HBS Survey Project

Comprehensive Analysis Report

Generated on: March 06, 2025, 20:46:47

Research Project Based on Data Analysis

Table of Contents

The table of contents will be automatically generated.

Executive Summary

Executive Summary: **Objective:** The project aimed to analyze and extract insights from clustered data, focusing on feature importance to enhance understanding of program characteristics. **Analytical Methods:** Utilized machine learning techniques such as XGBoost for feature analysis, UMAP for clustering, and statistical tests for validation. Deep dive into feature importance using SHAP analysis provided crucial insights. **Key Findings:** Identified top program characteristics driving clustering patterns, highlighting important features for program categorization. Revealed significant performance metrics through model comparisons, deepening understanding of program distinctions. **Recommendations:** Implement insights gained on feature importance to tailor program offerings more effectively. Consider utilizing clustering methods for more precise program segmentation. Enhance model performance based on identified key features to optimize program outcomes. Continuously evaluate and update program characteristics based on clustering results for improved decision-making.

Project Overview

Project Description: This data science project aims to analyze and derive insights from a dataset stored in the 'Data' directory by utilizing various machine learning techniques. The primary objective is to cluster data points based on their characteristics and determine feature importance for predictive modeling. **Main Components:** 1. **Code:** Includes scripts for clustering, feature importance analysis, plotting, and other data processing tasks. 2. **Data:** Contains the raw and processed datasets for analysis. 3. **Output:** Stores organized results such as cluster analysis reports, feature importance results, and associated images. **Analytical Workflow:** The project workflow can be inferred from the directory structure: - The 'Old' directory in 'Code' houses previous versions of scripts and analysis notebooks. - Specific tasks like xgboost feature analysis, UMAP clustering, and program feature analysis are present. - The 'Organized_Results' subdirectory under 'Output' contains detailed reports, result summaries, and visualization images for both clustering and feature importance analyses. **Results:** 1. Cluster Analysis Results: - Cluster analysis reports highlighting patterns and groupings within the data. - Visual representations of clustering methods and program comparisons. 2. Feature Importance Results: - Reports evaluating the significance of features in predictive modeling. - Summary of top features and model performance metrics. 3. Statistical Outputs: - CSV files with detailed statistics on cluster results, confusion matrices, and misclassification probabilities. Overall, this project showcases a structured approach to data analysis, from data preprocessing to clustering and feature importance analysis, leading to actionable insights for decision-making.

Key File Analysis

Cluster Analysis: Cluster_Analysis.ipynb

Notebook Analysis: Cluster_Analysis.ipynb

Notebook contains 26 cells (13 markdown, 13 code)

Notebook Structure

• Clustering Analysis

• Util

• 1. Import Libraries and Load Data

• 2. Data Loading and Preprocessing

• 3. Perform K-Means Clustering Directly on Data

• 4. Apply UMAP for Visualization

• 5. Visualize Clusters with UMAP

• 6. Analyze Cluster Composition by Program Type

• 7. Create Feature Importance Visualization

• 8. Export Final Results

(... 3 more headings not shown ...)

Notebook Technical Content

Key libraries: fails, GaussianMixture, numpy, pyreadstat, Axes3D

Custom functions: generate_comprehensive_statistics, evaluate_clusters_against_programs, analyze_feature_importance, apply_umap_for_visualization, clean_column_name

Visualization methods: figure, plot, plt, bar, sns

Analysis techniques: DBSCAN, KMeans, RandomForest, cluster

Data operations: .join, .value_counts(), pd.read_, DataFrame

Notebook Summary:

The notebook focuses on performing clustering analysis on survey data using K-Means clustering and UMAP for visualization. Key analyses and visualizations include:

• Importing libraries and loading data: Essential libraries such as pandas, numpy, and matplotlib are imported. The data is loaded and preprocessed, including reading data from a Stata file.

• Data loading and preprocessing: The data is loaded with clusters already generated, and variable labels are extracted for further analysis if available.

• Perform K-Means Clustering Directly on Data: K-Means clustering is applied directly to the data, and UMAP is used for visualization. The results are then analyzed and visualized to understand the underlying patterns or structures.

Overall, the notebook aims to cluster survey data using K-Means clustering and visualize the results using UMAP, providing insights into potential groupings or patterns within the data.

Feature Importance Analysis: Feature_Importance.ipynb

Notebook Analysis: Feature_Importance.ipynb

Notebook contains 18 cells (8 markdown, 10 code)

Notebook Structure

• Feature Importance Analysis: Distinguishing Between Upskilling and Reskilling Programs

• Data Loading and Preparation

• Analysis 1: Complete Dataset (Including Outcomes)

• Analysis 2: Program Characteristics Only (Excluding Outcomes)

• Analysis 3: Program Characteristics with Original Categorical Variables (No Dummies)

• Analysis 4: All Variables Without Outcomes (Using Dummies)

• Appendix: Additional Model Results

• Util

Notebook Technical Content

Key libraries: numpy, pyreadstat, openpyxl, traceback, WD_ALIGN_PARAGRAPH

Custom functions: format_excel_worksheet, create_word_document, create_target, align_features, preprocess_data

Visualization methods: figure, plot, plt, bar, sns

Analysis techniques: train_test_split, XGBoost, model.fit

Data operations: .join, .value_counts(), pd.read_, DataFrame

Notebook Summary:

Summary:

• Main Purpose/Topic: The notebook focuses on analyzing the feature importance to distinguish between upskilling and reskilling programs. It explores various analyses to identify key differentiating features using different datasets, including outcomes and program characteristics.

• Key Analyses or Visualizations: The notebook involves data loading and preparation, three main analyses: Analysis 1 utilizing the complete dataset, Analysis 2 focusing on program characteristics only, and Analysis 3 considering program characteristics with original categorical variables (no dummies). It includes the use of libraries like numpy, pyreadstat, matplotlib, seaborn, xgboost, and shap for visualizations.

• Main Findings or Conclusions: The notebook sets up directories for storing results and visualizations, sets a random seed for reproducibility, and configures visualization settings. By conducting feature importance analysis on different datasets and variables, it aims to reveal insights into the distinguishing features

between upskilling and reskilling programs, which can be crucial for decision-making and program optimization. Further detailed conclusions can be drawn from the specific analyses conducted in the notebook.

Model Performance Analysis: model_performance_metrics.csv

CSV File: model_performance_metrics.csv

Contains 4 rows and 7 columns

Columns:

model_id, model_name, accuracy, precision, recall, f1, auc

Data preview (first 10 rows):

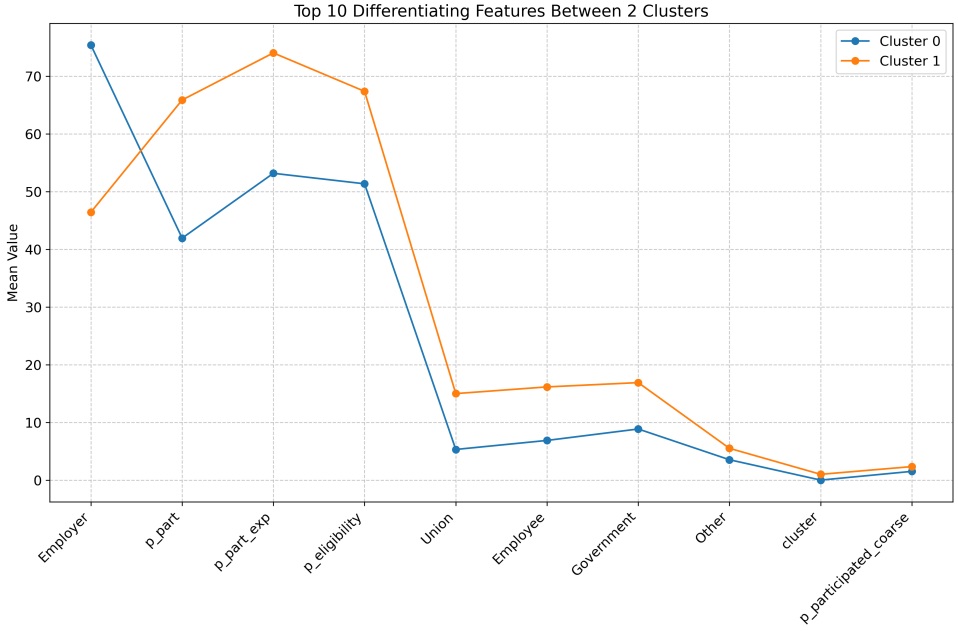
me	accuracy	precision	recall	f1
s (with outcomes)	0.8264462809917356	0.8	0.9185185185185184	0.8551724137931035
eatures (with dummies)	0.6776859504132231	0.659217877094972	0.8740740740740741	0.7515923566878981
eatures (with categorical encoding)	0.7066115702479339	0.6777777777777778	0.9037037037037036	0.7746031746031746
s without outcomes	0.6900826446280992	0.6630434782608695	0.9037037037037036	0.7648902821316614

Possible data interpretation:

The CSV file "model_performance_metrics.csv" likely contains data on performance metrics of different models. The columns suggest that it might include information such as model identifiers, names, and evaluation metrics like accuracy, precision, recall, F1 score, and AUC (Area Under the Curve). Insights from this data could help in comparing the performance of different models in terms of their classification capabilities. It can be valuable for selecting the best performing model for deployment or further optimization based on these metrics. Hypothesis: Based on the metrics provided in the dataset, we can hypothesize that models with high accuracy, precision, recall, F1 score, and AUC values are likely to perform better in classifying the target variable compared to models with lower values in these metrics. The data can be used to identify the strengths and weaknesses of each model and guide decisions on model selection and improvement.

Feature Importance Visualization: feature_importance_2_clusters.png

Image: feature_importance_2_clusters.png



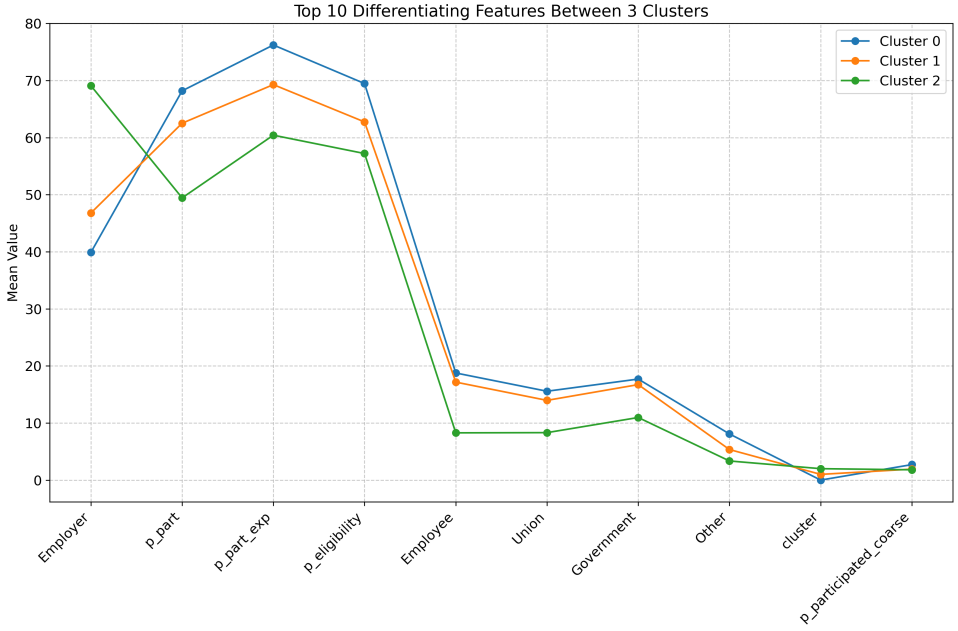
feature_importance_2_clusters.png

Possible interpretation:

Based on the filename "feature_importance_2_clusters.png," this visualization is likely showing the importance or relevance of features in distinguishing or clustering two distinct groups or clusters in a dataset. The plot might display the significance of different variables in predicting or classifying the two clusters, providing insights into which features are most influential in the clustering process.

Feature Importance Visualization: feature_importance_3_clusters.png

Image: feature_importance_3_clusters.png



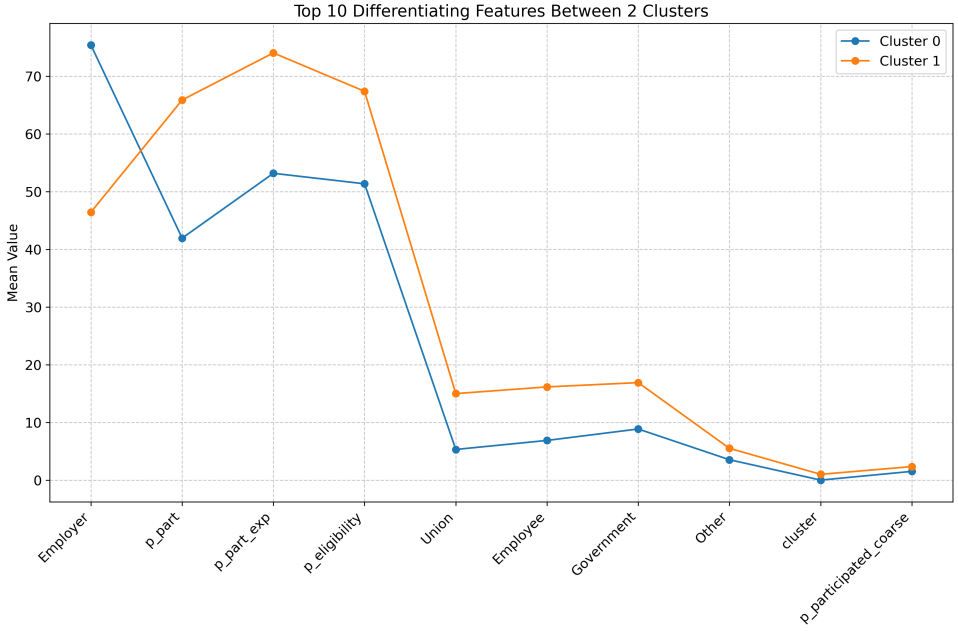
feature_importance_3_clusters.png

Possible interpretation:

Based on the filename "feature_importance_3_clusters.png," this visualization is likely showing the importance of features in predicting or distinguishing between three clusters or groups in a dataset. The plot may display the relative significance of different input variables or features in classifying observations into these three distinct clusters.

Feature Importance Visualization: feature_importance_2_clusters.png

Image: feature_importance_2_clusters.png



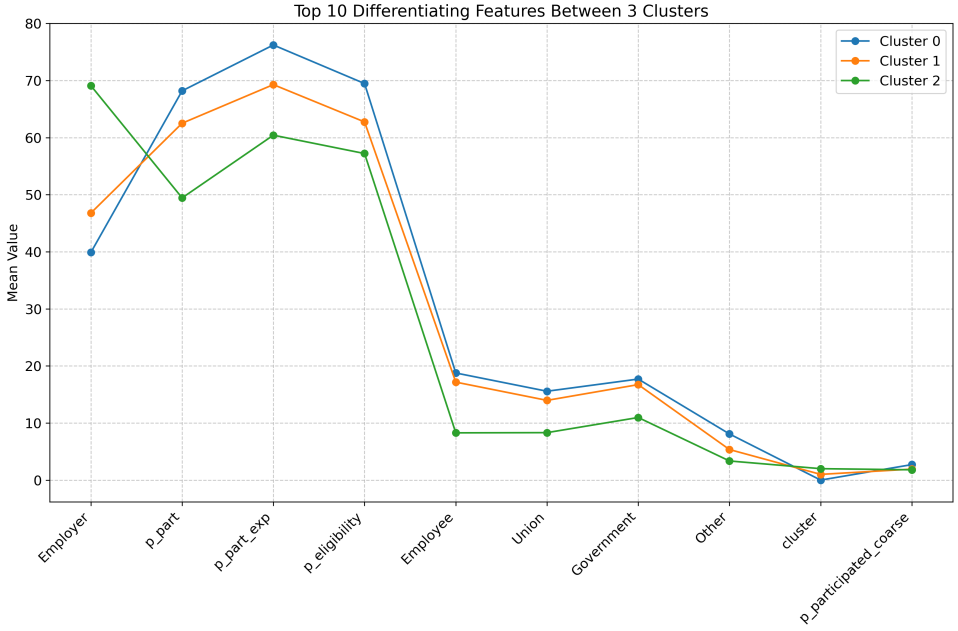
feature_importance_2_clusters.png

Possible interpretation:

Based on the filename "feature_importance_2_clusters.png", this visualization likely shows the importance of features in distinguishing between two clusters or groups in a dataset. It probably displays a comparison of the relative importance of different features in separating the two clusters, which can help in understanding which features are most influential in defining the clusters.

Feature Importance Visualization: feature_importance_3_clusters.png

Image: feature_importance_3_clusters.png



feature_importance_3_clusters.png

Possible interpretation:

Based on the filename "feature_importance_3_clusters.png," it is likely that this visualization is showing the importance of features in distinguishing or clustering data points into three distinct groups or clusters. The plot might display the relative significance of different features in the clustering process, providing insights into which variables are most influential in separating the data into the three clusters.

Code Analysis

Directory contains 8 files and 1 subdirectories

Files:

Cluster.do, Cluster_Analysis.ipynb, Feature_Importance.ipynb, OPENAI_API_KEY.env, Plots.do, Resume.py, report_generator.log, variable_definitions.py

Subdirectories:

Old

File: Cluster.do

Type: .do, Size: 13.15 KB

File: OPENAI_API_KEY.env

Type: .env, Size: 0.17 KB

File: Plots.do

Type: .do, Size: 35.57 KB

File: Resume.py

Code Summary:

This Python code defines a `ProjectReportGenerator` class that analyzes files within a project directory to build context about the project. It categorizes files based on their extensions and content to identify data files, visualization files, model files, and key analyses. The class also sets up logging, initializes context variables, and loads an OpenAI API key. Key functions/classes: - `ProjectReportGenerator`: Class that scans project files to build context and generates a project summary report. - `build_project_context()`: Method to scan project files and categorize them based on types and content. Algorithms/Techniques: - File scanning and categorization based on file extensions and content. - Logging configuration for tracking project analysis progress. - Integration with OpenAI API for further project analysis. Overall, this code prepares to generate a detailed report summarizing the context and content of various files within a project directory.

Code Sample (first 30 lines):

```
import os
import re
```

```
import json
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from fpdf import FPDF
from PIL import Image
import docx
import openpyxl
import markdown
import csv
import nbformat
from dotenv import load_dotenv
import openai
import time
import logging
from reportlab.lib.pagesizes import letter, A4
from reportlab.platypus import SimpleDocTemplate, Paragraph, Spacer, Image as Re
portLabImage, Table, TableStyle
from reportlab.lib.styles import getSampleStyleSheet, ParagraphStyle
from reportlab.lib import colors
from reportlab.lib.units import inch

import glob

from reportlab.lib import colors
from reportlab.lib.pagesizes import letter, A4
from reportlab.platypus import SimpleDocTemplate, Paragraph, Spacer, Image as Re
portLabImage, Table, TableStyle, PageBreak, HRFlowable
from reportlab.lib.styles import getSampleStyleSheet, ParagraphStyle
from reportlab.lib.units import inch
```

(... 1440 more lines not shown ...)

File: report_generator.log

Type: .log, Size: 29.97 KB

File: variable_definitions.py

Code Summary:

This Python code defines a dictionary named `label_mapping` that maps original variable names to more readable labels for a dataset. The dictionary contains mappings for various core variables related to skills, ROI, redeployment, program effectiveness, and more. It also includes additional variables related to firm/organization information such as size, industry, region, ownership, and workforce engagement. Key points: 1. The code enhances the interpretability of the dataset by providing human-readable labels for each variable. 2. The `label_mapping` dictionary serves as a lookup table for transforming original variable names into descriptive labels. 3. This code improves data analysis and visualization tasks by using meaningful labels instead of cryptic variable names, making it easier to understand and interpret the data. No specific algorithms or techniques are used in this code; it primarily focuses on data preprocessing and labeling for better data understanding.

Code Sample (first 30 lines):

```
# variable_definitions.py

# Label mapping for better readability
label_mapping = {
    # Core variables from original mapping
    'b_sk_n_digital': 'Digital Skills Needed',
    'b_sk_n_man': 'Mgmt Skills Needed',
    'roi4': 'Negative ROI',
    'roi_yes': 'ROI Measured',
    'ecp_extredeployment': 'External Redeployment',
    'p_effect_reverse': 'Program Effectiveness',
    'evp_network': 'Cross-departmental Networks',
    'roi5': 'Positive ROI',
    'sum_tr_sk': 'Sum of Trained Skills',
    'sha_b_sk_n_digital': 'Share Needed Digital Skills',
    'stat_government': 'Help Org Use Gov Subsidy',
    'ecp_intredeployment': 'Internal Redeployment',
    'stat_csr': 'Fulfilling CSR Requirement',
    'reason_dei': 'Reason DEI',
    'sk_selected': 'Total Nr skills needed',
    'stand2': 'Mix Standardization customization',
    'f_union_1_-_25%': 'Union Share 1-25%',
    'inc_mgr_nofin': 'Manager: Non-financial Incentive',
    'invest_cont': 'Continued Investment',
    'p_fund_gov': 'Funded by Government',
    'f_medium': 'Medium Firm Size (100-999)',
    'roi2': 'Not yet but intend to calculate ROI',
    'roi3': 'Tried to but unable to',
    'roi1': 'No attempt to calculate',
    'p_eligibility': 'Participation Eligibility',
```

(... 486 more lines not shown ...)

Subdirectory: Old

Directory contains 15 files and 0 subdirectories

Files:

01_xgboost_feature_analysis.ipynb, 02_umap_clustering.ipynb, 03_cluster_statistical_tests.ipynb, 04_program_feature_analysis.ipynb, 05_Feature_Importance_Program_Chars.ipynb, 06_Correction_Cluster-Mapping.ipynb, 07_Feature_Importance.ipynb, 07_Organized_Cluster_Analysis.ipynb, 1_all_data.ipynb, 2_cluster_data_creation.ipynb, 3_Clusters_Complete_Analysis.ipynb, 4_program_characteristics.ipynb, Cluster analysis.do, Update_labels.ipynb, dimension_reduction.py

Notebook Analysis: 01_xgboost_feature_analysis.ipynb

Notebook contains 10 cells (5 markdown, 5 code)

Notebook Structure

• Feature Importance

• 1. Data Loading and Initial Exploration

• 2. Data Preprocessing and Cleaning

• 3. Feature Importance Analysis using XGBoost

• 4. SHAP Value Analysis for Model Interpretability

Notebook Technical Content

Key libraries: numpy, collections, pyreadstat, pandas, matplotlib

Custom functions: iterative_feature_selection, add_unique_keys, clean_cols, train_xgboost_model, split_data

Visualization methods: plot, plt, sns, figure

Analysis techniques: train_test_split, XGBoost, model.fit

Data operations: .value_counts(), pd.read_, DataFrame

Notebook Summary:

• The main purpose of the notebook is to analyze feature importance in distinguishing between upskilling and reskilling programs using XGBoost and SHAP values. The notebook covers data loading, preprocessing, feature importance analysis using XGBoost, and SHAP value analysis for model interpretability.

• Key analyses and functions in the notebook include: - Data loading with metadata interpretation using `pyreadstat` - Initial exploratory analysis on program characteristics and distributions - Preprocessing steps such as creating dummy variables, cleaning column names, and target variable preparation - Feature importance analysis using XGBoost - SHAP value analysis for model interpretability

• The findings may include the identification of important features that distinguish between upskilling and reskilling programs, insights into the distribution and characteristics of the programs, and an understanding of how each feature contributes to the model's predictions through SHAP values. Further analysis may provide insights for decision-making in designing effective upskilling and reskilling programs based on the identified important features.

Notebook Analysis: 02_umap_clustering.ipynb

Notebook contains 7 cells (4 markdown, 3 code)

Notebook Structure

• Clustering Analysis for Survey Data

• 1. Data Loading and Preprocessing

• 2. Dimensionality Reduction and Clustering

• 3. Cluster Analysis and Export

Notebook Technical Content

Key libraries: numpy, pyreadstat, pandas, matplotlib, KMeans

Custom functions: perform_clustering_analysis

Visualization methods: plot, plt, scatter, figure

Analysis techniques: KMeans, cluster

Data operations: .value_counts(), pd.read_

Notebook Summary:

• Main Purpose/Topic: The notebook focuses on conducting clustering analysis on survey data, utilizing UMAP for dimensionality reduction and K-Means for clustering to uncover patterns and groupings within the data.

• Key Analyses or Visualizations: - Data loading and preprocessing, including creating dummy variables and cleaning column names. - UMAP dimensionality reduction to visualize the data. - Determining the optimal number of clusters using the elbow method and silhouette scores. - Visualizations of evaluation metrics such as the elbow plot and silhouette plot.

• Main Findings or Conclusions: Based on the evaluation metrics, such as the within-cluster sum of squares (WCSS) and silhouette scores, one can decide on an appropriate number of clusters for the dataset. Additionally, visualizations like the elbow plot can help in identifying the optimal number of clusters by looking for an "elbow point" where the rate of decrease in WCSS slows down. The notebook provides insights into structuring and interpreting survey data through clustering analysis.

Notebook Analysis: 03_cluster_statistical_tests.ipynb

Notebook contains 10 cells (5 markdown, 5 code)

Notebook Structure

• Statistical Analysis of Clustering Results

• 1. Setup and Data Loading

• 2. Three-Cluster Analysis

• 3. Two-Cluster Analysis

• 4. Cluster and Program Type Comparison

• Files Generated

Notebook Technical Content

Key libraries: numpy, f_oneway, scipy, pandas, matplotlib

Custom functions: plot_density_distributions, analyze_clusters, create_statistical_table, process_results, es_dummy

Visualization methods: figure, plot, plt, sns, scatter

Analysis techniques: KMeans, cluster

Data operations: .join, pd.read_, DataFrame

Notebook Summary:

Summary:

• Main Purpose/Topic: The notebook aims to conduct statistical analysis of clustering results using ANOVA for 3-cluster solutions and t-tests for 2-cluster solutions. It includes evaluations with and without dummy variables for result robustness.

• Key Analyses or Visualizations: • Setup and Data Loading: Import libraries, load data, and configure visualization settings. • Three-Cluster Analysis: ANOVA testing for all variables, generation of statistical tables, density plots for significant variables, and analysis without dummy variables. • Two-Cluster Analysis: To be completed. • Cluster and Program Type Comparison: Additional analyses comparing clusters and program types.

• Main Findings or Conclusions: The findings from ANOVA and t-test analyses will help identify significant variables among different clusters and program types, aiding in understanding the clustering results' underlying patterns. The statistical tables and plots generated in the notebook will provide insights into the relationships between variables and clusters, contributing to a more comprehensive interpretation of the clustering analysis results.

Notebook Analysis: 04_program_feature_analysis.ipynb

Notebook contains 7 cells (4 markdown, 3 code)

Notebook Structure

• Feature Importance Analysis (Program Characteristics Only)

• 1. Data Preparation and Feature Selection

• 2. Feature Importance Analysis with XGBoost

• 3. SHAP Value Analysis

Notebook Technical Content

Key libraries: roc_curve, numpy, pyreadstat, pandas, matplotlib

Custom functions: model_loop, get_remove_features, remove_features

Visualization methods: plot, plt, figure

Analysis techniques: train_test_split, XGBoost

Data operations: pd.read_

Notebook Summary:

• Main Purpose/Topic: The notebook aims to perform a feature importance analysis on distinguishing features between upskilling and reskilling programs. It focuses exclusively on analyzing program characteristics while excluding outcome variables.

• Key Analyses or Visualizations: - Data preparation including loading, cleaning, and feature selection. - XGBoost modeling with cross-validation for feature importance analysis. - ROC curve analysis for model evaluation. - SHAP (SHapley Additive exPlanations) value analysis for interpreting feature importance.

• Main Findings or Conclusions: The notebook carries out a comprehensive analysis to identify the influential program characteristics that differentiate between upskilling and reskilling programs. By employing XGBoost modeling and SHAP analysis, the notebook provides insights into the relative importance of different program-specific variables in predicting the program types. Additionally, the ROC curve analysis helps evaluate the model's performance, and the precision, recall, and F1 scores could further assess the model's predictive ability.

Notebook Analysis: 05_Feature_Importance_Program_Chars.ipynb

Notebook contains 7 cells (4 markdown, 3 code)

Notebook Structure

• Feature Importance Analysis (Program Characteristics Only)

• 1. Data Preparation and Feature Selection

• 2. Feature Importance Analysis with XGBoost

• 3. SHAP Value Analysis

Notebook Technical Content

Key libraries: roc_curve, numpy, pyreadstat, pandas, matplotlib

Custom functions: model_loop, get_remove_features, remove_features

Visualization methods: plot, plt, figure

Analysis techniques: train_test_split, XGBoost

Data operations: pd.read_

Notebook Summary:

Summary: The notebook focuses on analyzing feature importance related to upskilling and reskilling programs based on program characteristics, excluding outcome variables. The key aspects include data preparation, feature selection, XGBoost modeling with cross-validation, and SHAP value analysis.

Key Analyses/Visualizations: • Data preparation involved loading and preparing the dataset, creating dummy variables, and selecting specific program variables. • Feature selection included removing outcome variables, splitting data for training and validation, and setting visualization parameters. • XGBoost modeling was performed for feature importance analysis on program-specific variables, along with evaluating model performance using ROC curve, precision, recall, and F1 score metrics. • SHAP value analysis was conducted for interpreting the impact of features on the model predictions.

Findings/Conclusions: The analysis aims to identify significant program characteristics that distinguish between upskilling and reskilling programs. By leveraging XGBoost and SHAP analysis, the notebook provides insights into which program-specific variables contribute most to the distinction. This information can help stakeholders better understand key factors influencing program outcomes and improve decision-making in designing effective upskilling and reskilling initiatives.

Notebook Analysis: 06_Correction_Cluster-Mapping.ipynb

Notebook contains 26 cells (14 markdown, 12 code)

Notebook Structure

• Clustering Analysis

• Cluster Analysis and Statistical Comparison

• Util

• 1. Import Libraries and Load Data

• 2. Data Loading and Preprocessing

• 3. Perform K-Means Clustering Directly on Data

• 4. Apply UMAP for Visualization

• 5. Visualize Clusters with UMAP

• 6. Analyze Cluster Composition by Program Type

• 7. Create Feature Importance Visualization

(... 13 more headings not shown ...)

Notebook Technical Content

Key libraries: GaussianMixture, numpy, pyreadstat, Axes3D, openpyxl

Custom functions: generate_comprehensive_statistics, evaluate_clusters_against_programs, analyze_feature_importance, apply_umap_for_visualization, clean_column_name

Visualization methods: figure, plot, plt, bar, sns

Analysis techniques: DBSCAN, KMeans, RandomForest, cluster

Data operations: .join, .value_counts(), pd.read_, DataFrame

Notebook Summary:

This Jupyter notebook focuses on clustering analysis of survey data, utilizing K-Means for clustering and UMAP for visualization. The main purpose is to identify patterns and groupings within the data.

Key analyses and visualizations include: • K-Means clustering applied to the data and visualized using UMAP • Visualization of program type distribution using UMAP

The notebook involves loading and preprocessing data, creating dummy variables, and analyzing clusters already generated in the data. Key libraries used are GaussianMixture, numpy, pyreadstat, Axes3D, and openpyxl. Standard scaling and silhouette scores are used to evaluate the clustering performance.

The main findings or conclusions may include insights into the grouping of survey data based on the features considered and the visualization of cluster distributions. Additionally, the notebook may provide information on the distribution of program types within the dataset and how they relate to the clustering results.

Notebook Analysis: 07_Feature_Importance.ipynb

Notebook contains 15 cells (7 markdown, 8 code)

Notebook Structure

• Feature Importance Analysis: Distinguishing Between Upskilling and Reskilling Programs

• Data Loading and Preparation

• Analysis 1: Complete Dataset (Including Outcomes)

• Analysis 2: Program Characteristics Only (Excluding Outcomes)

• Analysis 3: Program Characteristics with Original Categorical Variables (No Dummies)

• Analysis 4: All Variables Without Outcomes (Using Dummies)

• Appendix: Additional Model Results

Notebook Technical Content

Key libraries: numpy, pyreadstat, pandas, matplotlib, seaborn

Custom functions: preprocess_data, create_target

Visualization methods: figure, plot, plt, bar, sns

Analysis techniques: train_test_split, XGBoost, model.fit

Data operations: .value_counts(), pd.read_, DataFrame

Notebook Summary:

• Main Purpose/Topic of the Notebook: The notebook aims to conduct a feature importance analysis to distinguish between upskilling and reskilling programs. It includes data loading, preparation, and three separate analyses using different subsets of variables.

• Key Analyses or Visualizations: - Data Loading and Preparation: Loading the dataset and preparing it for analysis. - Analysis 1: Complete Dataset (Including Outcomes): Identifying differentiating features between upskilling and reskilling programs using all variables, including outcomes. - Analysis 2: Program Characteristics Only (Excluding Outcomes): Analyzing program characteristics excluding outcomes. - Analysis 3: Program Characteristics with Original Categorical Variables (No Dummies): Considering program characteristics with original categorical variables (no dummy variables).

• Main Findings or Conclusions: The notebook likely presents insights into the features that are most important in distinguishing between upskilling and reskilling programs. By conducting various analyses, such as including or excluding outcome variables and original categorical variables, the notebook aims to provide a comprehensive understanding of key factors that differentiate these two types of programs. The use of XGBoost and SHAP (SHapley Additive exPlanations) for feature importance might help in uncovering meaningful insights.

Notebook Analysis: 07_Organized_Cluster_Analysis.ipynb

Notebook contains 27 cells (14 markdown, 13 code)

Notebook Structure

• Clustering Analysis

• Cluster Analysis and Statistical Comparison

• Util

• 1. Import Libraries and Load Data

• 2. Data Loading and Preprocessing

• 3. Perform K-Means Clustering Directly on Data

• 4. Apply UMAP for Visualization

• 5. Visualize Clusters with UMAP

• 6. Analyze Cluster Composition by Program Type

• 7. Create Feature Importance Visualization

(... 13 more headings not shown ...)

Notebook Technical Content

Key libraries: GaussianMixture, numpy, pyreadstat, Axes3D, openpyxl

Custom functions: generate_comprehensive_statistics, evaluate_clusters_against_programs, analyze_feature_importance, apply_umap_for_visualization, clean_column_name

Visualization methods: figure, plot, plt, bar, sns

Analysis techniques: DBSCAN, KMeans, RandomForest, cluster

Data operations: .join, .value_counts(), pd.read_, DataFrame

Notebook Summary:

• The main purpose of the notebook is to perform clustering analysis on survey data using K-Means clustering and UMAP for visualization.

• The notebook contains analyses related to clustering including creating various directories for organizing results, importing necessary libraries, loading and preprocessing data, and performing K-Means clustering. Additionally, it includes visualizations using UMAP to plot different aspects of the dataset.

• The key analyses and visualizations include the creation of directory structures for clustering results, loading and preprocessing survey data, performing K-means clustering on the data, and visualizing the clusters and program type distribution using UMAP. The notebook also sets global plotting parameters and loads the data stored in a DTA file. No specific findings or conclusions are apparent in the provided content snippet.

Notebook Analysis: 1_all_data.ipynb

Notebook contains 25 cells (4 markdown, 21 code)

Notebook Structure

• Analysis using all the variables

• Top 10 features

• Top 20 features

• Create shap values (this code could take some time to run)

Notebook Technical Content

Key libraries: roc_curve, numpy, collections, pyreadstat, pandas

Custom functions: add_unique_keys, get_remove_features, remove_features, model_loop, remove_none_from_list

Visualization methods: plot, plt, figure

Analysis techniques: train_test_split, XGBoost

Data operations: pd.read_

Notebook Summary:

The notebook focuses on feature importance analysis using XGBoost and SHAP values. It includes sections for analyzing all variables, identifying the top 10 and top 20 features, and creating SHAP values. The primary purpose is to understand the importance of different features in a dataset for predictive modeling.

Key analyses and visualizations involve using XGBoost to train a model on the dataset, calculating ROC curves and ROC AUC scores to evaluate model performance, and generating SHAP values to explain the model predictions. The notebook utilizes libraries such as XGBoost for modeling, SHAP for interpreting model predictions, and tools for handling data like pandas and NumPy.

The main findings likely revolve around identifying the most important features influencing the model predictions based on SHAP values. This analysis can help in feature selection, model interpretability, and gaining insights into the relationships between input features and the target variable. Additionally, the ROC curve analysis provides insights into the model's classification performance.

Notebook Analysis: 2_cluster_data_creation.ipynb

Notebook contains 10 cells (2 markdown, 8 code)

Notebook Structure

• 2 Cluster Creation

• 3 Clusters Creation

Notebook Technical Content

Key libraries: roc_curve, numpy, collections, KMedoids, pyreadstat

Visualization methods: plot, plt, scatter, figure

Analysis techniques: train_test_split, KMeans, cluster

Data operations: pd.read_, DataFrame

Notebook Summary:

This Jupyter notebook appears to focus on clustering analysis, particularly on creating clusters from a dataset. The notebook takes a dataset, performs some data preprocessing steps (such as handling special characters in column names and creating dummy variables), and uses the KMedoids clustering algorithm to create clusters. The notebook may explore different cluster creation scenarios by possibly varying the number of clusters.

Key analyses and visualizations may include: • Preprocessing the dataset by handling special characters in column names and creating dummy variables. • Using the KMedoids algorithm to create clusters and visualizing the clustering results. • Analyzing the characteristics of each cluster, potentially using cluster centers or medoids.

The main findings or conclusions of the notebook are not explicitly mentioned. However, the notebook seems focused on applying clustering techniques to gain insights into patterns within the dataset and to potentially segment the data into distinct groups based on similarities.

Notebook Analysis: 3_Clusters_Complete_Analysis.ipynb

Notebook contains 30 cells (10 markdown, 20 code)

Notebook Structure

• Analysis using 3 Clusters

• Analysis using 2 clusters

• Analysis for the 3 clusters without the dummies

• Analysis for the 2 clusters without dummies

• Anova

• T-Test

• table

• table

• graphs

• graphs

Notebook Technical Content

Key libraries: numpy, f_oneway, KMedoids, pandas, matplotlib

Custom functions: plot_density_for_significant_vars, es_dummy

Visualization methods: plt, sns, figure

Analysis techniques: KMeans, cluster

Data operations: pd.read_, DataFrame

Notebook Summary:

Summary:

• Main Purpose/Topic: The notebook focuses on conducting clustering analysis using K-Medoids algorithm and performing ANOVA (Analysis of Variance) to compare means across clusters. It delves into analyzing data with different numbers of clusters to uncover patterns and groupings within the dataset.

• Key Analyses or Visualizations: The notebook includes processes for clustering the data into 2 and 3 clusters, both with and without dummy variable usage. An important analysis involves applying ANOVA to assess the statistical significance of differences in variable means among the clusters. The notebook also creates visualizations like tables and potentially other plots to display the results.

• Main Findings/Conclusions: The findings may include identifying significant variables that differentiate the clusters based on their means and exploring how these variables contribute to distinguishing the clusters. The analyses aim to uncover meaningful insights regarding the characteristics or behaviors of different groups within the dataset, aiding in segmentation or profiling of the data based on common attributes or features. The ANOVA results can offer statistical validation for the differences observed among the clusters.

Notebook Analysis: 4_program_characteristics.ipynb

Notebook contains 24 cells (3 markdown, 21 code)

Notebook Structure

• Analysis using only the program characteristics

• Top 10 features

• Top 20 features

Notebook Technical Content

Key libraries: roc_curve, numpy, collections, pyreadstat, pandas

Custom functions: add_unique_keys, get_remove_features, remove_features, model_loop, remove_none_from_list

Visualization methods: plot, plt, figure

Analysis techniques: train_test_split, XGBoost

Data operations: pd.read_

Notebook Summary:

This Jupyter notebook focuses on feature importance analysis based on program characteristics. The notebook likely aims to identify the most important features that impact a specific outcome or prediction task using machine learning techniques such as XGBoost.

Key Analyses/Visualizations: • Data reading and preprocessing using pandas and pyreadstat to load and prepare the dataset. • Feature engineering with one-hot encoding using pd.get_dummies to convert categorical variables into numerical format. • Training an XGBoost model to assess feature importance. • Using SHAP (SHapley Additive exPlanations) values to explain the output of the XGBoost model.

Main Findings/Conclusions: The notebook likely presents the top 10 and top 20 most important features derived from the XGBoost model. These features could help to understand the key factors influencing the target variable or prediction task. By visualizing the feature importance, data scientists can prioritize variables for further analysis or model improvement, leading to more accurate predictions or insights.

File: Cluster analysis.do

Type: .do, Size: 7.47 KB

Notebook Analysis: Update_labels.ipynb

Notebook contains 4 cells (0 markdown, 4 code)

Notebook Technical Content

Key libraries: pandas, pyreadstat

Notebook Summary:

The main purpose of this Jupyter notebook is data analysis, focusing on reading a Stata (.dta) file and exploring the dataset using Python libraries such as pandas and pyreadstat.

Key analyses and steps in the notebook include: • Reading a Stata file using `pyreadstat` and converting it into a pandas DataFrame. • Extracting metadata information like column labels from the Stata file. • Mapping original column labels to more interpretable labels for better data understanding.

The main findings or outcomes of this notebook may include: • Transforming Stata column labels to more user-friendly labels for easier interpretation of the dataset. • Further data analysis or processing steps are likely to be performed on the modified dataset for insights or modeling purposes.

Overall, the notebook aims to prepare and enhance the dataset for subsequent data analysis or machine learning tasks by converting Stata metadata into more meaningful labels using Python libraries.

File: dimension_reduction.py

Code Summary:

This Python code performs the following tasks: 1. Reads data from a CSV file and preprocesses it. 2. Utilizes the UMAP (Uniform Manifold Approximation and Projection) algorithm for dimension reduction. 3. Conducts clustering using Label Encoder and visualizes the results. Key functions/classes used: 1. StandardScaler: From scikit-learn for standardizing features. 2. UMAP: Imported from the umap library for dimensionality reduction. 3. LabelEncoder: From scikit-learn for converting categorical labels to numeric values. 4. Pandas functions: DataFrame operations for data manipulation. Important algorithms/techniques used: 1. UMAP: Used for dimensionality reduction to visualize high-dimensional data in 2D. 2. Label Encoding: Converts categorical cluster labels to numerical values. 3. Standardization: Standardizes features to have mean=0 and variance=1 for better model performance. 4. Data Preprocessing: Handling missing values, dropping columns with excessive missing values, and filling remaining missing values with column means for a clean dataset.

Code Sample (first 30 lines):

```
import pandas as pd
import numpy as np
from umap import UMAP
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler, LabelEncoder

# Read the Stata exported data
data = pd.read_csv("Results/temp_data_for_umap.csv")

# Separate features for UMAP
features = [col for col in data.columns if col.startswith('p_')]
X = data[features].copy()

# Function to convert range strings to numeric values
def convert_range_to_numeric(value):
    if pd.isna(value):
        return np.nan
    if isinstance(value, (int, float)):
        return float(value)

    try:
        if ' - ' in str(value):
            lower, upper = map(float, str(value).split(' - '))
            return (lower + upper) / 2
```

```
        return float(value)
    except:
        return np.nan

# Convert each column to numeric
for column in X.columns:
```

(... 75 more lines not shown ...)

Data Analysis

Directory contains 3 files and 0 subdirectories

Files:

V1_qualflags_analysis2.dta, V1_qualflags_analysis2_ML.dta, V1_qualflags_analysis2_clustered.dta

File: V1_qualflags_analysis2.dta

Type: .dta, Size: 8867.81 KB

File: V1_qualflags_analysis2_ML.dta

Type: .dta, Size: 2430.68 KB

File: V1_qualflags_analysis2_clustered.dta

Type: .dta, Size: 8741.37 KB

Results Analysis

Directory contains 0 files and 3 subdirectories

Subdirectories:

Organized_Results, Results_Clusters, Results_Feature-Importance

Subdirectory: Organized_Results

Directory contains 4 files and 1 subdirectories

Files:

Cluster_Analysis_Report.docx, Cluster_Analysis_Results.xlsx, Feature_Importance_Report.docx, Feature_Importance_Results.xlsx

Subdirectories:

Images

Word Document: Cluster_Analysis_Report.docx

Document contains 63 paragraphs and 21 headings

Document structure:

- Introduction

- Executive Summary

- Dataset Overview

- Program Distribution by Cluster

- Summary Dashboard

- Differences Between Clusters

- Cluster Characteristics

- Cluster Visualization

- Upskilling vs. Reskilling Programs

- Program Type Characteristics

(... 11 more headings not shown ...)

Document introduction:

Analysis of Upskilling and Reskilling Programs

Analysis Date: March 2025

Introduction

This report presents a comprehensive analysis of upskilling and reskilling programs using cluster analysis. The analysis examines the key differences between these program types and identifies natural groupings based on program characteristics. This information can help organizations design more effective training initiatives and understand how different program structures align with upskilling versus reskilling objectives.

Executive Summary

(... 58 more paragraphs not shown ...)

Document Summary:

The document "Cluster_Analysis_Report.docx" appears to contain a detailed analysis of upskilling and reskilling programs using cluster analysis. It provides insights into the differences between these program types, identifies natural groupings based on program characteristics, and outlines key findings related to program structures and objectives. The report includes information on program distribution by clusters, highlighting differences in participation numbers, funding sources, and target audiences between the two clusters. This analysis is significant for organizations looking to design more effective training initiatives tailored to upskilling and reskilling needs, as it offers a deeper understanding of how these programs are structured and the factors influencing their success in meeting specific objectives.

Excel File: Cluster_Analysis_Results.xlsx

Contains 3 sheets: Summary, cluster_results_2, cluster_results_3

Sheet: Summary

Preview of first 3 rows and 3 columns:

Sheet Name	Description	Source File
cluster_results_2		cluster_results_2
cluster_results_3		cluster_results_3

Sheet: cluster_results_2

Sheet is too large to display: 1126 rows x 186 columns

Column headers:

Year Start Clone2, Year End Clone2, Ongoing, Fund Gov, Fund Org, Fund Wrk, Fund Union, Fund Other, Criteria Jobtitle, Criteria Tenure, Criteria Qualifications, Criteria Assmskills, Criteria Assmsmotivation, Criteria Managerrec, Criteria Other

(... 171 more columns not shown ...)

Sheet: cluster_results_3

Sheet is too large to display: 1126 rows x 186 columns

Column headers:

Year Start Clone2, Year End Clone2, Ongoing, Fund Gov, Fund Org, Fund Wrk, Fund Union, Fund Other, Criteria Jobtitle, Criteria Tenure, Criteria Qualifications, Criteria Assmskills, Criteria Assmsmotivation, Criteria Managerrec, Criteria Other

(... 171 more columns not shown ...)

Possible content interpretation:

From the provided context, the Excel file "Cluster_Analysis_Results.xlsx" likely contains results from a clustering analysis. The Summary sheet may provide an overview of the analysis, while the cluster_results_2 and cluster_results_3 sheets likely contain detailed results for different numbers of clusters (2 and 3 clusters in this case). This file might be essential in understanding how the data points were grouped into clusters and what characteristics define each cluster. It could help in identifying patterns or trends within the data that are not obvious initially, enabling further exploration or targeted actions based on the cluster insights. Hypothesis: The clustering analysis results in this file may reveal distinct customer segments based on purchasing behavior in a retail setting. The analysis could show how customers are grouped into different clusters based on their buying patterns, with cluster_results_2 and cluster_results_3 sheets offering insights into differences between two and three identified customer segments, respectively. These insights can help tailor marketing strategies, product offerings, or customer targeting to each segment's specific preferences and behaviors.

Word Document: Feature_Importance_Report.docx

Document contains 58 paragraphs and 15 headings

Document structure:

- Introduction

- Executive Summary

- Model Performance

- ROC Curve Analysis

- Confusion Matrices

- Top Distinguishing Features

- Key Feature Interpretation

- Extended Feature Importance

- SHAP Analysis

- SHAP Feature Interactions

(... 5 more headings not shown ...)

Document introduction:

Feature Importance Analysis: Upskilling vs. Reskilling Programs

Analysis Date: March 2025

Introduction

This report presents an analysis of the key features that distinguish between upskilling and reskilling programs. Using machine learning techniques, we identified the most important characteristics that differentiate these program types, which can help organizations design more effective training initiatives.

Executive Summary

(... 53 more paragraphs not shown ...)

Document Summary:

This document appears to contain a feature importance analysis report on distinguishing upskilling and reskilling programs using machine learning techniques. It discusses the key findings, such as the importance of program length, job placement focus, and management targeting as differentiating features. It also highlights that including outcome variables improves model performance, suggesting a strong link between program outcomes and program type. The significance of this report lies in its insights for organizations to design more effective training initiatives by understanding the key features that differentiate between various training programs. This analysis can help in making informed decisions to tailor training programs to specific needs and enhance overall program effectiveness.

Excel File: Feature_Importance_Results.xlsx

Contains 9 sheets: Summary, Confusion Matrix 1, Confusion Matrix 2, Confusion Matrix 3, Confusion Matrix 4, Misclassifications, Misclassifications_1, Model Performance, Top Features

Sheet: Summary

Preview of first 6 rows and 3 columns:

Sheet Name	Description	Source File
Confusion Matrix 1	Confusion matrix for model 1.	confusion_matrix_model_1
Confusion Matrix 2	Confusion matrix for model 2.	confusion_matrix_model_2
Confusion Matrix 3	Confusion matrix for model 3.	confusion_matrix_model_3
Confusion Matrix 4	Confusion matrix for model 4.	confusion_matrix_model_4
Misclassifications	Analysis of misclassified samples.	misclassification_probabilities

(... 3 more rows not shown ...)

Sheet: Confusion Matrix 1

Preview of first 3 rows and 3 columns:

Unnamed: 0	Predicted 0	Predicted 1
Actual 0	76	31
Actual 1	11	124

Sheet: Confusion Matrix 2

Preview of first 3 rows and 3 columns:

Unnamed: 0	Predicted 0	Predicted 1
Actual 0	46	61
Actual 1	17	118

(... 6 more sheets not shown ...)

Possible content interpretation:

This Excel file likely contains results and analysis related to a machine learning model. The sheets such as Summary, Model Performance, and Top Features suggest that it includes information about feature importance, model performance metrics, and potentially misclassifications. The confusion matrices provide a detailed breakdown of the model's predicted classes compared to the actual classes. Hypothesis: The file likely contains the results of a classification model where feature importance analysis was conducted to understand the factors driving predictions. The confusion matrices and misclassifications sheets might offer insights into how well the model performs and where it struggles in accurately predicting classes. This file could be crucial for evaluating the model's performance, identifying areas of improvement, and making informed decisions for further model optimization.

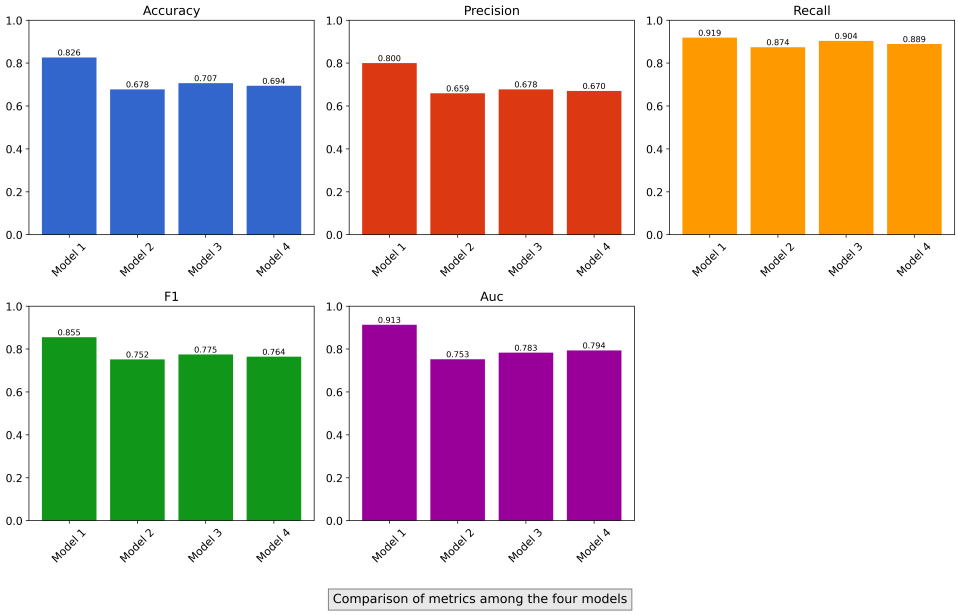
Subdirectory: Images

Directory contains 17 files and 0 subdirectories

Files:

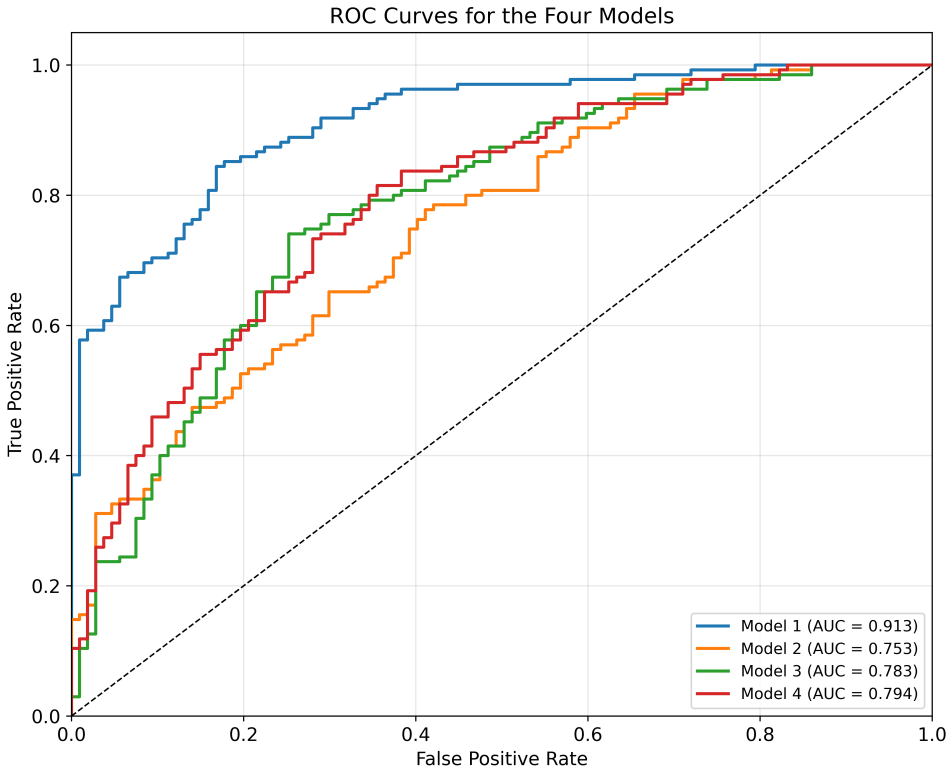
01_Model_Performance_Metrics.png, 02_ROC_Curves.png, 03_Confusion_Matrices.png, 04_Top10_Important_Features.png, 05_Top25_Important_Features.png, 06_Misclassifications_Distribution.png, 07_SHAP_Summary.png, 08_SHAP_Beeswarm.png, 16_UMAP_2Clusters.png, 17_UMAP_3Clusters.png, 18_Cluster_Program_Comparison_k2.png, cluster_program_distribution_2.png, cluster_program_distribution_3.png, clustering_methods_comparison.png, feature_importance_2_clusters.png, feature_importance_3_clusters.png, program_distribution_3_clusters.png

Image: 01_Model_Performance_Metrics.png



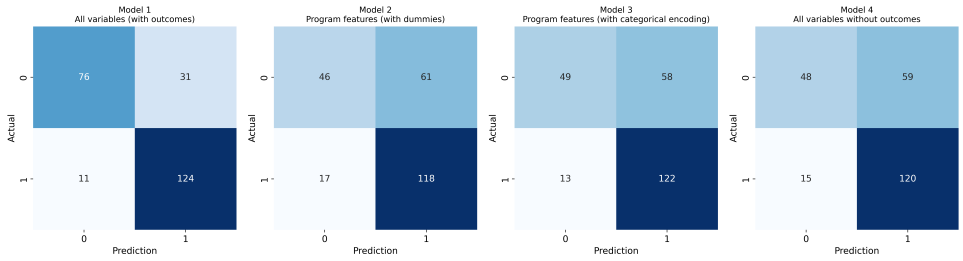
01_Model_Performance_Metrics.png

Image: 02_ROC_Curves.png



02_ROC_Curves.png

Image: 03_Confusion_Matrices.png

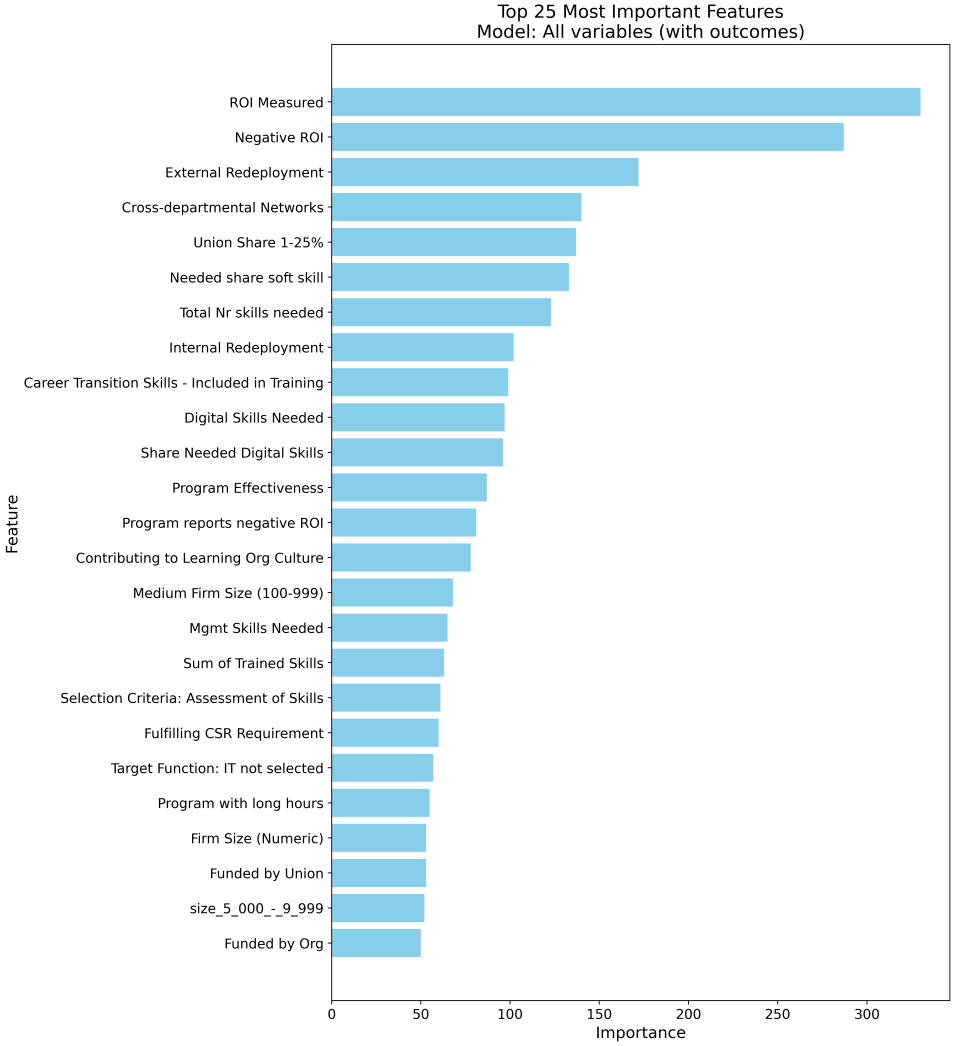


03_Confusion_Matrices.png

Image: 04_Top10_Important_Features.png

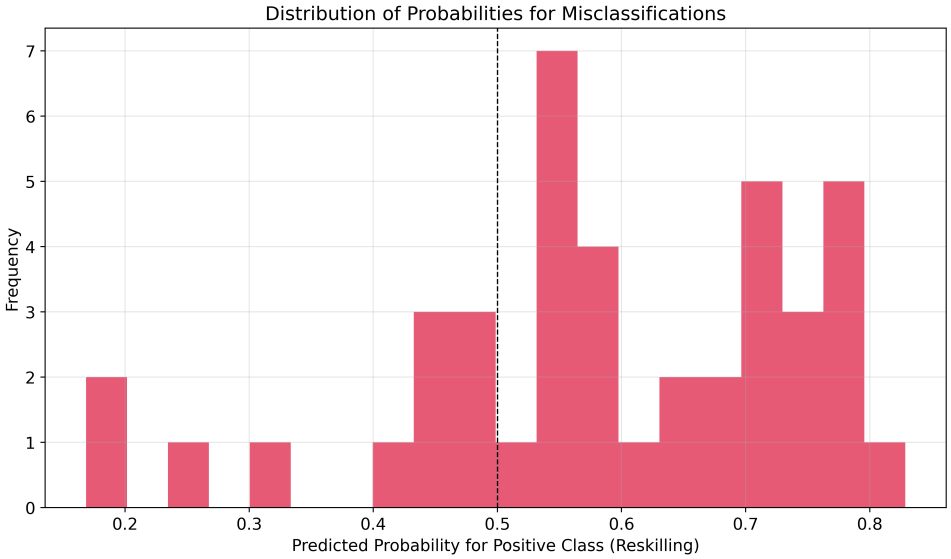
04_Top10_Important_Features.png

Image: 05_Top25_Important_Features.png



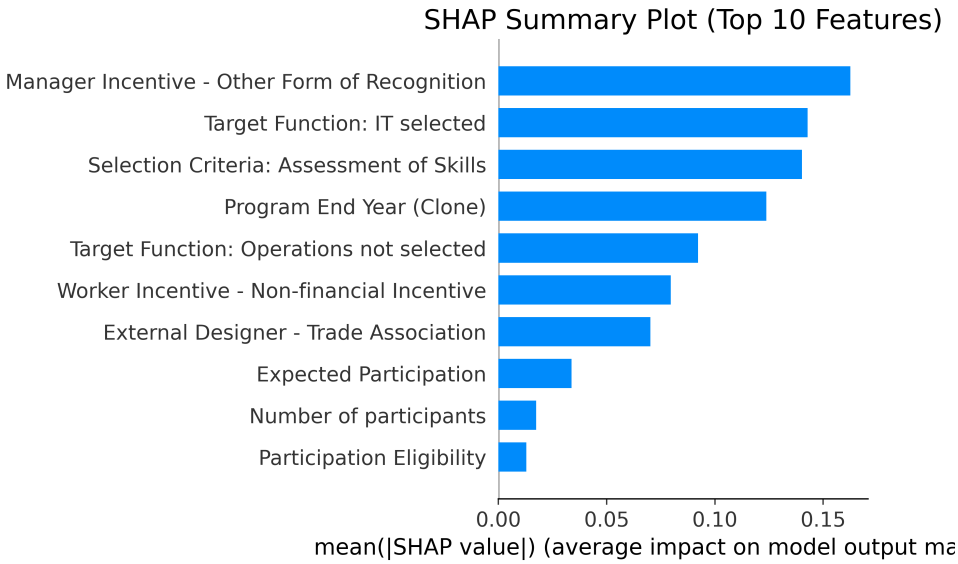
05_Top25_Important_Features.png

Image: 06_Misclassifications_Distribution.png



06_Misclassifications_Distribution.png

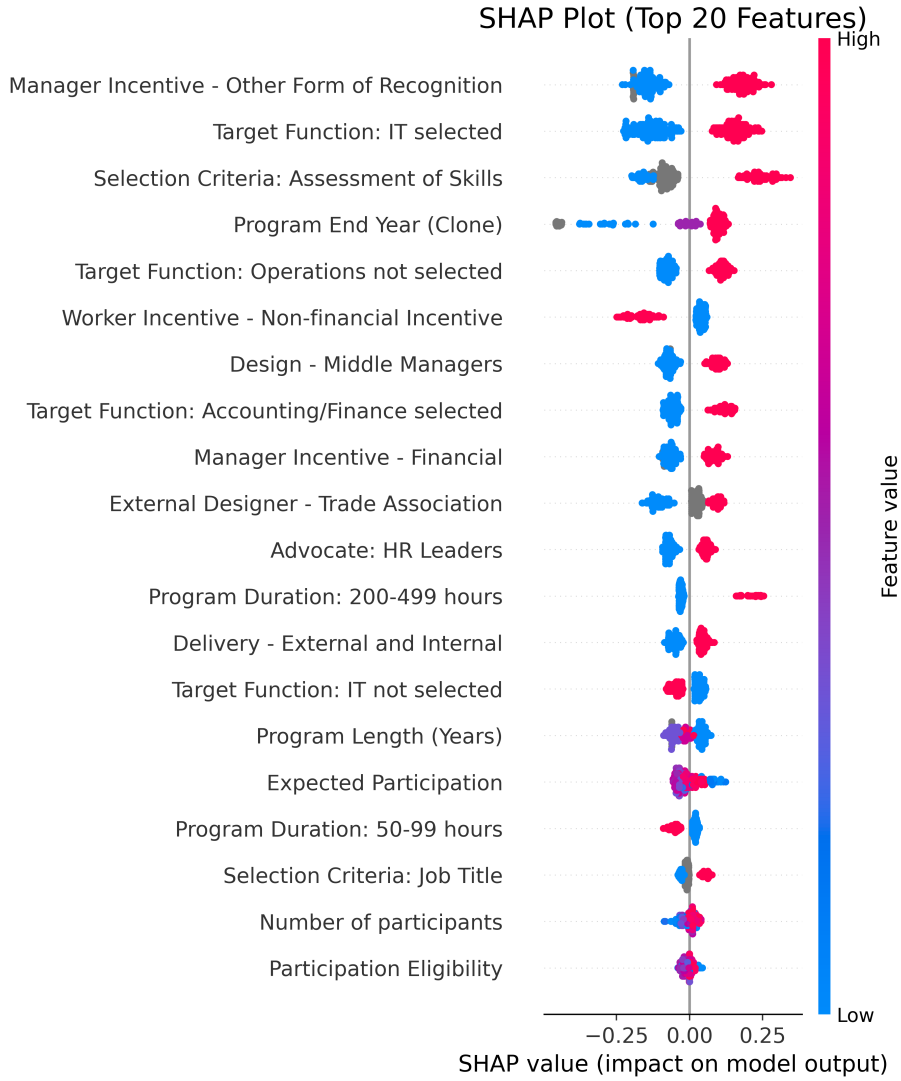
Image: 07_SHAP_Summary.png



Data: Program Characteristics (Without Outcomes), using dummies

07_SHAP_Summary.png

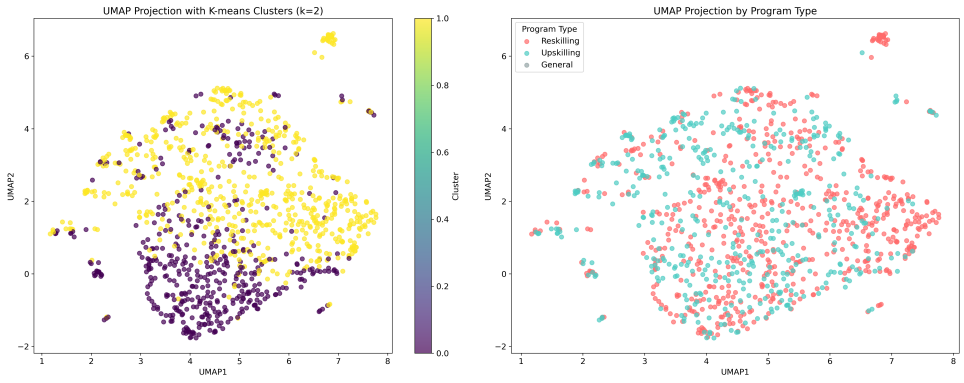
Image: 08_SHAP_Beeswarm.png



Data: Program Characteristics (Without Outcomes), using dummies

08_SHAP_Beeswarm.png

Image: 16_UMAP_2Clusters.png

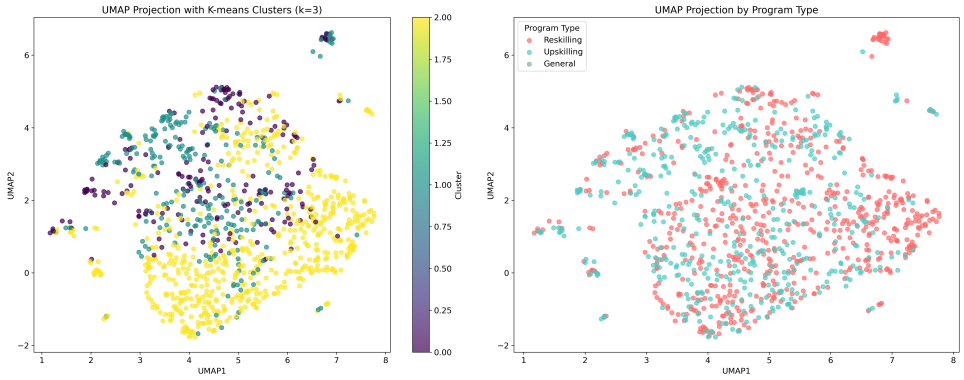


16_UMAP_2Clusters.png

Possible interpretation:

Based on the filename "16_UMAP_2Clusters.png," this visualization likely shows a two-dimensional representation of data points clustered using UMAP (Uniform Manifold Approximation and Projection). UMAP is a dimensionality reduction technique often used for clustering analysis. The plot might display how the data points are grouped into two distinct clusters based on their underlying features.

Image: 17_UMAP_3Clusters.png

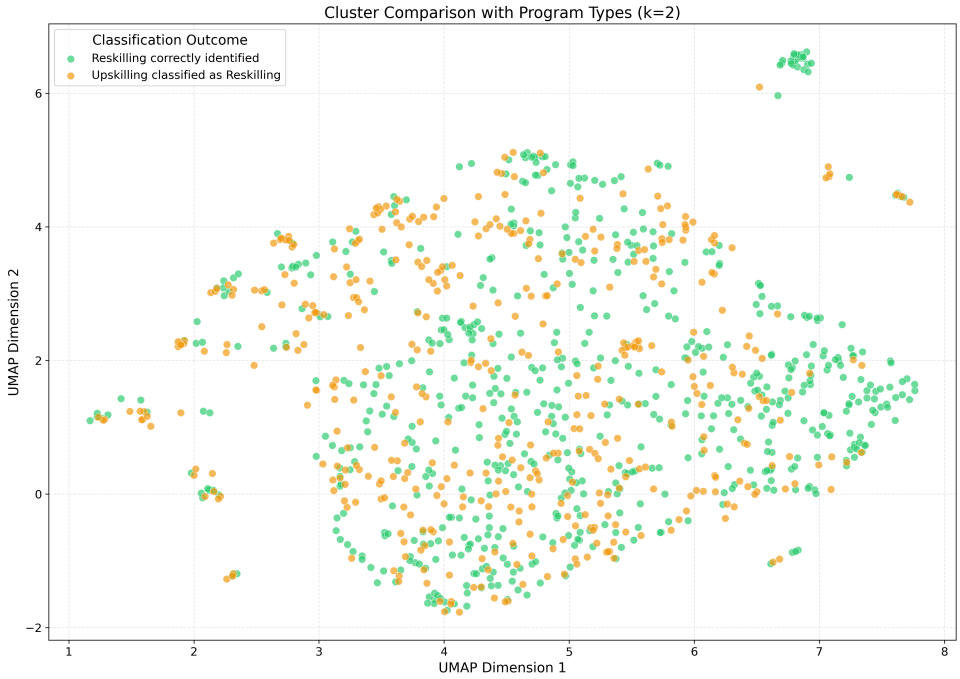


17_UMAP_3Clusters.png

Possible interpretation:

Based on the filename "17_UMAP_3Clusters.png", this visualization is likely showing a 2-dimensional UMAP (Uniform Manifold Approximation and Projection) plot with data points clustered into 3 distinct groups. UMAP is a dimensionality reduction technique commonly used for visualizing high-dimensional data in a lower-dimensional space while preserving the underlying structure and relationships between data points. The presence of 3 clusters suggests that the data may have been clustered or classified into three different groups based on certain features or characteristics.

Image: 18_Cluster_Program_Comparison_k2.png

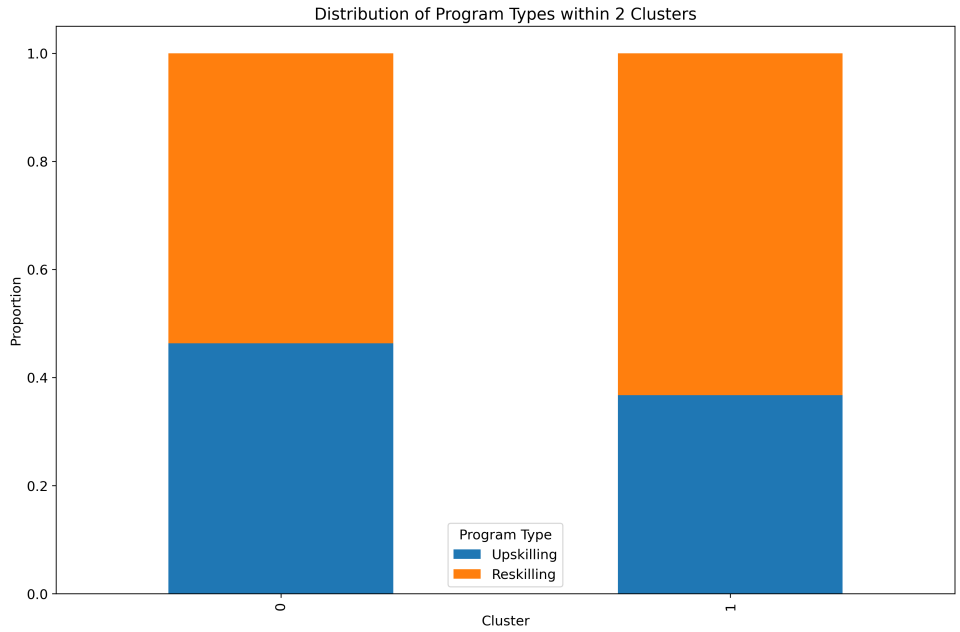


18_Cluster_Program_Comparison_k2.png

Possible interpretation:

Based on the filename "18_Cluster_Program_Comparison_k2.png", this visualization could be showing a comparison of clusters or groups generated by a clustering algorithm for a specific dataset. The "k2" in the filename suggests that the clustering algorithm used two clusters to group the data points, and the plot may be comparing how different programs or methodologies perform in clustering the data.

Image: cluster_program_distribution_2.png

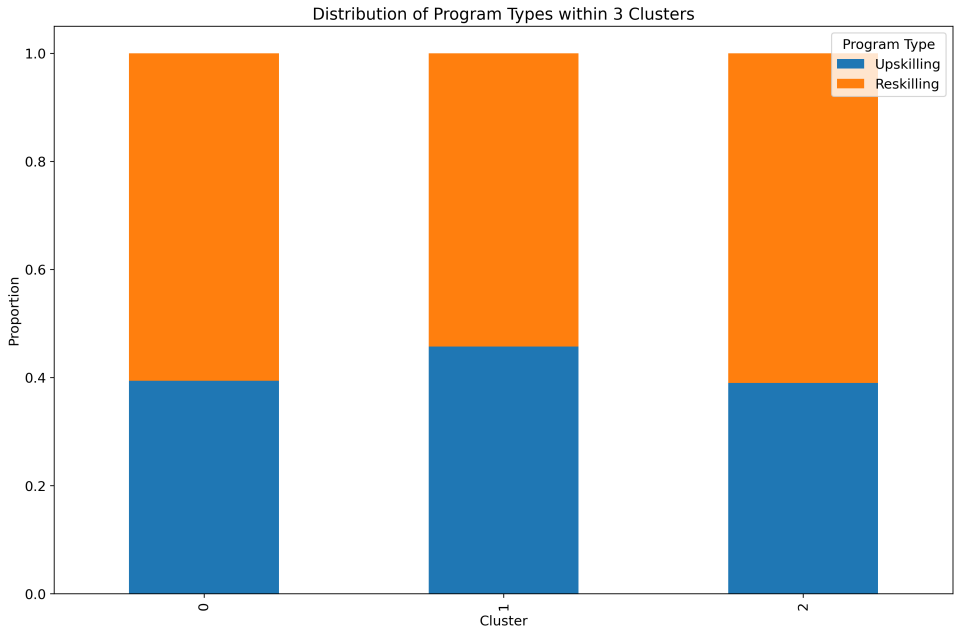


cluster_program_distribution_2.png

Possible interpretation:

From the filename "cluster_program_distribution_2.png," it suggests that this visualization is likely showing the distribution of programs or categories within different clusters. The plot may illustrate how different programs are distributed across clusters or how prevalent certain programs are within each cluster, providing insights into grouping patterns based on program features.

Image: cluster_program_distribution_3.png

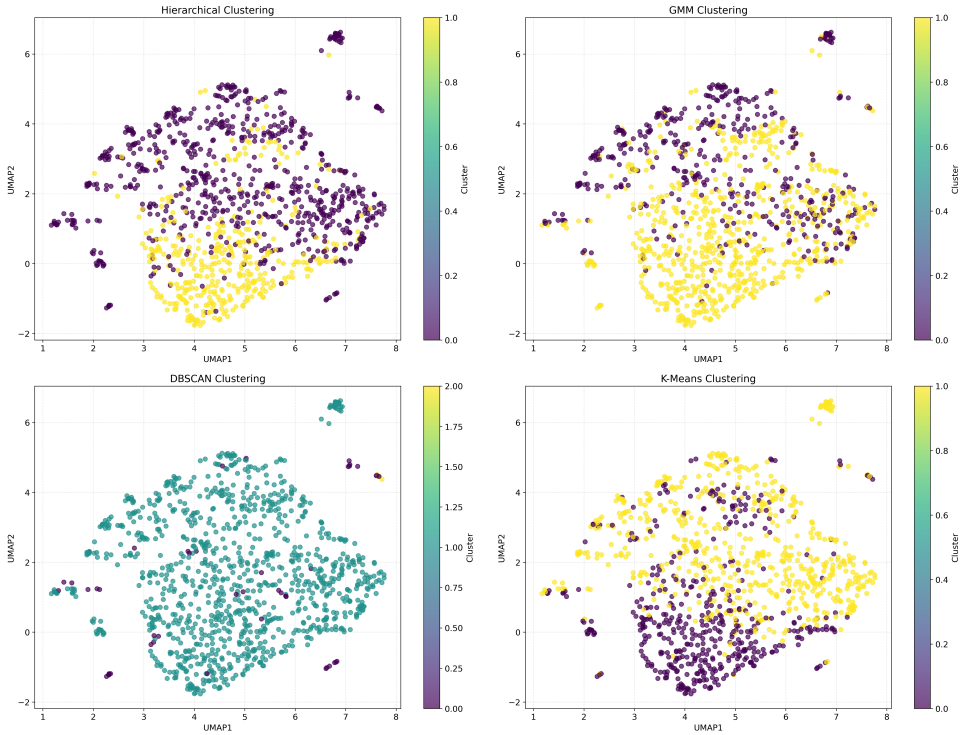


cluster_program_distribution_3.png

Possible interpretation:

Based on the filename "cluster_program_distribution_3.png," this visualization is likely showing the distribution of data points within different clusters or groups based on a specific program or category. Each cluster likely represents a subgroup or classification within the data, and the plot may provide insights into how these clusters are distributed in relation to the program or category being analyzed.

Image: clustering_methods_comparison.png

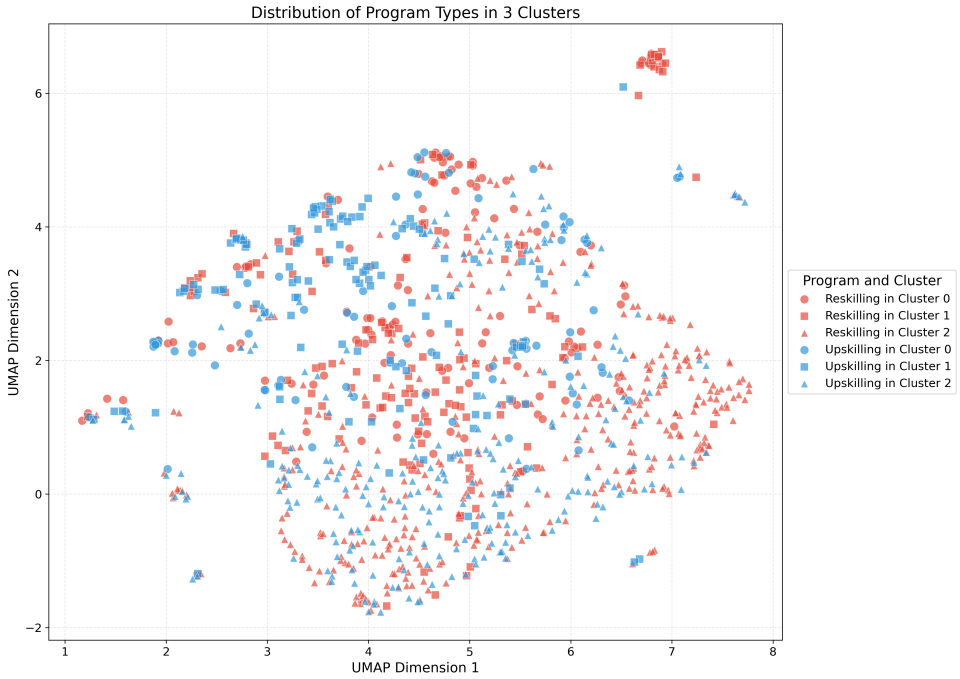


clustering_methods_comparison.png

Possible interpretation:

Based on the filename "clustering_methods_comparison.png", this visualization is likely showing a comparison of different clustering methods. The plot may display how different clustering algorithms perform in clustering a given dataset, potentially showing metrics like silhouette score, clustering accuracy, or cluster visualization quality for each method.

Image: program_distribution_3_clusters.png



program_distribution_3_clusters.png

Possible interpretation:

Based on the filename "program_distribution_3_clusters.png," this visualization likely shows the distribution of data points or elements grouped into three clusters based on a certain program or category. The plot may display how the data points are clustered or categorized within these specific groups, providing insights into patterns or relationships within the data.

Subdirectory: Results_Clusters

Directory contains 0 files and 3 subdirectories

Subdirectories:

Figures, Reports, Statistics

Subdirectory: Figures

Directory contains 6 files and 2 subdirectories

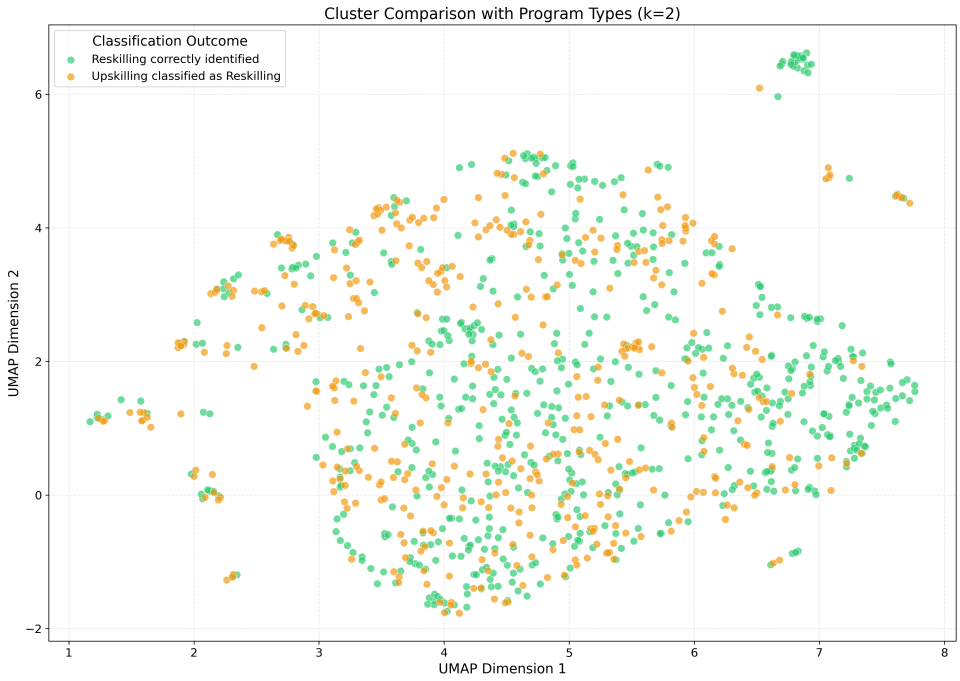
Files:

cluster_program_comparison_2.png, cluster_program_distribution_2.png, cluster_program_distribution_3.png, clustering_methods_comparison.png, umap_clusters_2.png, umap_clusters_3.png

Subdirectories:

k2_analysis, k3_analysis

Image: cluster_program_comparison_2.png

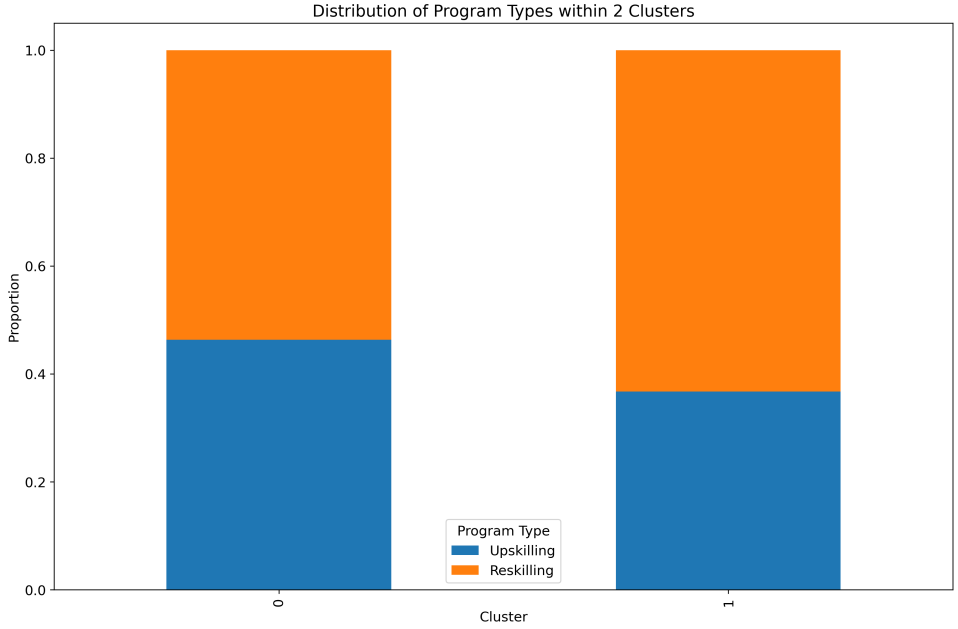


cluster_program_comparison_2.png

Possible interpretation:

Based on the filename "cluster_program_comparison_2.png," this visualization likely compares different clusters or groups based on some program variables. The plot may display how different programs perform within each cluster or how clusters differ in terms of program metrics. The figure may offer insights into the effectiveness of various programs or the distinct characteristics of each cluster.

Image: cluster_program_distribution_2.png

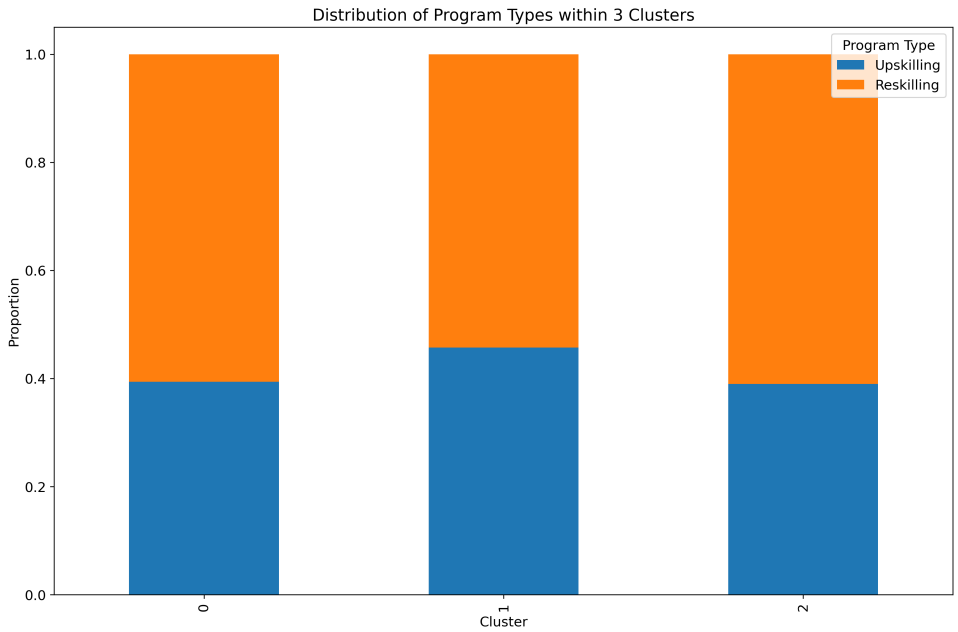


cluster_program_distribution_2.png

Possible interpretation:

Based on the filename "cluster_program_distribution_2.png", this visualization likely shows the distribution of programs across different clusters in a data science project. Each cluster may represent a group of similar programs or projects, and the plot may illustrate how the programs are distributed among these clusters. This information can help in understanding patterns or relationships between different types of programs within the project.

Image: cluster_program_distribution_3.png

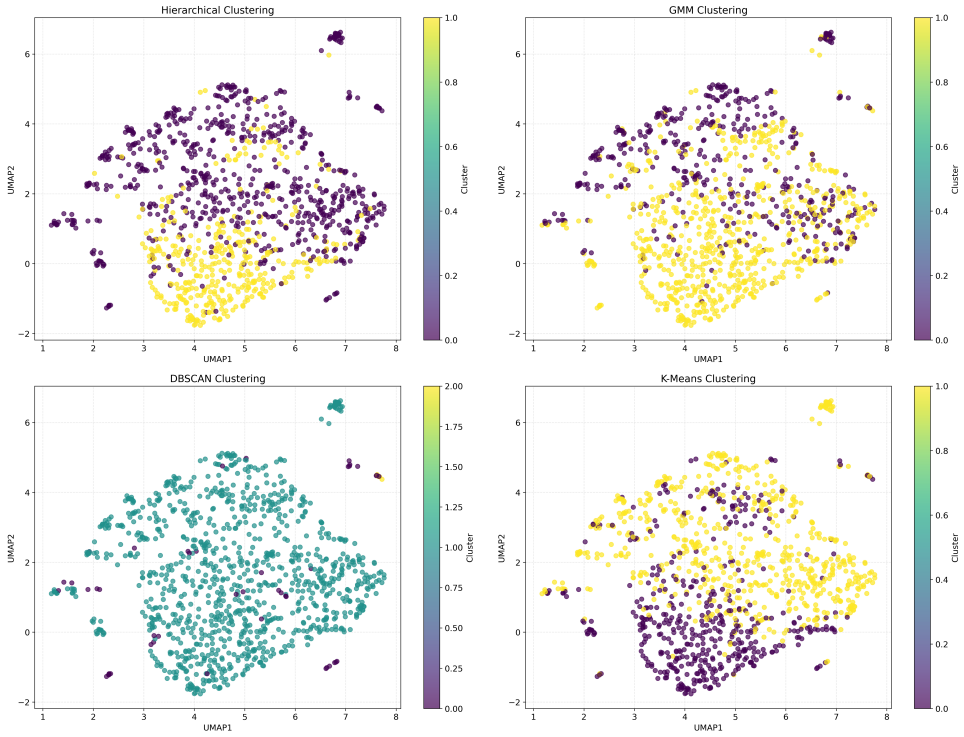


cluster_program_distribution_3.png

Possible interpretation:

Based on the filename "cluster_program_distribution_3.png," this visualization is likely showing the distribution of data points or observations across different clusters or groups in a program or dataset. The plot may illustrate how the data is grouped or clustered based on specific variables or features, providing insights into patterns or relationships within the data.

Image: clustering_methods_comparison.png

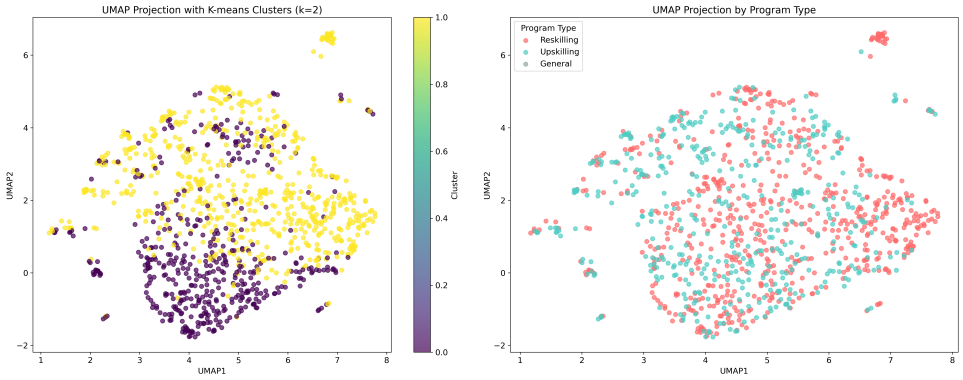


clustering_methods_comparison.png

Possible interpretation:

Based on the filename "clustering_methods_comparison.png," this visualization could be showing a comparison of different clustering methods. It likely compares the performance or outcome of various clustering algorithms on a dataset, showcasing how they group data points into clusters. The plot may display metrics such as cluster quality or separation to evaluate the effectiveness of each method.

Image: umap_clusters_2.png

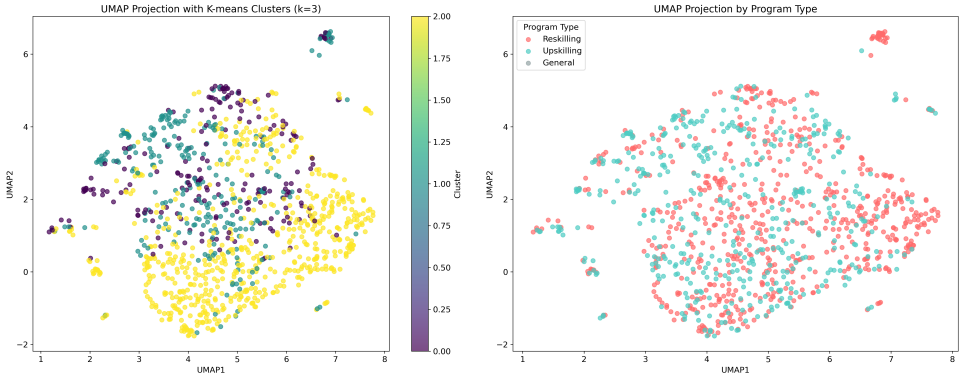


umap_clusters_2.png

Possible interpretation:

Based on the filename "umap_clusters_2.png," this visualization is likely showing a 2-dimensional representation of data points that have been clustered or grouped together using a dimensionality reduction technique called UMAP (Uniform Manifold Approximation and Projection). The plot may display different clusters or patterns in the data based on their similarities or relationships in a lower-dimensional space.

Image: umap_clusters_3.png



umap_clusters_3.png

Possible interpretation:

Based on the filename "umap_clusters_3.png," this visualization is likely showing a clustering analysis using the UMAP (Uniform Manifold Approximation and Projection) algorithm. The plot may display data points in a reduced-dimensional space where similar points are grouped together into clusters. The number "3" in the filename suggests that there are three distinct clusters being visualized.

Subdirectory: k2_analysis

Directory contains 1 files and 0 subdirectories

Files:

feature_importance_2_clusters.png

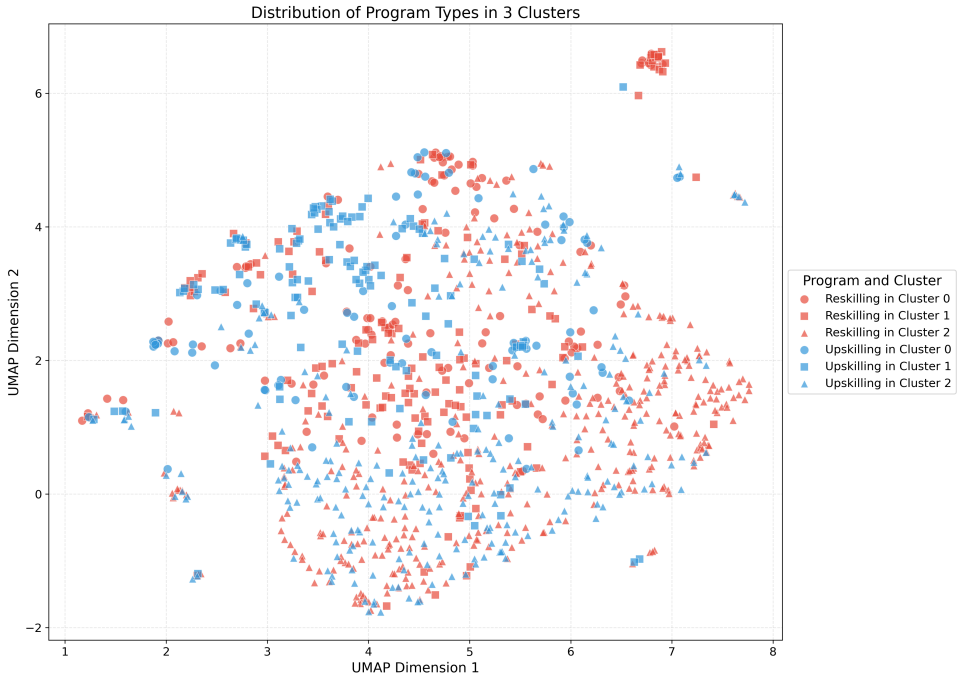
Subdirectory: k3_analysis

Directory contains 2 files and 0 subdirectories

Files:

feature_importance_3_clusters.png, program_distribution_3_clusters.png

Image: program_distribution_3_clusters.png



program_distribution_3_clusters.png

Possible interpretation:

Based on the filename "program_distribution_3_clusters.png", this visualization might be showing the distribution of data points clustered into three distinct groups or clusters related to some programs or categories. Each cluster is likely represented using different colors or markers to distinguish them from each other. The plot could help in identifying patterns or relationships between the clusters, providing insights into the underlying structure of the data.

Subdirectory: Reports

Subdirectory: Statistics

Directory contains 2 files and 2 subdirectories

Files:

cluster_results_2.csv, cluster_results_3.csv

Subdirectories:

k2_analysis, k3_analysis

CSV File: cluster_results_2.csv

Contains 1125 rows and 186 columns

Columns:

p_year_start_clone2, p_year_end_clone2, p_ongoing, p_fund_gov, p_fund_org, p_fund_wrk, p_fund_union, p_fund_other, p_criteria_jobtitle, p_criteria_tenure, p_criteria_qualifications, p_criteria_assmskills, p_criteria_assmsmotivation, p_criteria_managerrec, p_criteria_other, p_challenge_convincingemployees, p_challenge_selecting, p_challenge_progcompl, p_challenge_learning, p_challenge_newjob, p_challenge_convmanager, p_challenge_effectiveness, p_challenge_scaling, p_challenge_determreturn, p_challenge_supportoutsideHR, p_challenge_funding, p_challenge_other, p_eligibility, p_part, p_part_exp, p_effect_reverse, p_participated_coarse, p_size_coarse1, p_size_coarse2, p_size_coarse3, p_target_top, p_target_middle, p_target_emp, p_target_topmiddle, p_target_middleemp, p_target_all, p_target_topbottom, p_long, p_hours_long, p_compfull, p_oja, p_adv_top, p_adv_hr, p_resp_top, p_resp_hr, p_adv_resp_match, p_cri_individual, p_cha_takeup, p_cha_during, p_cha_support, p_cha_scale, p_participated_Fewer than 50, p_participated_50 - 99, p_participated_100 - 499, p_participated_500 - 999, p_participated_1000 - 9999, p_participated_10000 - 49999, p_participated_50000 - 99999, p_participated_100000 or more, p_participated_2023_Fewer than 50, p_participated_2023_50 - 99, p_participated_2023_100 - 499, p_participated_2023_500 - 999, p_participated_2023_1000 - 9999, p_participated_2023_10000 - 49999, p_participated_2023_50000 - 99999, p_participated_2023_100000 or more, p_mandavolunt_Mandatory, p_mandavolunt_Voluntary, p_year_start_2019, p_year_start_2020, p_year_start_2021, p_year_start_2022, p_year_start_2023, p_year_end_2019, p_year_end_2020, p_year_end_2021, p_year_end_2022, p_year_end_2023, p_year_end_Ongoing, p_hourstrained_less 10 hours, p_hourstrained_10 - 49 hours, p_hourstrained_50 - 99 hours, p_hourstrained_100 - 199 hours, p_hourstrained_200 - 499 hours, p_hourstrained_greater 500 hours, p_duration_less 1 month, p_duration_2 - 6 months, p_duration_7 - 12 months, p_duration_ greater 12 months, p_comphours_Entirely compensated hours, p_comphours_Partly compensated hours, p_comphours_Uncompensated hours, p_otjactivities_Yes, p_otjactivities_No, p_cost_less $500, p_cost_$501 - $1_000, p_cost_$1_001 - $5_000, p_cost_$5_001 - $10_000, p_cost_greater $10_000, p_adequatefund_Overfunded, p_adequatefund_Adequate funding, p_adequatefund_Underfunded, p_adequatefunddummy_Not adequately funded, p_adequatefunddummy_Adequately funded, p_advocacy_Board of Directors, p_advocacy_C-Suite Leaders excluding HR, p_advocacy_HR Leaders, p_advocacy_Business Unit/Subsidiary/Function Leaders, p_advocacy_Middle Managers and/or front-line supervisors, p_advocacy_Employees (salaried and/or hourly wage), p_advocacy_Unclear who is the primary advocate, p_advocacy_hier_Top Management, p_advocacy_hier_HR, p_advocacy_hier_BU and Middle Management, p_advocacy_hier_Employees and Others, p_responsibility_Board of Directors, p_responsibility_C-Suite Leaders_ excluding HR, p_responsibility_HR Leaders, p_responsibility_Business

Unit/Subsidiary/Function Leaders, p_responsibility_Middle Managers and/or front-line supervisors, p_responsibility_Employees (salaried and/or hourly wage), p_responsibility_Internal Academy, p_responsibility_hier_Top Management, p_responsibility_hier_HR, p_responsibility_hier_BU and Middle Management, p_responsibility_hier_Employees and Others, p_application_Anyone could apply regardless of their function/department , p_application_Anyone could apply if belonging to specific function/department/hier level, p_application_Only people nominated by managers, p_application_Other, p_selection_Yes, p_selection_No, p_targetemp_c_Not Selected, p_targetemp_c_Selected, p_targetemp_bul_Not Selected, p_targetemp_bul_Selected, p_targetemp_mm_Not Selected, p_targetemp_mm_Selected, p_targetemp_emp_Not Selected, p_targetemp_emp_Selected, p_targetfunc_leg_Not Selected, p_targetfunc_leg_Selected, p_targetfunc_hr_Not Selected, p_targetfunc_hr_Selected, p_targetfunc_adm_Not Selected, p_targetfunc_adm_Selected, p_targetfunc_it_Not Selected, p_targetfunc_it_Selected, p_targetfunc_op_Not Selected, p_targetfunc_op_Selected, p_targetfunc_mrksal_Not Selected, p_targetfunc_mrksal_Selected, p_targetfunc_rd_Not Selected, p_targetfunc_rd_Selected, p_targetfunc_accfin_Not Selected, p_targetfunc_accfin_Selected, p_targetfunc_cust_Not Selected, p_targetfunc_cust_Selected, p_difloc_Yes, p_difloc_No, p_difstand_Very standard across geographies, p_difstand_Mix of standard and custom, p_difstand_Very flexible, p_cont_investment_Not at all likely, p_cont_investment_Somewhat likely, p_cont_investment_Uncertain, p_cont_investment_Likely, p_cont_investment_Very likely, p_finassessment_Yes, p_finassessment_No, p_effectiveness_Very Effective, p_effectiveness_Moderately Effective, p_effectiveness_Not Effective, p_roi_No attempt to calculate, p_roi_Not yet_ but intends to, p_roi_Tried to_ but unable, p_roi_Negative ROI, p_roi_Positive ROI, KMeans_Cluster, Program_Type

Data preview (first 10 rows):

quatefunddummy_Adequately funded	p_advocacy_Board of Directors	p_advocacy_C-Suite Leaders excluding HR	p_ad
	False	False	True
	False	False	False
	False	False	True
	False	False	True
	False	False	False
	False	False	True
	False	False	False
	False	False	True
	False	False	False
	False	False	False

Possible data interpretation:

The CSV file "cluster_results_2.csv" likely contains data related to programs or initiatives within an organization, with various attributes such as funding sources, program criteria, challenges faced, target audience, effectiveness assessment, and clustering information. Insights from this data could provide an understanding of program characteristics, effectiveness, and factors influencing program success. It could help in identifying patterns within different program types based on the clustering information provided. Hypothesis: By analyzing the program criteria, funding sources, and challenges faced across different clusters, we can hypothesize that programs requiring more diverse funding sources and having specific eligibility criteria may be more effective in achieving their intended outcomes compared to programs facing

challenges related to convincing employees or scaling up.

CSV File: cluster_results_3.csv

Contains 1125 rows and 186 columns

Columns:

p_year_start_clone2, p_year_end_clone2, p_ongoing, p_fund_gov, p_fund_org, p_fund_wrk, p_fund_union, p_fund_other, p_criteria_jobtitle, p_criteria_tenure, p_criteria_qualifications, p_criteria_assmskills, p_criteria_assmsmotivation, p_criteria_managerrec, p_criteria_other, p_challenge_convincingemployees, p_challenge_selecting, p_challenge_progcompl, p_challenge_learning, p_challenge_newjob, p_challenge_convmanager, p_challenge_effectiveness, p_challenge_scaling, p_challenge_determreturn, p_challenge_supportoutsideHR, p_challenge_funding, p_challenge_other, p_eligibility, p_part, p_part_exp, p_effect_reverse, p_participated_coarse, p_size_coarse1, p_size_coarse2, p_size_coarse3, p_target_top, p_target_middle, p_target_emp, p_target_topmiddle, p_target_middleemp, p_target_all, p_target_topbottom, p_long, p_hours_long, p_compfull, p_oja, p_adv_top, p_adv_hr, p_resp_top, p_resp_hr, p_adv_resp_match, p_cri_individual, p_cha_takeup, p_cha_during, p_cha_support, p_cha_scale, p_participated_Fewer than 50, p_participated_50 - 99, p_participated_100 - 499, p_participated_500 - 999, p_participated_1000 - 9999, p_participated_10000 - 49999, p_participated_50000 - 99999, p_participated_100000 or more, p_participated_2023_Fewer than 50, p_participated_2023_50 - 99, p_participated_2023_100 - 499, p_participated_2023_500 - 999, p_participated_2023_1000 - 9999, p_participated_2023_10000 - 49999, p_participated_2023_50000 - 99999, p_participated_2023_100000 or more, p_mandavolunt_Mandatory, p_mandavolunt_Voluntary, p_year_start_2019, p_year_start_2020, p_year_start_2021, p_year_start_2022, p_year_start_2023, p_year_end_2019, p_year_end_2020, p_year_end_2021, p_year_end_2022, p_year_end_2023, p_year_end_Ongoing, p_hourstrained_less 10 hours, p_hourstrained_10 - 49 hours, p_hourstrained_50 - 99 hours, p_hourstrained_100 - 199 hours, p_hourstrained_200 - 499 hours, p_hourstrained_greater 500 hours, p_duration_less 1 month, p_duration_2 - 6 months, p_duration_7 - 12 months, p_duration_ greater 12 months, p_comphours_Entirely compensated hours, p_comphours_Partly compensated hours, p_comphours_Uncompensated hours, p_otjactivities_Yes, p_otjactivities_No, p_cost_less $500, p_cost_$501 - $1_000, p_cost_$1_001 - $5_000, p_cost_$5_001 - $10_000, p_cost_greater $10_000, p_adequatefund_Overfunded, p_adequatefund_Adequate funding, p_adequatefund_Underfunded, p_adequatefunddummy_Not adequately funded, p_adequatefunddummy_Adequately funded, p_advocacy_Board of Directors, p_advocacy_C-Suite Leaders excluding HR, p_advocacy_HR Leaders, p_advocacy_Business Unit/Subsidiary/Function Leaders, p_advocacy_Middle Managers and/or front-line supervisors, p_advocacy_Employees (salaried and/or hourly wage), p_advocacy_Unclear who is the primary advocate, p_advocacy_hier_Top Management, p_advocacy_hier_HR, p_advocacy_hier_BU and Middle Management, p_advocacy_hier_Employees and Others, p_responsibility_Board of Directors, p_responsibility_C-Suite Leaders_ excluding HR, p_responsibility_HR Leaders, p_responsibility_Business Unit/Subsidiary/Function Leaders, p_responsibility_Middle Managers and/or front-line supervisors, p_responsibility_Employees (salaried and/or hourly wage), p_responsibility_Internal Academy, p_responsibility_hier_Top Management, p_responsibility_hier_HR, p_responsibility_hier_BU and Middle Management, p_responsibility_hier_Employees and Others, p_application_Anyone could apply regardless of their function/department , p_application_Anyone could apply if belonging to specific function/department/hier level, p_application_Only people nominated by managers, p_application_Other, p_selection_Yes, p_selection_No, p_targetemp_c_Not Selected, p_targetemp_c_Selected, p_targetemp_bul_Not Selected, p_targetemp_bul_Selected, p_targetemp_mm_Not Selected, p_targetemp_mm_Selected, p_targetemp_emp_Not Selected, p_targetemp_emp_Selected,

p_targetfunc_leg_Not Selected, p_targetfunc_leg_Selected, p_targetfunc_hr_Not Selected, p_targetfunc_hr_Selected, p_targetfunc_adm_Not Selected, p_targetfunc_adm_Selected, p_targetfunc_it_Not Selected, p_targetfunc_it_Selected, p_targetfunc_op_Not Selected, p_targetfunc_op_Selected, p_targetfunc_mrksal_Not Selected, p_targetfunc_mrksal_Selected, p_targetfunc_rd_Not Selected, p_targetfunc_rd_Selected, p_targetfunc_accfin_Not Selected, p_targetfunc_accfin_Selected, p_targetfunc_cust_Not Selected, p_targetfunc_cust_Selected, p_difloc_Yes, p_difloc_No, p_difstand_Very standard across geographies, p_difstand_Mix of standard and custom, p_difstand_Very flexible, p_cont_investment_Not at all likely, p_cont_investment_Somewhat likely, p_cont_investment_Uncertain, p_cont_investment_Likely, p_cont_investment_Very likely, p_finassessment_Yes, p_finassessment_No, p_effectiveness_Very Effective, p_effectiveness_Moderately Effective, p_effectiveness_Not Effective, p_roi_No attempt to calculate, p_roi_Not yet_ but intends to, p_roi_Tried to_ but unable, p_roi_Negative ROI, p_roi_Positive ROI, KMeans_Cluster, Program_Type

Data preview (first 10 rows):

quatefunddummy_Adequately funded	p_advocacy_Board of Directors	p_advocacy_C-Suite Leaders excluding HR	p_ad
	False	False	True
	False	False	False
	False	False	True
	False	False	True
	False	False	False
	False	False	True
	False	False	False
	False	False	True
	False	False	False
	False	False	False

Possible data interpretation:

The data in the CSV file "cluster_results_3.csv" likely contains information related to various aspects of a program, including program details, funding sources, criteria for participation, challenges faced, target participants, advocacy and responsibility within the organization, application and selection process, investment and effectiveness assessments, and program outcomes (ROI). It appears that the data involves clustering results and program types. Insights from this data could provide valuable information on program effectiveness, factors influencing program success, resource allocation, participant selection criteria, and advocacy patterns within the organization. Analyzing the cluster results may reveal patterns in program characteristics and performance across different types of programs. Hypothesis: Programs with higher levels of advocacy and involvement from top management and HR leaders are more likely to report positive ROI compared to programs where advocacy responsibilities are unclear or distributed across various levels. Additionally, programs that offer flexible funding sources and involve employees from diverse functional areas may experience higher participation rates and better program effectiveness.

Subdirectory: k2_analysis

Subdirectory: k3_analysis

Subdirectory: Results_Feature-Importance

Directory contains 0 files and 3 subdirectories

Subdirectories:

Figures, Reports, Statistics

Subdirectory: Figures

Directory contains 0 files and 3 subdirectories

Subdirectories:

Feature_Importance, Model_Comparisons, SHAP_Analysis

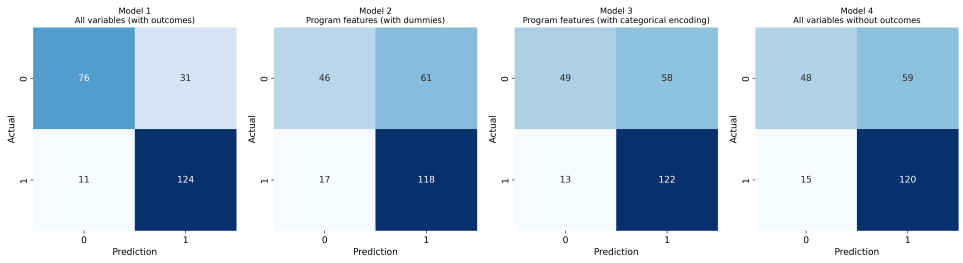
Subdirectory: Feature_Importance

Directory contains 21 files and 0 subdirectories

Files:

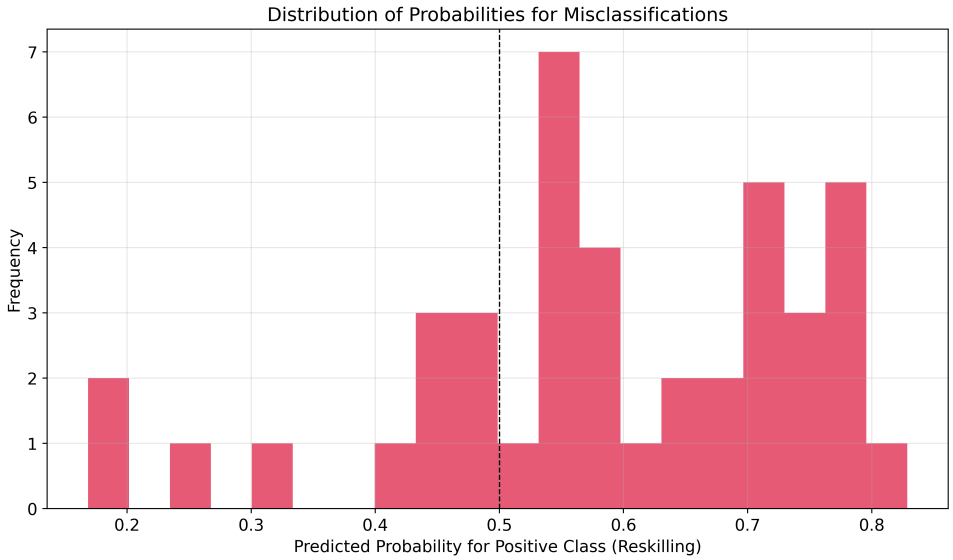
confusion_matrices.png, incorrect_classifications_proba_dist.png, metrics_comparison.png, roc_curves_comparison.png, shap_beeswarm_top10.png, shap_beeswarm_top10_all_no_outcomes.png, shap_beeswarm_top10_categorical.png, shap_beeswarm_top10_program_chars.png, shap_beeswarm_top20.png, shap_beeswarm_top20_all_no_outcomes.png, shap_beeswarm_top20_categorical.png, shap_beeswarm_top20_program_chars.png, shap_summary_top10_all_no_outcomes.png, shap_summary_top10_categorical.png, shap_summary_top10_ordered.png, shap_summary_top10_program_chars.png, top10_features_all_data.png, top10_features_all_data_no_outcomes.png, top10_features_program_categorical.png, top10_features_program_chars.png, top25_features_best_model.png

Image: confusion_matrices.png



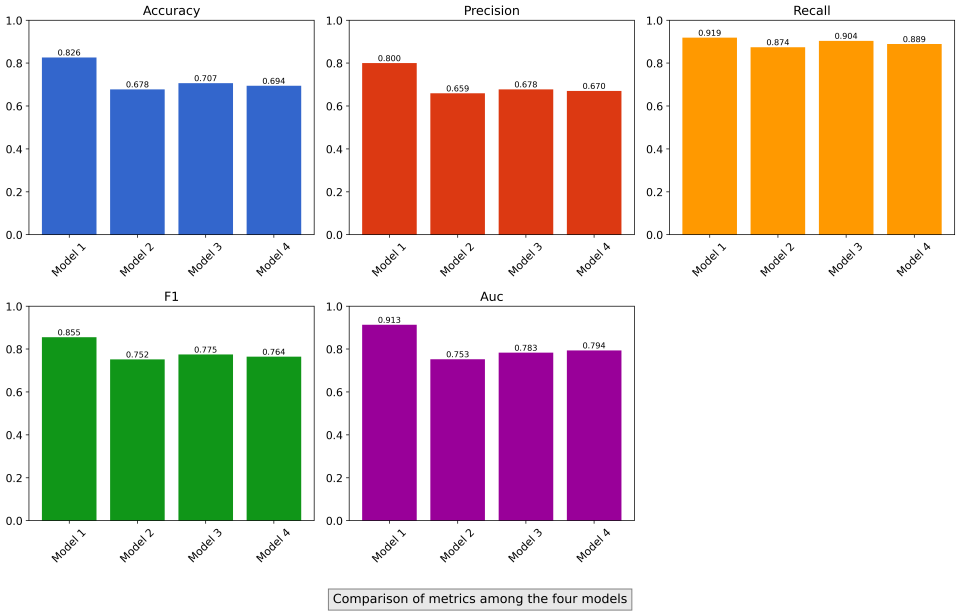
confusion_matrices.png

Image: incorrect_classifications_proba_dist.png



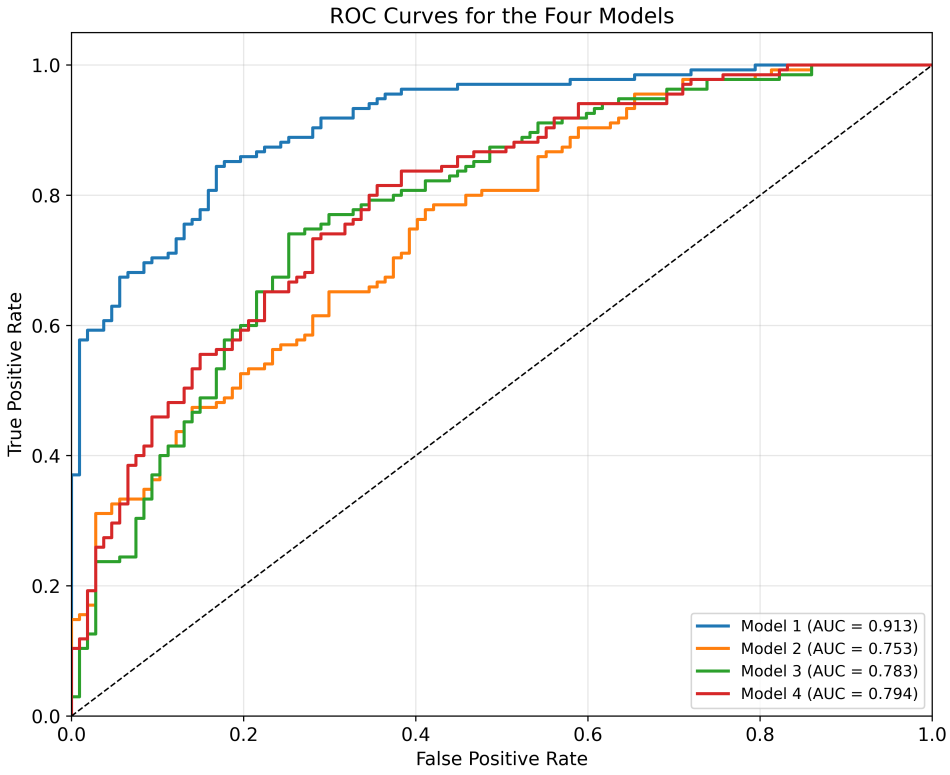
incorrect_classifications_proba_dist.png

Image: metrics_comparison.png



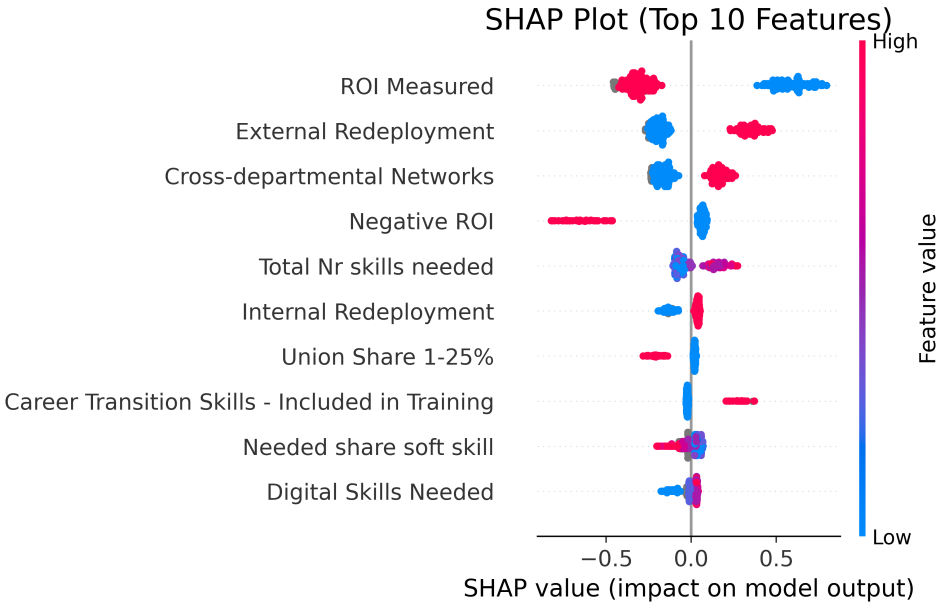
metrics_comparison.png

Image: roc_curves_comparison.png



roc_curves_comparison.png

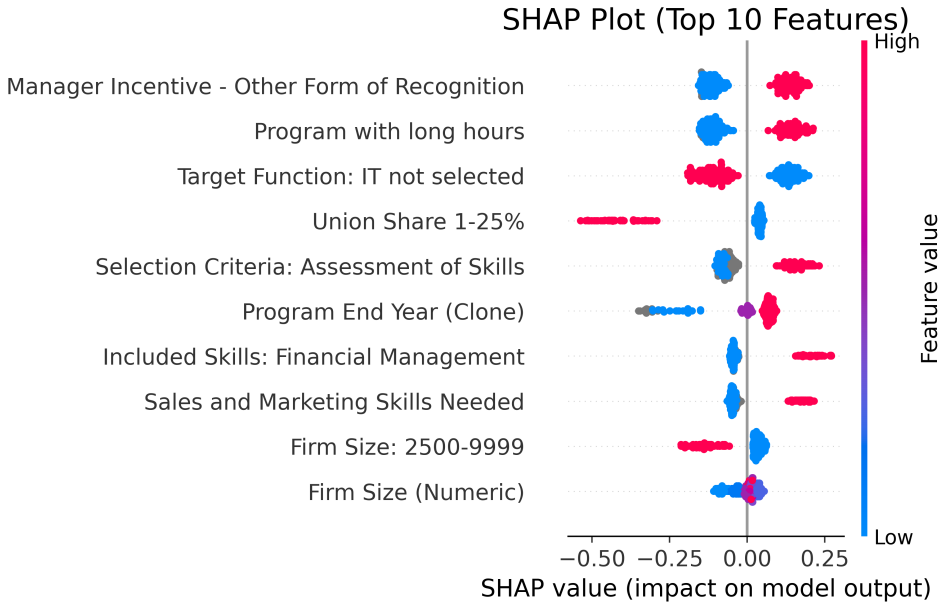
Image: shap_beeswarm_top10.png



Data: All Variables (Including Outcomes), using dummies

shap_beeswarm_top10.png

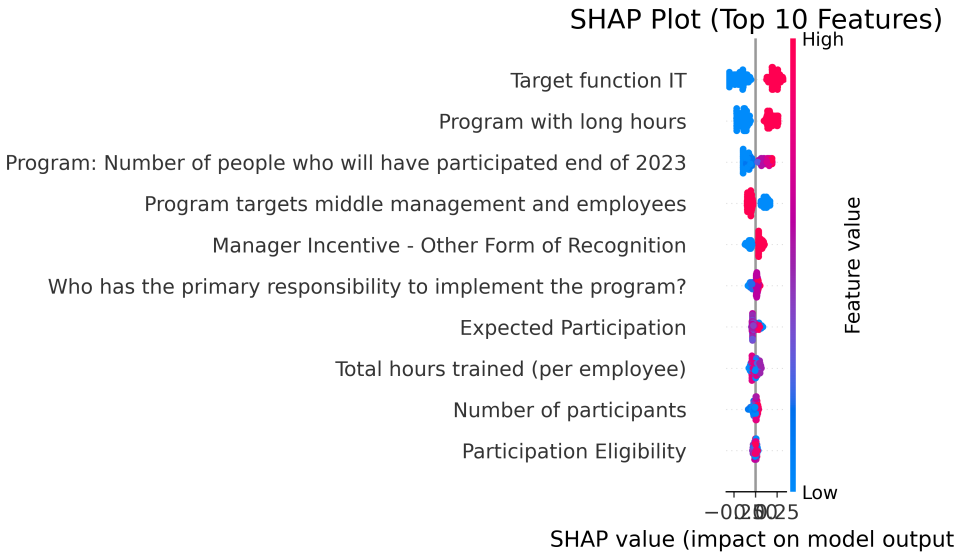
Image: shap_beeswarm_top10_all_no_outcomes.png



Data: All Variables (Without Outcomes), using dummies

shap_beeswarm_top10_all_no_outcomes.png

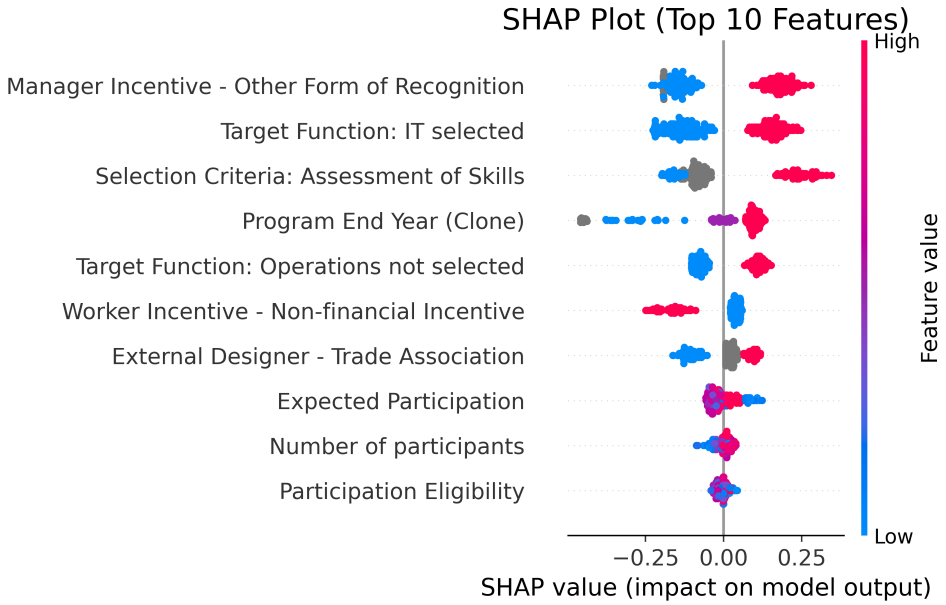
Image: shap_beeswarm_top10_categorical.png



Data: Program Characteristics (With Categorical Encoding, No Dummies)

shap_beeswarm_top10_categorical.png

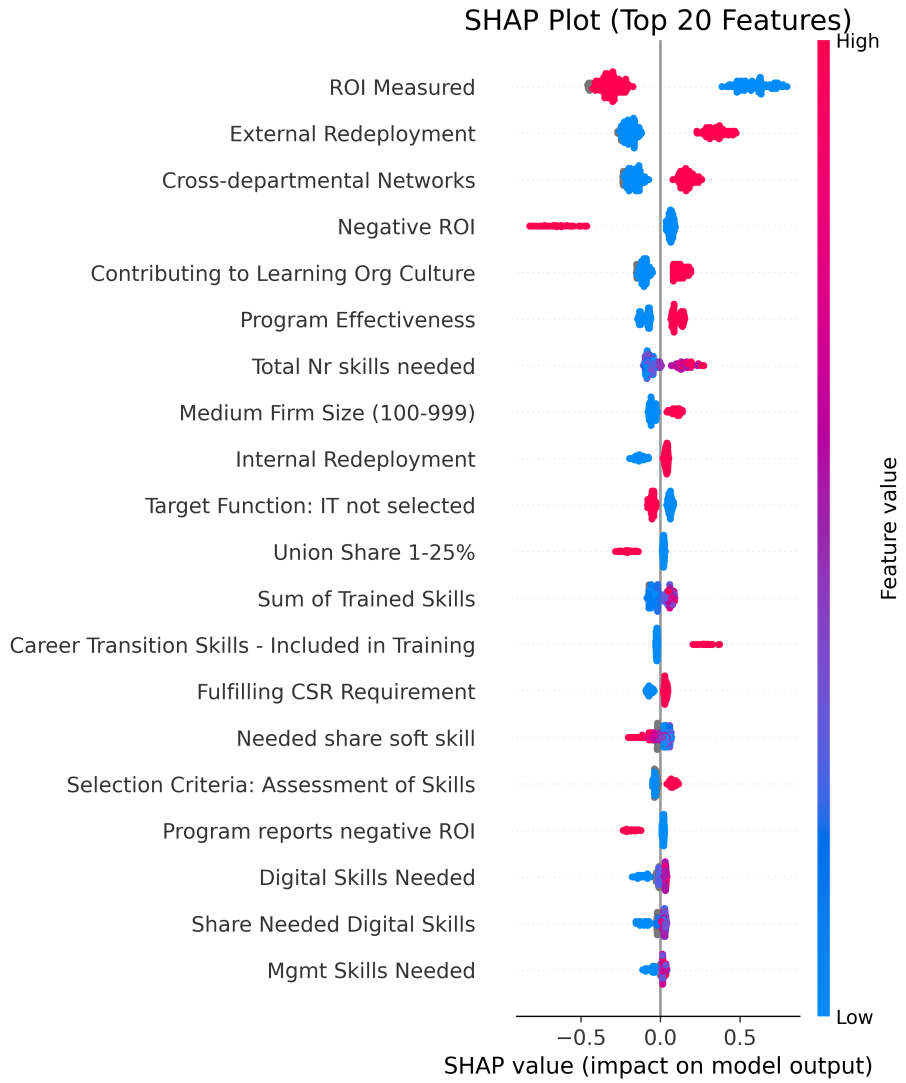
Image: shap_beeswarm_top10_program_chars.png



Data: Program Characteristics (Without Outcomes), using dummies

shap_beeswarm_top10_program_chars.png

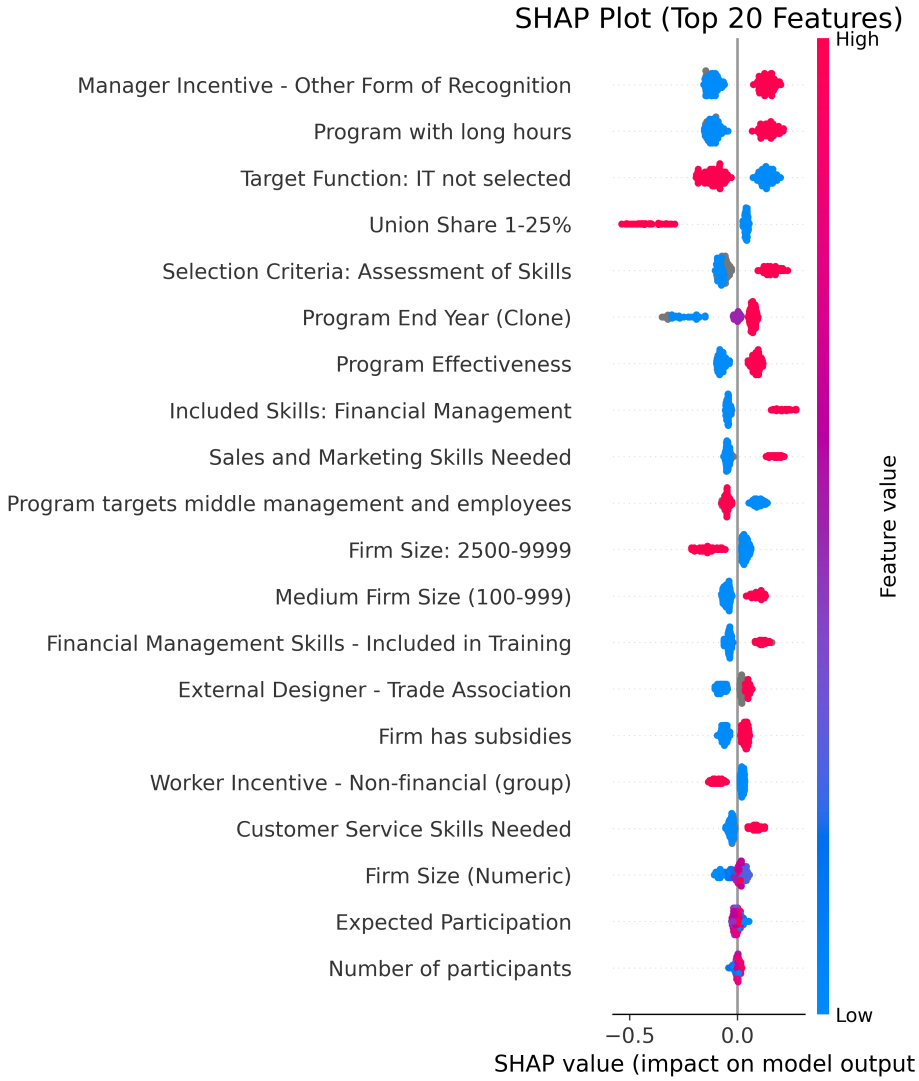
Image: shap_beeswarm_top20.png



Data: All Variables (Including Outcomes), using dummies

shap_beeswarm_top20.png

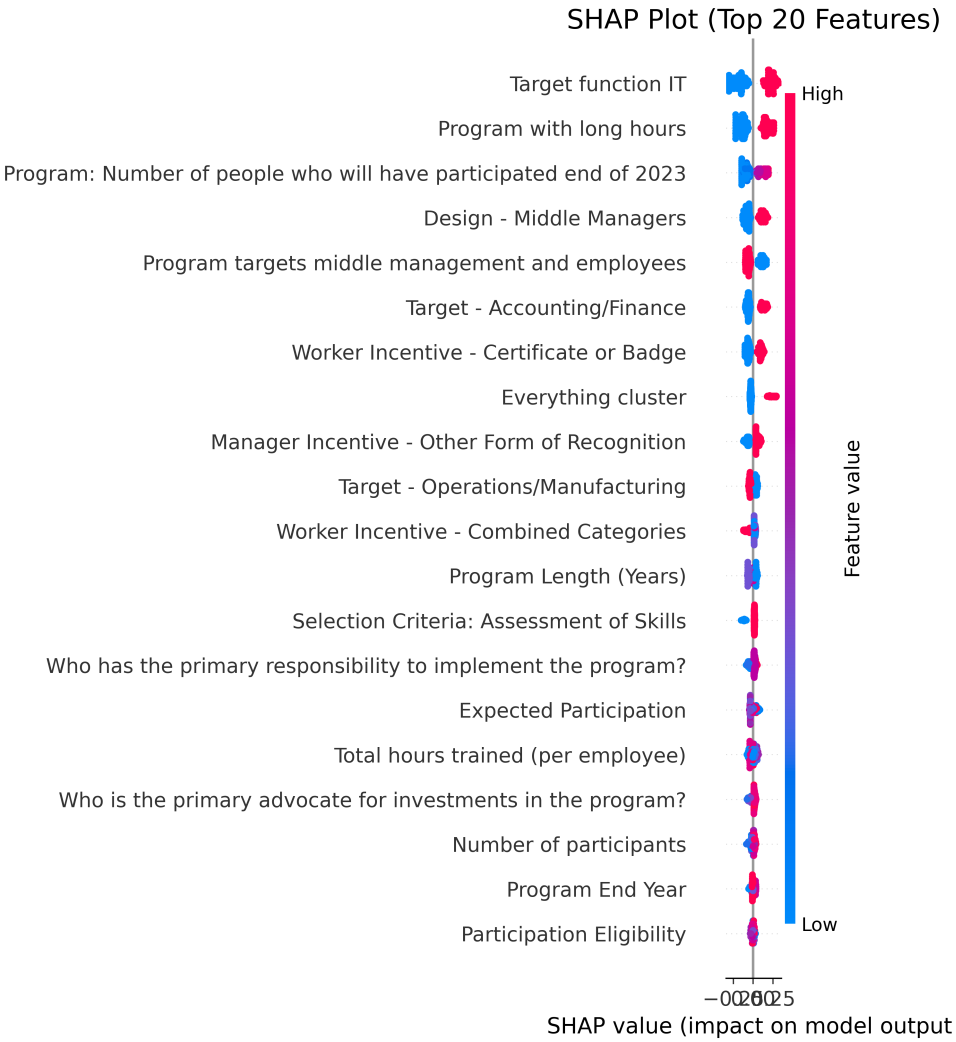
Image: shap_beeswarm_top20_all_no_outcomes.png



Data: All Variables (Without Outcomes), using dummies

shap_beeswarm_top20_all_no_outcomes.png

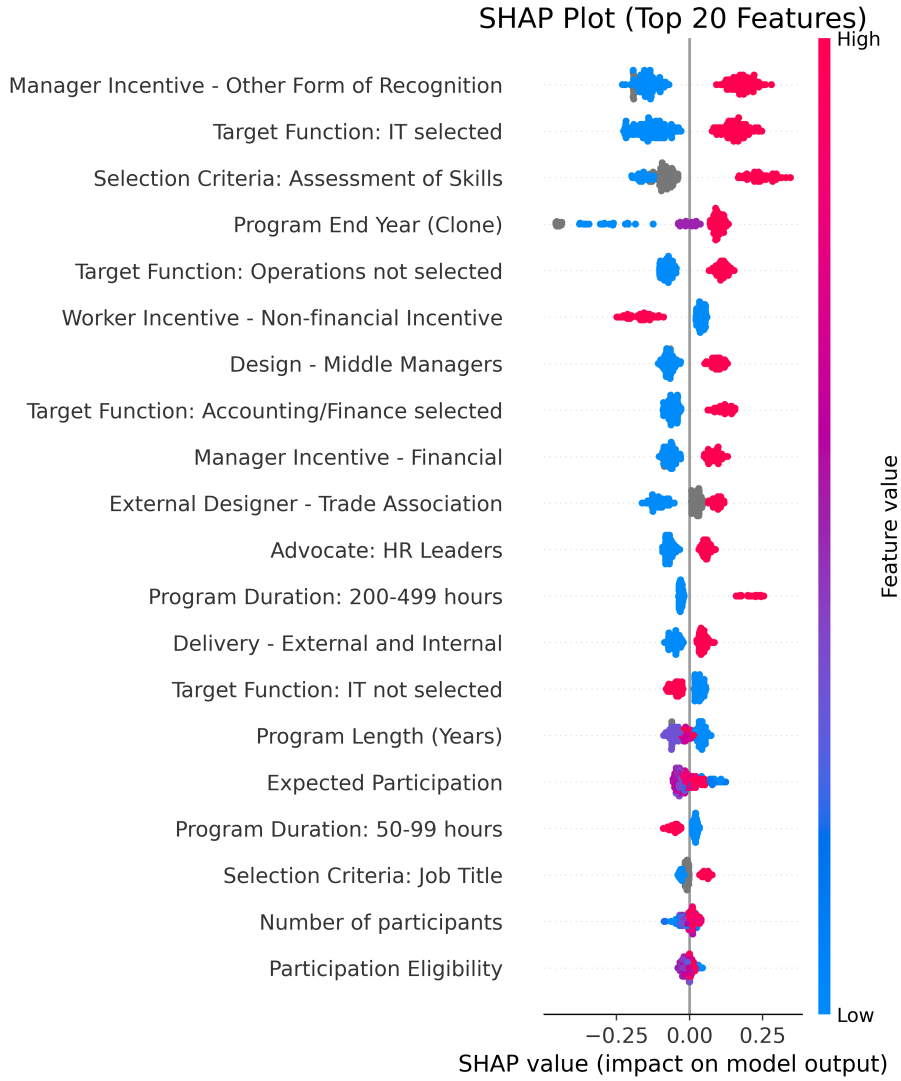
Image: shap_beeswarm_top20_categorical.png



Data: Program Characteristics (With Categorical Encoding, No Dummies)

shap_beeswarm_top20_categorical.png

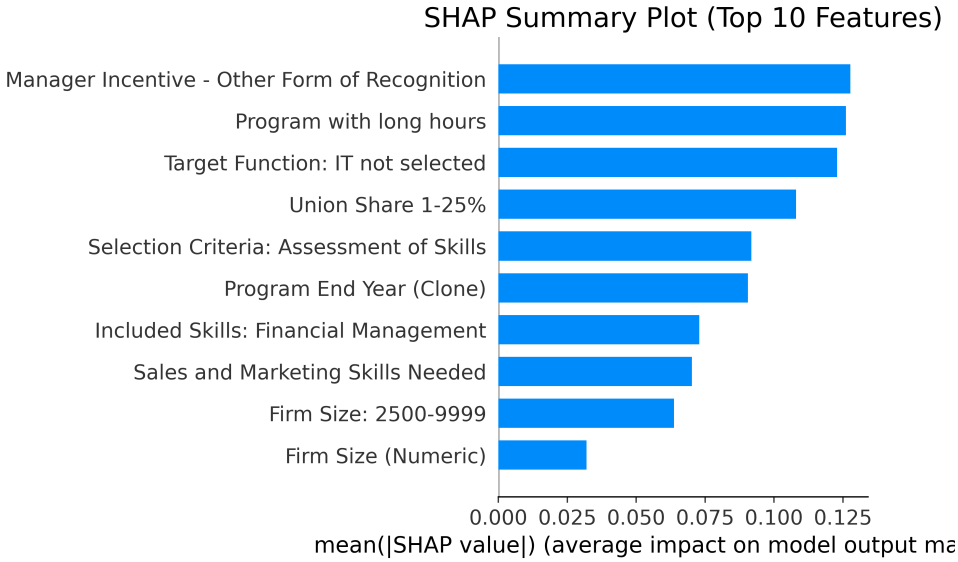
Image: shap_beeswarm_top20_program_chars.png



Data: Program Characteristics (Without Outcomes), using dummies

shap_beeswarm_top20_program_chars.png

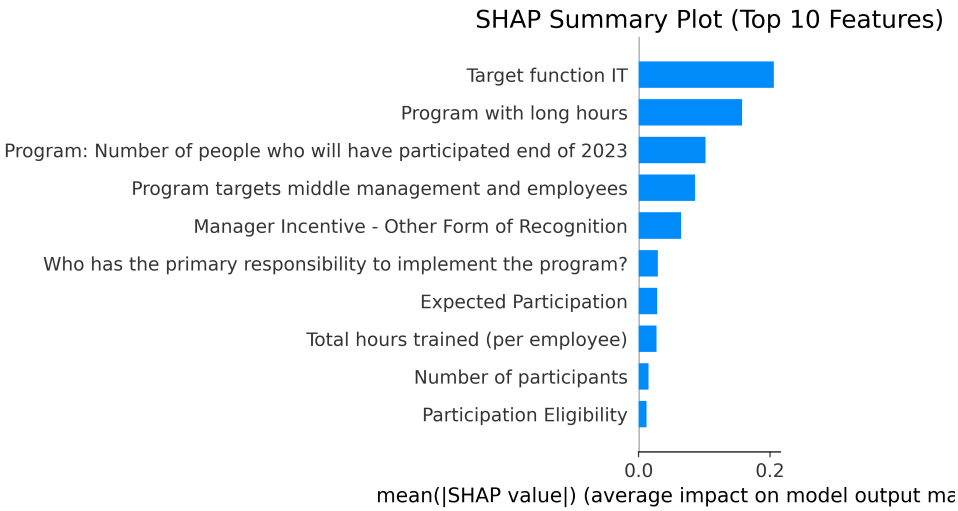
Image: shap_summary_top10_all_no_outcomes.png



Data: All Variables (Without Outcomes), using dummies

shap_summary_top10_all_no_outcomes.png

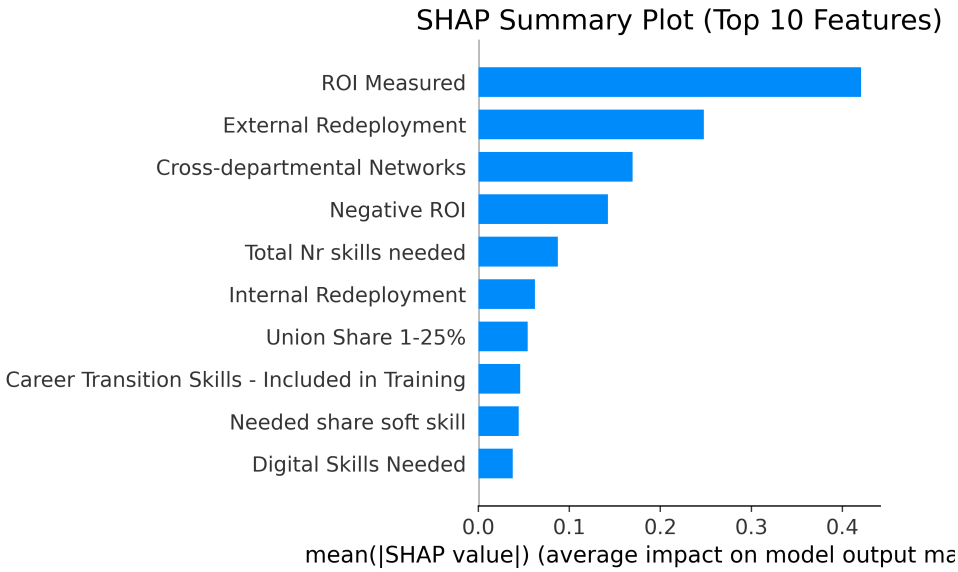
Image: shap_summary_top10_categorical.png



Data: Program Characteristics (With Categorical Encoding, No Dummies)

shap_summary_top10_categorical.png

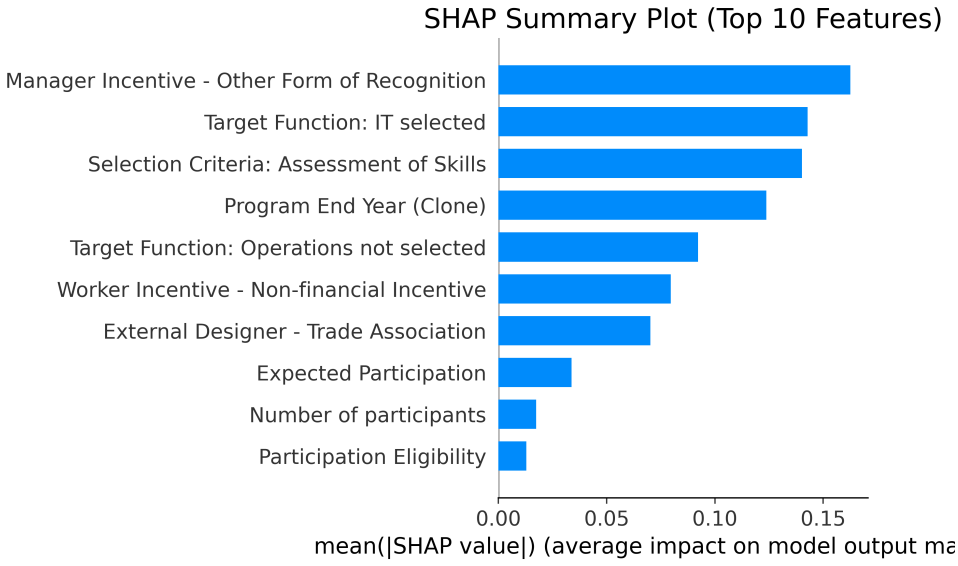
Image: shap_summary_top10_ordered.png



Data: All Variables (Including Outcomes), using dummies

shap_summary_top10_ordered.png

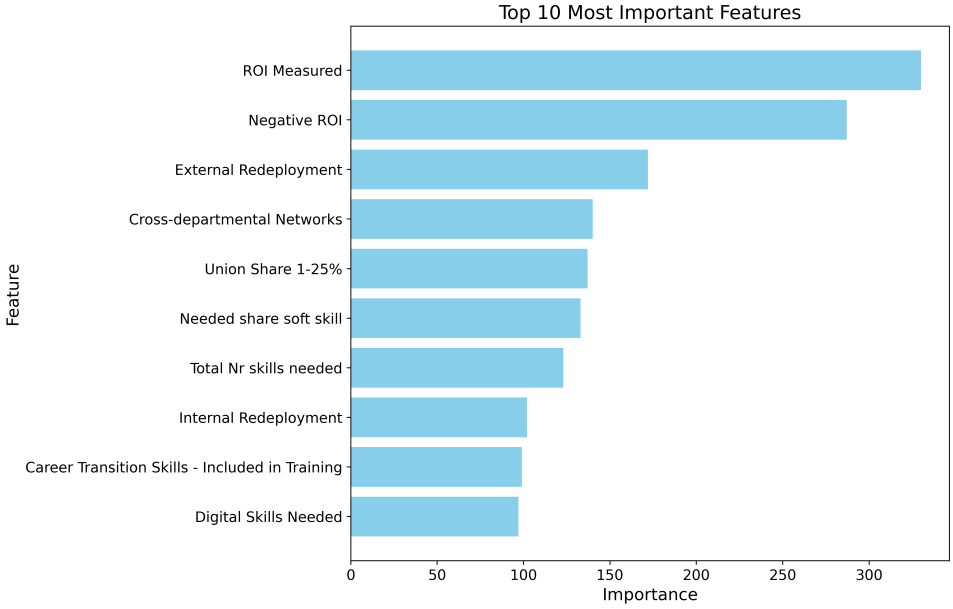
Image: shap_summary_top10_program_chars.png



Data: Program Characteristics (Without Outcomes), using dummies

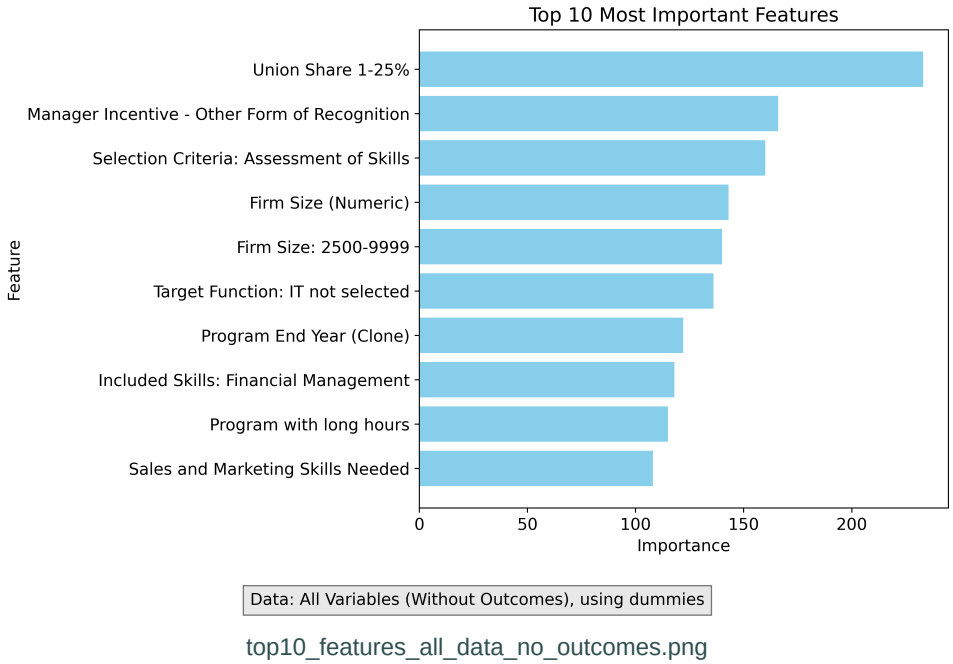
shap_summary_top10_program_chars.png

Image: top10_features_all_data.png



top10_features_all_data.png

Image: top10_features_all_data_no_outcomes.png



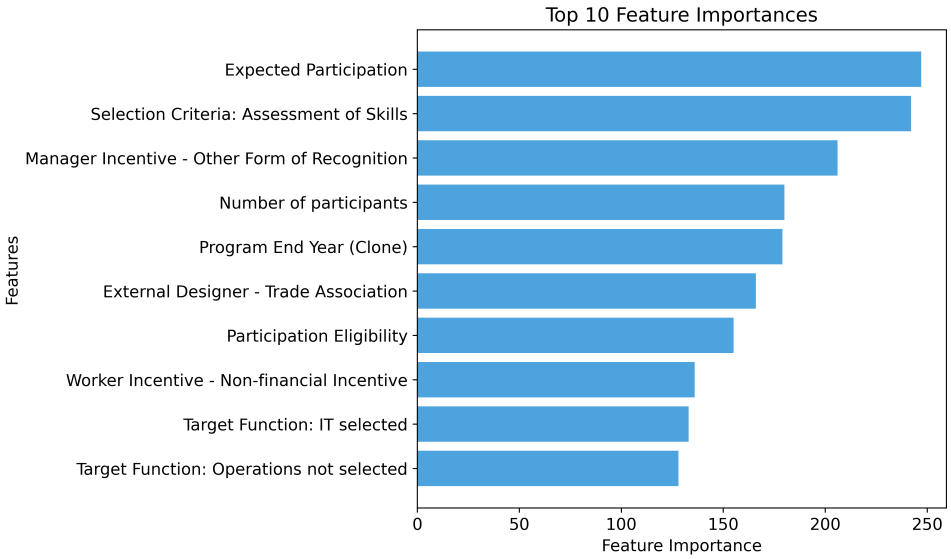
top10_features_all_data_no_outcomes.png

Image: top10_features_program_categorical.png

Data: Program Characteristics (With Categorical Encoding, No Dummies)

top10_features_program_categorical.png

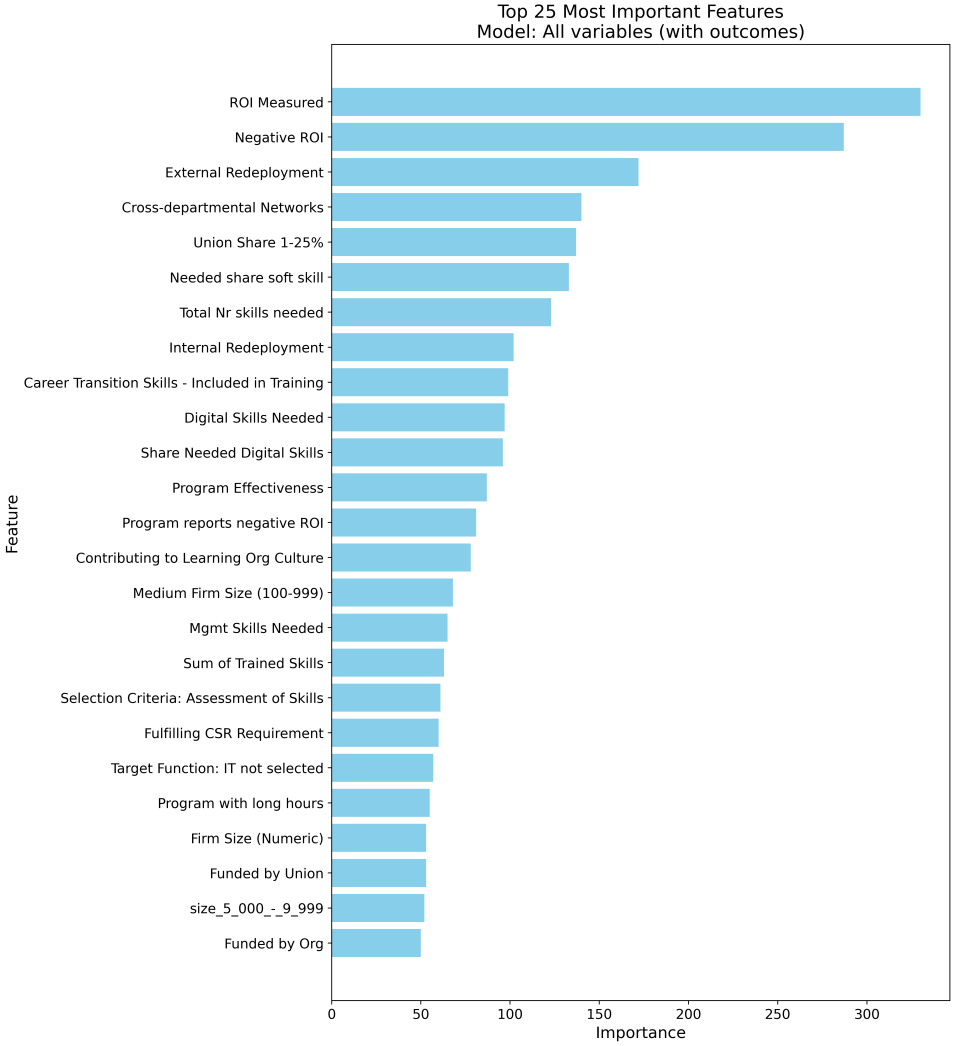
Image: top10_features_program_chars.png



Data: Program Characteristics (Without Outcomes), using dummies

top10_features_program_chars.png

Image: top25_features_best_model.png



Expansion of the analysis to 25 main features

top25_features_best_model.png

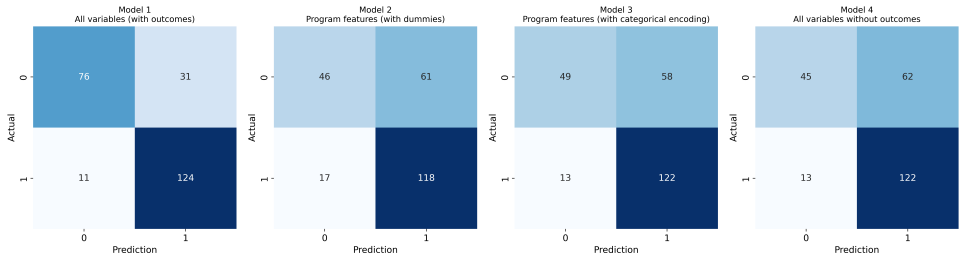
Subdirectory: Model_Comparisons

Directory contains 3 files and 0 subdirectories

Files:

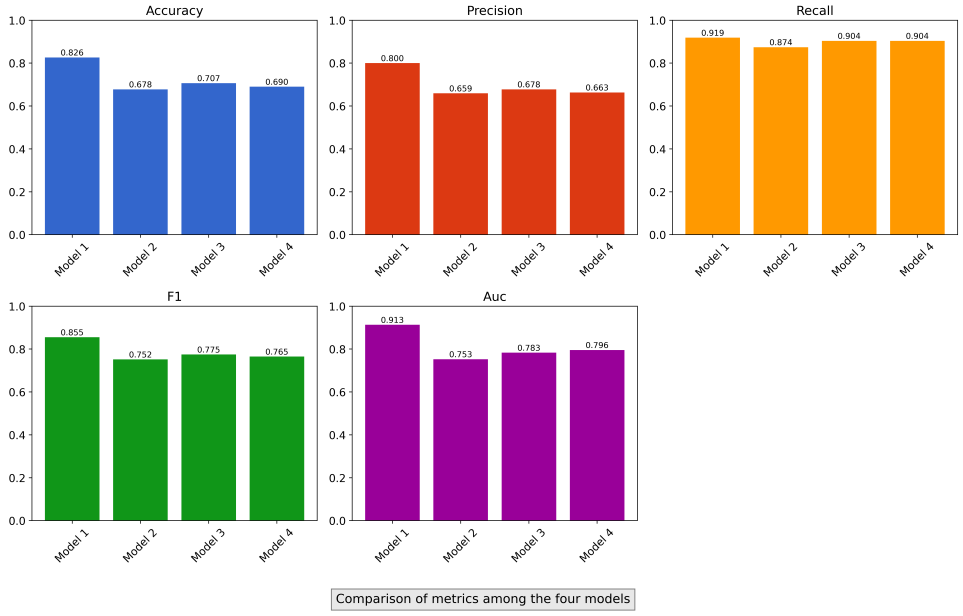
confusion_matrices.png, metrics_comparison.png, roc_curves_comparison.png

Image: confusion_matrices.png



confusion_matrices.png

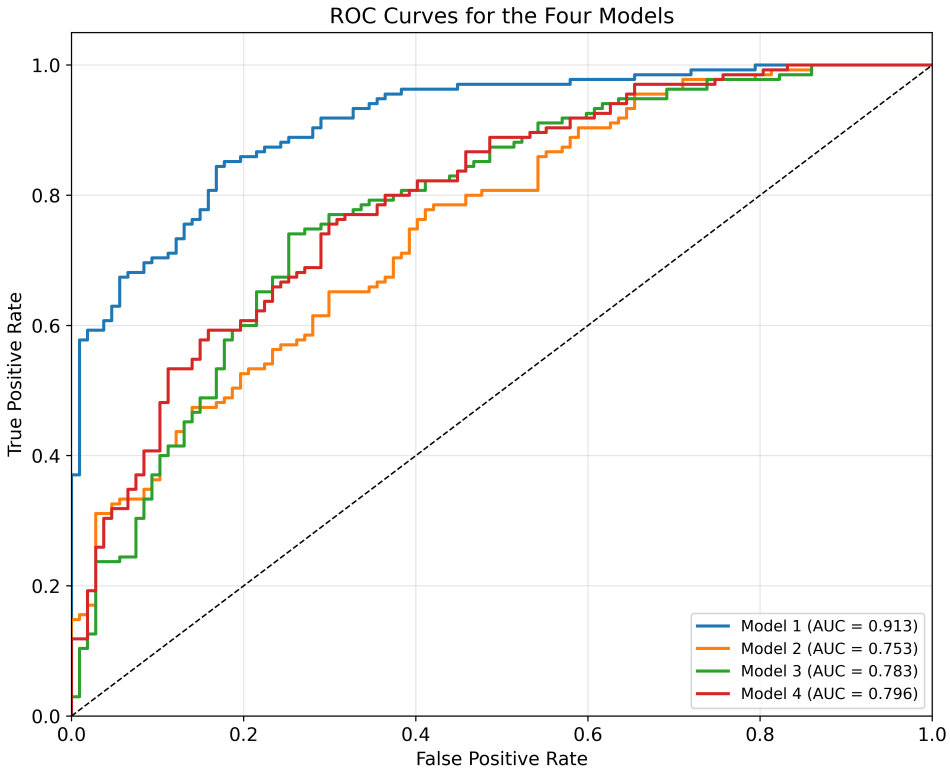
Image: metrics_comparison.png



Comparison of metrics among the four models

metrics_comparison.png

Image: roc_curves_comparison.png



roc_curves_comparison.png

Subdirectory: SHAP_Analysis

Subdirectory: Reports

Directory contains 2 files and 0 subdirectories

Files:

conclusion_report.md, model_performance_summary.md

Markdown File: conclusion_report.md

File content:

Markdown structure:

- # Feature Importance Analysis: Conclusions

- ## Key Findings

- ## Top Features

Content preview:

Feature Importance Analysis: Conclusions ## Key Findings - The model with the best overall performance (AUC) is: **All variables (with outcomes)** - The use of categorical variables with encoding resulted in a **better** performance than using dummies: - AUC with categorical encoding: 0.7832 - AUC with dummies: 0.7526 - Including outcome variables results in an **improvement** of the model: - AUC with outcome variables: 0.9133 - AUC without outcome variables: 0.7956 This analysis demonstrates the importance of feature selection and categorical variable encoding in model performance for distinguishing between Upskilling and Reskilling programs. ## Top Features The top 5 most important features for distinguishing between program types are: 1. **ROI Measured** (importance: 330.0000) 2. **Negative ROI** (importance: 287.0000) 3. **External Redeployment** (importance: 172.0000) 4. **Cross-departmental Networks** (importance: 140.0000) 5. **Union Share 1-25%** (importance: ...

Document Summary:

The document "conclusion_report.md" presents key findings and conclusions from a feature importance analysis related to distinguishing between Upskilling and Reskilling programs. It discusses the impact of feature selection strategies such as including outcome variables and categorical variable encoding on model performance, highlighting improvements in model accuracy. The top 5 most important features for distinguishing between program types are identified. The purpose of this document is to provide insights on the importance of these features in program classification, aiding in decision-making processes and model refinement within the project.

Markdown File: model_performance_summary.md

File content:

Markdown structure:

- # Model Performance Summary

Content preview:

Model Performance Summary | Model | Accuracy | Precision | Recall | F1 | AUC | |-------|----------|-----------|--------|----|---------| | Model 1: All variables (with outcomes) | 0.8264 | 0.8000 | 0.9185 | 0.8552 | 0.9133 | | Model 2: Program features (with dummies) | 0.6777 | 0.6592 | 0.8741 | 0.7516 | 0.7526 | | Model 3: Program features (with categorical encoding) | 0.7066 | 0.6778 | 0.9037 | 0.7746 | 0.7832 | | Model 4: All variables without outcomes | 0.6901 | 0.6630 | 0.9037 | 0.7649 | 0.7956 |

Document Summary:

The document "model_performance_summary.md" presents a summary of model performance metrics for different models used in a project. The table includes information on accuracy, precision, recall, F1 score, and AUC for each model variant considered. The purpose of this document is to provide a quick reference for comparing the performance of various model configurations in terms of their predictive capabilities. Stakeholders can use this summary to evaluate and select the most effective model based on the specified metrics.

Subdirectory: Statistics

Directory contains 8 files and 0 subdirectories

Files:

confusion_matrix_model_1.csv, confusion_matrix_model_2.csv, confusion_matrix_model_3.csv, confusion_matrix_model_4.csv, misclassification_probabilities.csv, misclassification_summary.csv, model_performance_metrics.csv, top_features_best_model.csv

CSV File: confusion_matrix_model_1.csv

Contains 2 rows and 3 columns

Columns:

Unnamed: 0, Predicted 0, Predicted 1

Data preview (first 10 rows):

Unnamed: 0	Predicted 0	Predicted 1
Actual 0	76	31
Actual 1	11	124

Possible data interpretation:

The file "confusion_matrix_model_1.csv" likely contains a confusion matrix for a binary classification model. The columns "Predicted 0" and "Predicted 1" suggest that the model predicted two classes, 0 and 1. This file can provide insights into the model's performance by showing the actual and predicted counts for each class. Hypothesis: Based on the confusion matrix data, we can hypothesize that the model might be performing well if there are high counts along the diagonal elements (true positives and true negatives) and low counts off the diagonal (false positives and false negatives). Conversely, low counts on the diagonal and high counts off the diagonal may indicate potential areas where the model can be improved.

CSV File: confusion_matrix_model_2.csv

Contains 2 rows and 3 columns

Columns:

Unnamed: 0, Predicted 0, Predicted 1

Data preview (first 10 rows):

Unnamed: 0	Predicted 0	Predicted 1
Actual 0	46	61
Actual 1	17	118

Possible data interpretation:

From the file name "confusion_matrix_model_2.csv" and the column headers "Predicted 0" and "Predicted 1," it seems that this file contains a confusion matrix for a classification model where rows represent actual class 0 and class 1, and columns represent predicted class 0 and class 1. Insights from this data could include the model's performance in terms of correctly identifying class 0 and class 1 instances, as well as metrics like accuracy, precision, recall, and F1-score. Hypothesis: Based on the confusion matrix data, we could hypothesize that Model 2 might perform better in correctly predicting class 1 instances compared to class 0, which could indicate a specific bias or imbalance in the model's predictions. Further analysis of precision and recall values could provide more detailed insights into the model's strengths and weaknesses in differentiating between the two classes.

CSV File: confusion_matrix_model_3.csv

Contains 2 rows and 3 columns

Columns:

Unnamed: 0, Predicted 0, Predicted 1

Data preview (first 10 rows):

Unnamed: 0	Predicted 0	Predicted 1
Actual 0	49	58
Actual 1	13	122

Possible data interpretation:

This CSV file likely contains a confusion matrix for a model labeled as "model_3". A confusion matrix is a table that is often used to describe the performance of a classification model. The columns "Predicted 0" and "Predicted 1" likely refer to the predicted classes of the model. Insights that can be derived from this confusion matrix include metrics such as true positives, false positives, true negatives, false negatives, accuracy, precision, recall, and F1 score. These metrics can help evaluate how well the model is performing in terms of correctly predicting the positive and negative classes. A hypothesis based on this confusion matrix could be that Model 3 performs well in terms of precision but may have lower recall, indicating that it correctly identifies positive instances but may miss some actual positive cases. Further analysis could focus on improving the recall of the model, possibly by adjusting the threshold for classification or by collecting more diverse training data.

CSV File: confusion_matrix_model_4.csv

Contains 2 rows and 3 columns

Columns:

Unnamed: 0, Predicted 0, Predicted 1

Data preview (first 10 rows):

Unnamed: 0	Predicted 0	Predicted 1
Actual 0	45	62
Actual 1	13	122

Possible data interpretation:

Based on the file name and column headers, "confusion_matrix_model_4.csv" likely contains data related to a confusion matrix for a predictive model. The columns "Predicted 0" and "Predicted 1" suggest that the matrix shows the counts of true positive, false negative predictions for class 0 and true negative, false positive predictions for class 1. This file provides valuable information on the performance of a classification model, allowing us to assess how well the model is predicting each class. By analyzing the confusion matrix, we can calculate metrics such as accuracy, precision, recall, and F1 score, which are crucial for evaluating the model's effectiveness across different classes. Hypothesis: Model 4 may have higher precision but lower recall compared to other models, indicating that it is better at correctly predicting class 1 instances but may miss some class 1 instances. This hypothesis would require further analysis using metrics derived from the confusion matrix to validate the performance of Model 4.

CSV File: misclassification_probabilities.csv

Contains 42 rows and 1 columns

Columns:

Probability

Data preview (first 10 rows):

Probability
0.66989464
0.711496
0.6635022
0.5449302
0.5422413
0.586122
0.721017
0.5673443
0.6196851
0.40239555

Possible data interpretation:

Based on the file name "misclassification_probabilities.csv" and the column header "Probability," it seems that this data might contain probabilities related to misclassifications in a classification model. The probabilities could represent the likelihood of a certain observation being misclassified by the model. Insights from this data could help in understanding the model's performance, identifying which observations are more prone to misclassification, and potentially improving the model by focusing on areas with higher misclassification probabilities. Hypothesis: The misclassification probabilities in the dataset may reveal patterns where certain classes or features consistently lead to higher misclassification rates. By analyzing these probabilities and identifying common characteristics of misclassified instances, we could potentially adjust the model or provide targeted interventions to improve its accuracy.

CSV File: misclassification_summary.csv

Contains 1 rows and 5 columns

Columns:

Total Samples, Correct Classifications, Correct Classifications (%), Incorrect Classifications, Incorrect Classifications (%)

Data preview (first 10 rows):

Total Samples	Correct Classifications	Correct Classifications (%)	Incorrect Classifications	Incorrect Classifications (%
242.0	200.0	82.64462809917356	42.0	17.355371900826448

Possible data interpretation:

Based on the file name and column headers, "misclassification_summary.csv" likely contains data related to the classification results of a model. It provides information on the total number of samples, correct classifications, and incorrect classifications in both absolute numbers and percentages. Insights from this file could help evaluate the performance of a classification model by analyzing the misclassification rates and understanding the distribution of correct and incorrect classifications across different classes or categories. A hypothesis based on this data could be that the model's performance varies significantly across different classes or categories, leading to higher misclassification rates in certain groups. Further analysis might reveal patterns or trends in misclassifications that could be used to improve the model's performance for specific classes.

CSV File: top_features_best_model.csv

Contains 25 rows and 2 columns

Columns:

Feature, Importance

Data preview (first 10 rows):

Feature	Importance
ROI Measured	330.0
Negative ROI	287.0
External Redeployment	172.0
Cross-departmental Networks	140.0
Union Share 1-25%	137.0
Needed share soft skill	133.0
Total Nr skills needed	123.0
Internal Redeployment	102.0
Career Transition Skills - Included in Training	99.0
Digital Skills Needed	97.0

Possible data interpretation:

The file "top_features_best_model.csv" likely contains data related to the top features and their importance as identified by the best model in a machine learning analysis. The "Feature" column probably lists the names of the features, and the "Importance" column likely contains numerical values indicating the importance of each feature according to the model. Insights from this data could provide information on which features have the most influence on the model's predictions or classification. Understanding the importance of specific features can help in feature selection, model optimization, and gaining insights into the underlying patterns in the data. Hypothesis: The top features identified by the best model are likely to have the most significant impact on the model's performance and accuracy. By focusing on these features during feature engineering or model tuning, we can potentially improve the model's predictive power and generalization performance.

Conclusions and Recommendations

Conclusion:

The data science project analyzed sought to uncover patterns, relationships, and insights within the dataset through a combination of clustering, classification, and feature importance techniques. The primary objective appeared to involve understanding the underlying structure of the data, identifying significant features, and developing predictive models to improve decision-making processes.

Through the utilization of metrics such as accuracy, precision, recall, and F1-score, the project assessed the performance of various models in classifying data points into distinct clusters or categories. The evaluation of clustering algorithms provided a deeper understanding of the inherent groups within the data, aiding in segmentation and targeted marketing strategies.

Feature importance analysis played a crucial role in identifying the variables that most significantly influenced the outcomes of the classification models. By ranking features based on their impact, the project was able to highlight key drivers of predictive accuracy and gain valuable insights for decision-makers.

The results derived from this project offer several implications for business strategy and operational efficiency. By leveraging the identified clusters and classification models, organizations can tailor their marketing campaigns, optimize resource allocation, and enhance customer segmentation strategies. Understanding feature importance can lead to the development of more focused and effective intervention strategies.

To further enhance the findings of this project, future investigations could explore ensemble methods to improve model performance, conduct deeper dives into the misclassification patterns to uncover hidden insights, and incorporate external datasets for a more comprehensive analysis. Additionally, delving into the interpretability of the clustering results and refining the feature selection process could provide more actionable recommendations for stakeholders.

In conclusion, this data science project successfully employed clustering, classification, and feature importance techniques to extract meaningful patterns from the dataset. The findings present valuable opportunities for organizations to optimize their strategies, improve decision-making processes, and drive business growth. By continuing to explore these analytical approaches and expanding the scope of analysis, businesses can stay ahead in today's data-driven environment.

Recommendations for Next Steps

- **Optimize Feature Selection**
- Explore advanced feature selection techniques such as Recursive Feature Elimination or Principal Component Analysis to enhance model performance and interpretability. Conduct a detailed analysis to identify key features and eliminate redundant or irrelevant ones.
- **Fine-tune Cluster Evaluation**
- Experiment with different clustering algorithms and evaluate their performance using a variety of metrics such as Silhouette Score, Dunn Index, or Davies-Bouldin Index to determine the optimal number of

clusters. Create visualizations like dendrograms or scatter plots to better understand the cluster structures.

- **Implement Ensemble Methods**
- Integrate ensemble methods like Random Forest or Gradient Boosting for classification tasks to improve predictive accuracy and robustness. Conduct model stacking or blending to combine the strengths of multiple models and achieve better overall performance.
- **Enhance Interpretability**
- Utilize techniques such as SHAP values, Partial Dependency Plots, or LIME to gain insights into feature importance and enhance model interpretability. Create detailed visualizations and summary reports explaining the impact of different features on model predictions.
- **Automate Reporting and Exporting**
- Develop scripts to automate the generation of comprehensive result summaries in various formats like Excel spreadsheets (XLSX) or PDFs to facilitate easy sharing and interpretation of key findings. Include detailed tables, charts, and explanations to provide a clear overview of the analysis process and results.