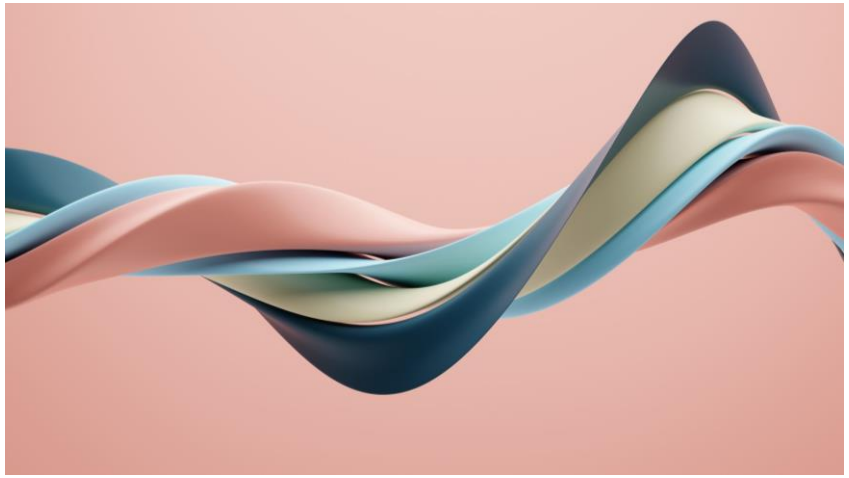


A Predictive Model to Forecast Employee Churn for HR Analytics

FINAL PAPER - PRESENTATION

BY SUDEEP SAPKOTA – UNCC 2024 SPRING– AML(ITCS 5156)

04/19/2024

Outline

- Problem & Challenges
- Motivation
- Existing related approaches
- The method
- Results and observation
- Conclusion and future work

Background and Original Research Information

- Authors: Vengai Musanga1 and Colin Chibaya
- Year- 2023
- Conference/Journal Name: Proceedings of NEMISA Digital Skills Conference 2023: Scaling Data Skills For Multidisciplinary Impact

Problem and Dataset

- **Problem** :
- Employee churn is a major issue for businesses and organizations.
- The departure of valuable employees can harm service delivery, customer loyalty, and productivity.
- Predicting employee churn is crucial for retaining valuable employees

- **Dataset**: IBM HR Analytics Employee Attrition & Performance which contains employee data for 1,470 employees with various information about the employees. This dataset to predict when employees are going to quit by understanding the main drivers of employee churn.

- Link to the dataset : WA_Fn-UseC_-HR-Employee-Attrition.xlsx (live.com)

Motivation

- Retaining skilled employees is essential for organizational success
- Machine learning can analyze past employee data to predict churn
- Improved retention strategies and cost savings can be achieved through accurate predictions

Existing Related Approaches

- Previous studies have explored machine learning techniques for employee churn prediction.
- Logistic regression identified key factors such as job satisfaction and work-life balance.
- Random forest and support vector machine models showed strong performance in predicting churn.

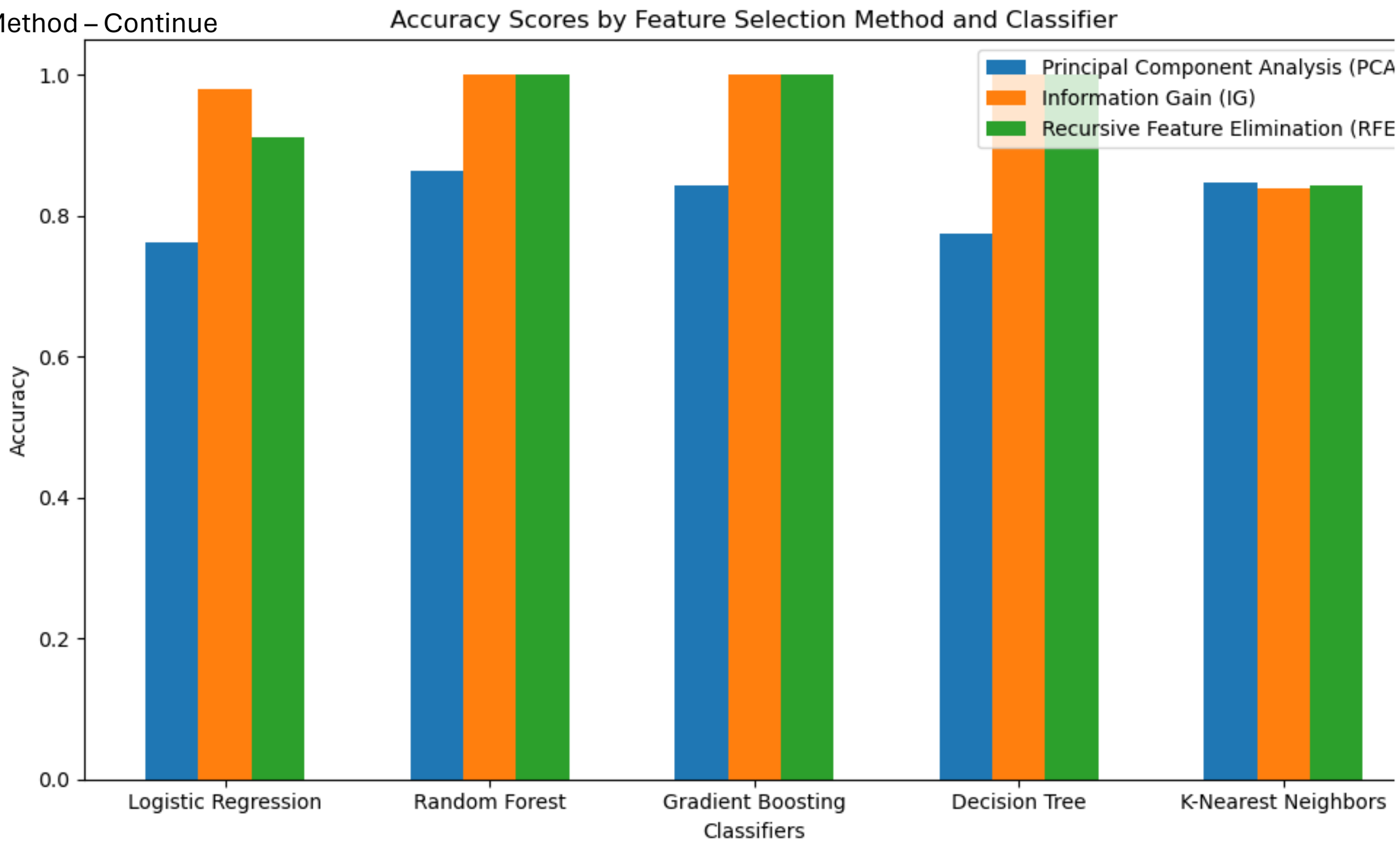
- 1st Paper:
- **Title**: APPLICATION OF ADAPTIVE CROSS VALIDATION AND PRINCIPAL COMPONENT ANALYSIS OPTIMIZATION FOR EMPLOYEE TURNOVER PREDICTION USING ENHANCED GRAPH EMBEDDING
- **Authors**: ABDULLAHI JIBRIL ABDULLAHI, NASIMA IBRAHIM and AISHA FARIDA AHMAD
- **Year**: March, 2023
- **Conference/Journal name** : Bima Journal of Science and Technology, Vol. 7 (1) Mar, 2023 ISSN: 2536-6041

- 2nd Paper:
- **Title**:Predictive Analytics of Employee Attrition using K-Fold Methodologies
- **Authors**: Vijayalakshmi Kakulapati, Subhani Shaik
- **Year**: January, 2023
- **Conference/Journal name** :https://www.researchgate.net/publication/368308492_Predictive_Analytics_of_Employee_Attrition_using_K-Fold_Methodologies

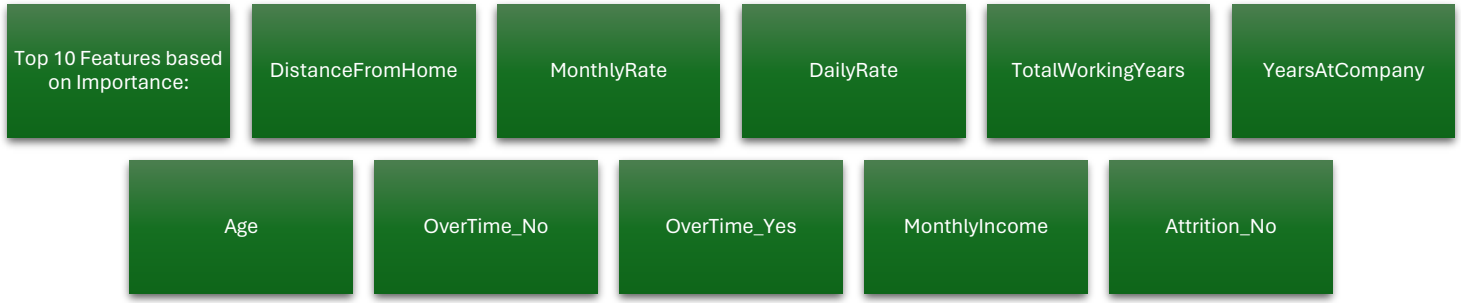
The Method

- Proposed predictive model using machine learning
- Feature selection through Pearson correlation, information gain, and recursive feature elimination
- Classification methods: random forest, logistic regression, decision trees, gradient boosting machines, and K-nearest neighbors

Method – Continue



Results and Observation



Feature Selection Method	Logistic Regression	Random Forest	Gradient Boosting	Decision Tree	K-Nearest Neighbors
Principal Component Analysis (PCA)	76.19%	86.39%	84.35%	77.55%	84.69%
Information Gain (IG)	97.96%	100.00%	100.00%	100.00%	84.01%
Recursive Feature Elimination (RFE)	91.16%	100.00%	100.00%	100.00%	84.35%

Continue .. Results and Observation

-

In [17]:

```python
# Top 10 Features

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.decomposition import PCA
from sklearn.feature_selection import SelectKBest, mutual_info_classif, RFE

# Load the dataset
data = pd.read_excel("WA_Fn-UseC_-HR-Employee-Attrition.xlsx")

# Data preprocessing
# Perform any necessary data cleaning and feature engineering

# Columns to remove
columns_to_remove = ['EmployeeCount', 'EmployeeNumber', 'Over18', 'StandardHours']

# Remove irrelevant columns
data = data.drop(columns_to_remove, axis=1)

# Convert categorical variables to numerical using one-hot encoding
data_encoded = pd.get_dummies(data)

# Split the data into features (X) and target variable (y)
X = data_encoded.drop("Attrition_Yes", axis=1)  # Assuming "Attrition_Yes" is the encoded column for "Attrition" with 'Ye
y = data_encoded["Attrition_Yes"]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize and train the Random Forest classifier
rf_classifier = RandomForestClassifier()
rf_classifier.fit(X_train, y_train)

# Feature importance
feature_importance = rf_classifier.feature_importances_
feature_names = X.columns.values
sorted_indices = np.argsort(feature_importance)

# Display top 10 features based on importance
top_10_features = feature_names[sorted_indices][-10:]
print("Top 10 Features based on Importance:")
for feature in top_10_features:
    print(feature)

# Make predictions on the test set
rf_predictions = rf_classifier.predict(X_test)

# Calculate accuracy score
rf_accuracy = accuracy_score(y_test, rf_predictions)
print("RF Accuracy Score (Original Features):", rf_accuracy)
```

```
Top 10 Features based on Importance:
DistanceFromHome
MonthlyRate
DailyRate
TotalWorkingYears
YearsAtCompany
Age
```

In [21]:

```python
## Compare and contrast the accuracy result with different feature selection methods and algorithms
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.decomposition import PCA
from sklearn.feature_selection import SelectKBest, mutual_info_classif, RFE
from tabulate import tabulate

# Load the dataset
data = pd.read_excel("WA_Fn-UseC_-HR-Employee-Attrition.xlsx")

# Data preprocessing
# Perform any necessary data cleaning and feature engineering

# Columns to remove
columns_to_remove = ['EmployeeCount', 'EmployeeNumber', 'Over18', 'StandardHours']

# Remove irrelevant columns
data = data.drop(columns_to_remove, axis=1)

# Convert categorical variables to numerical using one-hot encoding
data_encoded = pd.get_dummies(data)

# Split the data into features (X) and target variable (y)
X = data_encoded.drop("Attrition_Yes", axis=1)  # Assum
y = data_encoded["Attrition_Yes"]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, t

# Feature selection using Principal Component Analysis (PCA)
pca = PCA(n_components=10)  # Selecting top 10 principal compo
X_train_pca = pca.fit_transform(X_train)
X_test_pca = pca.transform(X_test)

# Feature selection using Information Gain
selector_ig = SelectKBest(score_func=mutual_info_classif, k=10)  # Selecting t
X_train_ig = selector_ig.fit_transform(X_train, y_train)
X_test_ig = selector_ig.transform(X_test)

# Feature selection using Recursive Feature Elimination (RFE)
rf_classifier = RandomForestClassifier()  # RF classifier for RFE
selector_rfe = RFE(rf_classifier, n_features_to_select=10)  # Selecting top
X_train_rfe = selector_rfe.fit_transform(X_train, y_train)
X_test_rfe = selector_rfe.transform(X_test)
```

```python
# Initialize and train the classifiers
classifiers = {
    "Logistic Regression": LogisticRegression(),
    "Random Forest": RandomForestClassifier(),
    "Gradient Boosting": GradientBoostingClassifier(),
    "Decision Tree": DecisionTreeClassifier(),
    "K-Nearest Neighbors": KNeighborsClassifier()
}

feature_selection_methods = {
    "Principal Component Analysis (PCA)": (X_train_pca, X_test_pca),
    "Information Gain (IG)": (X_train_ig, X_test_ig),
    "Recursive Feature Elimination (RFE)": (X_train_rfe, X_test_rfe)
}

accuracy_scores = []

for feature_method, (X_train_selected, X_test_selected) in feature_selection_methods.items():
    row = [feature_method]
    for classifier_name, classifier in classifiers.items():
        classifier.fit(X_train_selected, y_train)
        y_pred = classifier.predict(X_test_selected)
        accuracy = accuracy_score(y_test, y_pred)
        row.append(accuracy)
    accuracy_scores.append(row)

# Print results in a table
headers = ["Feature Selection Method"] + list(classifiers.keys())
print(tabulate(accuracy_scores, headers=headers, floatfmt=".2%"))

# Plot accuracy scores in a chart
classifiers_names = list(classifiers.keys())
x = np.arange(len(classifiers_names))
width = 0.2

plt.figure(figsize=(10, 6))
for i, (method, scores) in enumerate(zip(feature_selection_methods.keys(), accuracy_scores)):
    plt.bar(x + i * width, scores[1:], width, label=method)

plt.xlabel("Classifiers")
plt.ylabel("Accuracy")
plt.title("Accuracy Scores by Feature Selection Method and Classifier")
plt.xticks(x + width * (len(feature_selection_methods) / 2 - 0.5), classifiers_names)
plt.legend()
plt.tight_layout()
plt.show()
```

Continue .. Results and Observation

Conclusion and Future Work

- In conclusion, predictive model has proven effective in identifying critical factors associated with employee retention. Through the analysis of the IBM dataset and the application of feature selection methods, I have successfully identified key features that significantly impact churn. By leveraging these findings, organizations can gain valuable insights and take proactive measures to improve staff retention efforts.

- The implementation of our predictive model offers several benefits to HR practitioners and organizations as a whole. By accurately identifying employees at risk of churn, HR practitioners can focus their efforts on implementing targeted retention strategies. This can include interventions such as personalized development plans, improved work-life balance initiatives, or career advancement opportunities. Ultimately, the predictive model empowers HR practitioners to make data-driven decisions to mitigate churn and retain valuable employees, leading to a more stable and productive workforce.

FUTURE WORK

- For future work, there are several avenues to explore. First, additional feature selection methods could be investigated to further refine the predictive model. Alternative techniques such as Lasso regression, principal component analysis (PCA), or mutual information-based methods may provide complementary insights into the most influential features for churn prediction.

- Furthermore, expanding the dataset to include more diverse and comprehensive employee data can enhance the model's performance and generalizability. This could involve incorporating data from multiple organizations or industries, considering a wider range of demographic factors, or integrating external data sources such as employee sentiment analysis or social media data.

Thank you