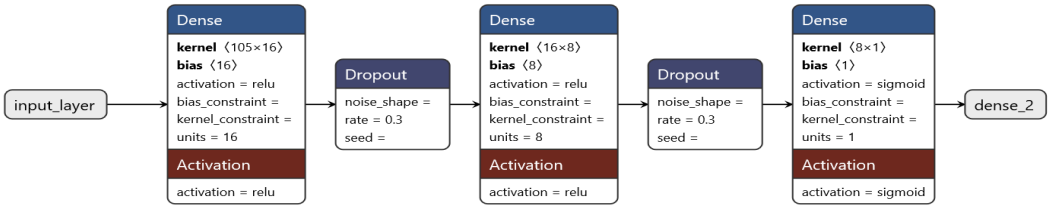
- **Optimizer:** Adam (Adaptive Moment Estimation)
- **Loss Function:** Binary Crossentropy
- **Metrics:** Accuracy
- **Hyperparameters:**
    - Batch Size: 64
    - Epochs: 20
    - Validation Split: 20%

Training and validation loss/accuracy curves were generated to analyze model performance over epochs:



As observed from the graphs above, the training accuracy is increasing and training loss is decreasing over epochs. On the other hand, the validation accuracy is decreasing, and validation loss is increasing. This strongly suggests that the model is overfitting despite implementing dropout.

Our hypothesis was that the model may be too complex, so we tried to implement a simpler model with the following architecture:



Using Early Stopping, we found the optimal number of epochs as 11, so for this model, we reduced the number of epochs.

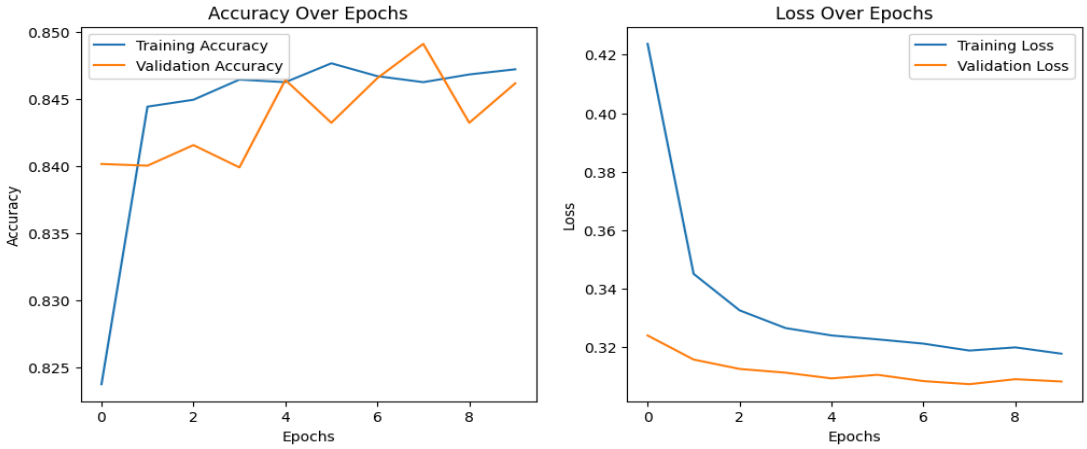
4. **Input Layer:**
    - Accepts data of shape matching the feature dimension of the dataset after preprocessing.
    - **Input Dimension:** Number of encoded features.
5. **Hidden Layers:**
    - **First Hidden Layer:**
        - Neurons: 16
        - Activation: ReLU (Rectified Linear Unit)
        - Dropout Rate: 30% (to prevent overfitting)
    - **Second Hidden Layer:**
        - Neurons: 8
        - Activation: ReLU
        - Dropout Rate: 30%

6. **Output Layer:**
    - Neurons: 1
    - Activation: Sigmoid (for binary classification)

Compilation and Training

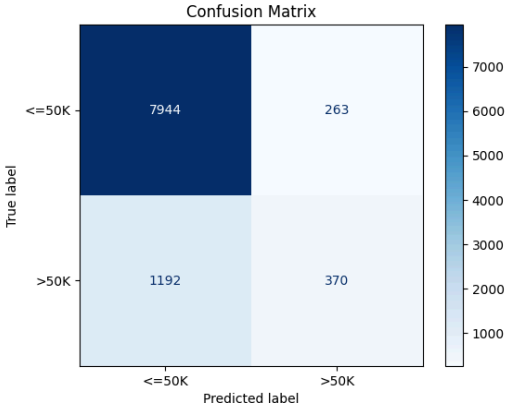
- **Optimizer:** Adam (Adaptive Moment Estimation)
- **Loss Function:** Binary Crossentropy
- **Metrics:** Accuracy
- **Hyperparameters:**
    - Batch Size: 64
    - Epochs: 10
    - Validation Split: 20%

Training and validation loss/accuracy curves were generated to analyze model performance over epochs:



As observed, the training and validation metrics now behave in expected manner, i.e. they improve steadily. The test accuracy for this model comes out to be 84.6%

The confusion matrix and classification report generated for this model is as follows:




```
Classification Report:

              precision    recall  f1-score   support

       <=50K       0.87      0.97      0.92      8207
        >50K       0.58      0.24      0.34      1562

    accuracy                           0.85      9769
   macro avg       0.73      0.60      0.63      9769
weighted avg       0.82      0.85      0.82      9769
```

Conclusion and Future scope:

This project implemented exploratory data analysis and multiple machine learning algorithms, including K-Means Clustering, Gradient Boosting, Hierarchical Clustering, Neural Networks, and Random Forests, to explore and benchmark different approaches for classification. The future scope for the project includes:

1. **Improvement in Clustering Methods:**
    - Explore density-based clustering techniques like DBSCAN to handle noise and outliers better.
2. **Cross-Algorithm Comparison:**
    - Conduct rigorous cross-validation and statistical testing to compare algorithmic performance.
    - Evaluate ensemble combinations of clustering and classification methods for robust solutions.
3. **Real-World Applications:**
    - Extend the project to include time-series analysis for temporal attributes like work hours or age progression.

References:

1. https://archive.ics.uci.edu/dataset/2/adult
2. https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf
3. https://arxiv.org/pdf/1002.2425
4. https://www.ripublication.com/irph/ijict_spl/14_ijictv3n11spl.pdf
5. https://arxiv.org/abs/1811.12808
6. https://jerryfriedman.su.domains/ftp/trebst.pdf