Predicting Happiness: A Data-Driven Approach

This presentation explores the relationship between socio-demographic and lifestyle factors and happiness, using a data-driven approach to uncover key insights and understand the factors that contribute to well-being.

Albatool Moathen
Mona Albarqi

Problem Statement & Dataset

Objective

Identify key predictors of happiness and understand the factors that contribute to well-being.

Dataset

This study utilizes a dataset comprising 6,000 rows and 11 columns, analyzing various socio-demographic and lifestyle factors.

Methods & Models

1

Logistic Regression

Serves as a baseline model to establish a basic understanding of the relationship between predictors and happiness.

2

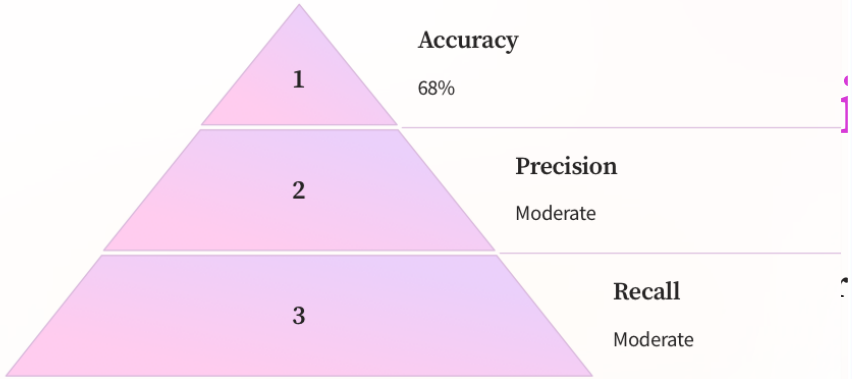
Random Forest

Employed to improve model performance and identify feature importance, as it can handle non-linear relationships between predictors and happiness.

Data Preprocessing

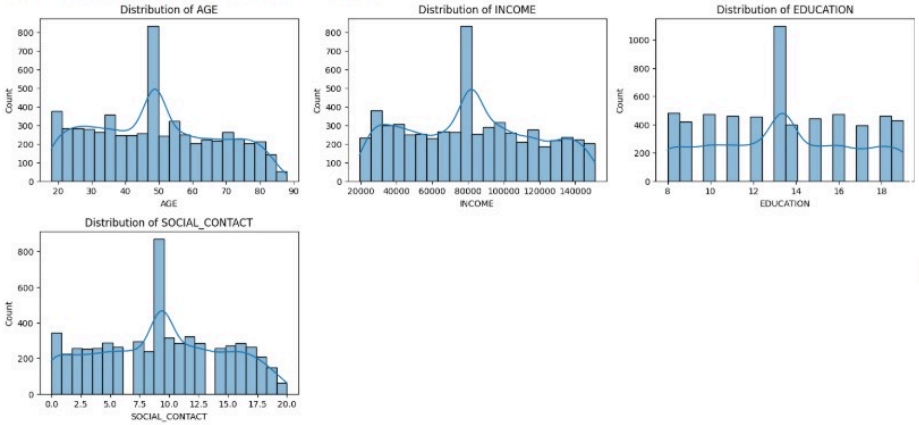
1. Handle missing values using mean imputation for numerical features and mode imputation for categorical features.
2. Encode categorical variables using label encoding, converting categorical values into numerical representations.
3. Apply StandardScaler to scale numerical features to a common range, improving the performance and convergence of machine learning models.
4. Split the dataset into 80% for training and 20% for testing, ensuring a robust evaluation of model performance.

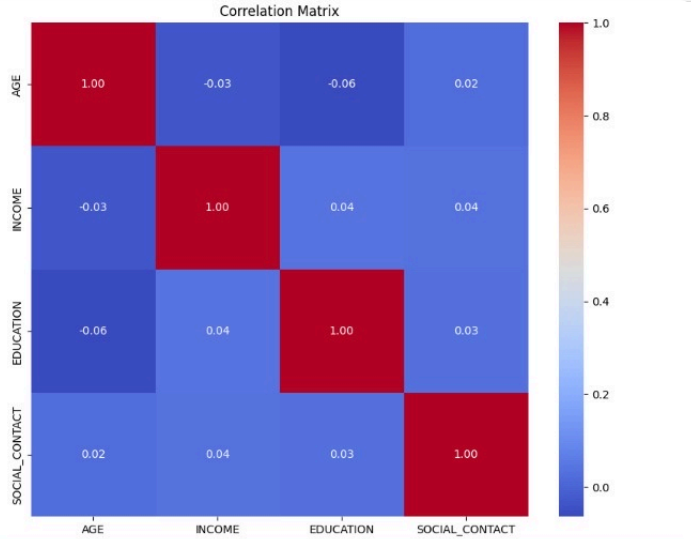
Model Results - Logistic Regression




```
[15]: # Exploratory Data Analysis (EDA)

import seaborn as sns
import matplotlib.pyplot as plt

# Summary statistics of numeric variables
print("\nSummary Statistics for Numeric Variables:")
print(data.describe())

# Visualizing the distribution of numeric variables
plt.figure(figsize=(15, 10))
for i, col in enumerate(numeric_cols, 1):
    plt.subplot(3, 3, i)
    sns.histplot(data[col], kde=True)
    plt.title(f'Distribution of {col}')
plt.tight_layout()
plt.show()

# Correlation heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(data[numeric_cols].corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix')
plt.show()

# Visualizing the relationship between the target variable and key predictors
for col in categorical_cols:
    plt.figure(figsize=(8, 6))
    sns.countplot(data=data, x=col, hue='HAPPINESS')
    plt.title(f'HAPPINESS by {col}')
    plt.xticks(rotation=45)
    plt.legend(title='Happiness Level')
    plt.show()
```

```
Summary Statistics for Numeric Variables:
                AGE         INCOME    EDUCATION  SOCIAL_CONTACT
count  6000.000000    6000.000000  6000.000000     6000.000000
mean     48.848889   81566.800185    13.423519        9.326296
std      18.390355   35664.984255     3.273101        5.380118
min      18.000000   19315.000000     8.000000        0.000000
25%      34.000000   51080.500000    11.000000        5.000000
50%      48.848889   81566.800185    13.423519        9.326296
75%      63.000000  108784.250000    16.000000       13.000000
max      88.000000  150926.000000    19.000000       20.000000
```





Model Results - Random Forest

1

Accuracy

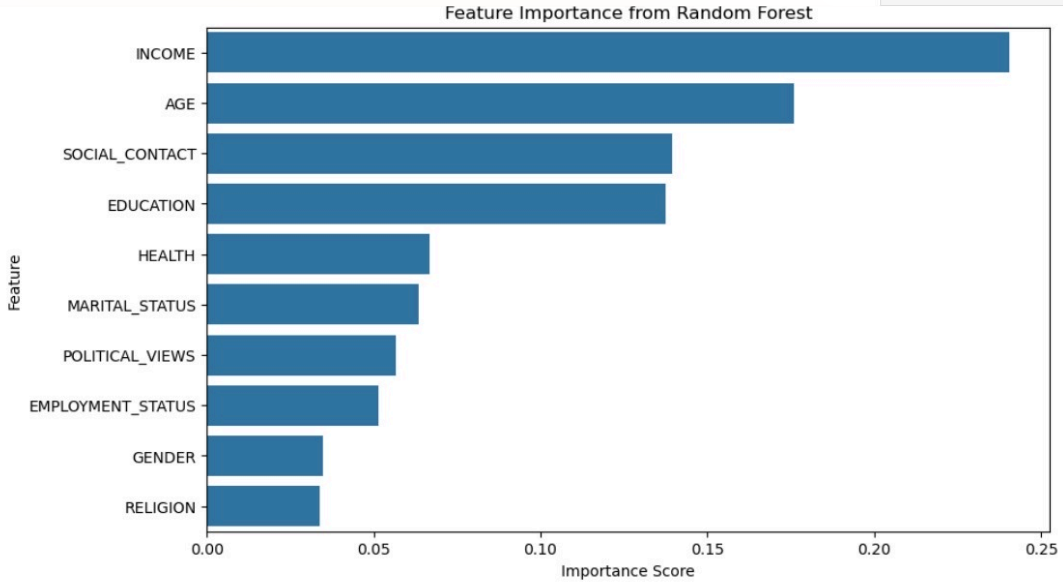
74%

2

Feature Importance

Income, Health, Social Contac

[25]:
```
# Ensure all features are numeric before training Random Forest
from sklearn.preprocessing import LabelEncoder

non_numeric_features = X.select_dtypes(include=['object']).columns
for col in non_numeric_features:
    le = LabelEncoder()
    X[col] = le.fit_transform(X[col])

# Train the Random Forest Model again if necessary
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Get Feature Importances
importances = rf_model.feature_importances_
features = X.columns

# Create a DataFrame for visualization
importance_df = pd.DataFrame({'Feature': features, 'Importance': importances})
importance_df = importance_df.sort_values(by='Importance', ascending=False)

# Plot the Feature Importances
plt.figure(figsize=(10, 6))
sns.barplot(x='Importance', y='Feature', data=importance_df)
plt.title('Feature Importance from Random Forest')
plt.xlabel('Importance Score')
plt.ylabel('Feature')
plt.show()
```



Feature Importance

1

Income

Financial stability plays a significant role in happiness.

2

Health

Physical and mental well-being contribute to happiness.

3

Social Contact

Strong social connections enhance happiness.

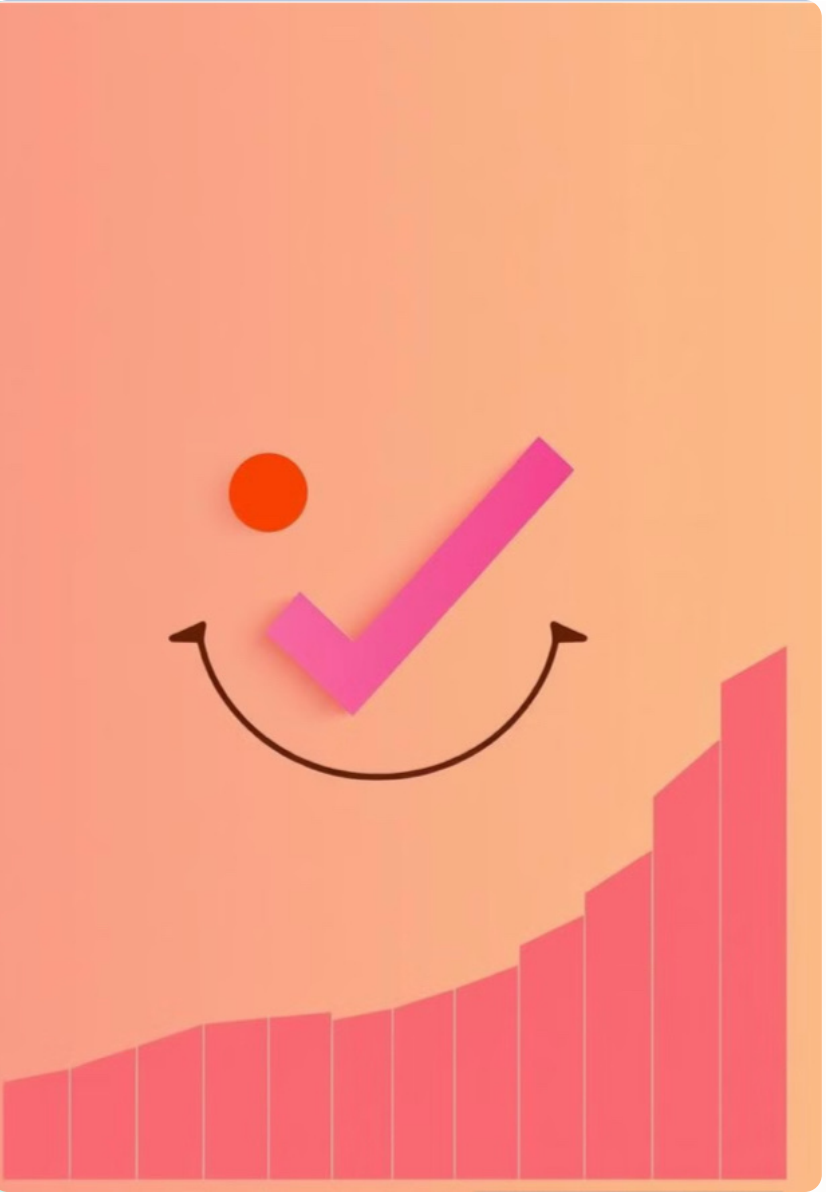
Challenges & Lessons Learned

Challenges

Limitations of synthetic datasets and potential bias introduced by missing data imputation are significant considerations.

Lessons

Random Forest effectively handles class imbalances and aligns well with real-world expectations.



Conclusion

Summary

Random Forest emerges as the superior model with an accuracy of 74%, identifying income, health, and social contact as key predictors of happiness.

Future Steps

The study will be further enhanced by utilizing real-world data for greater generalization and exploring additional predictors that might contribute to happiness.

Thank You

We appreciate your attention and welcome any questions or discussions. Feel free to connect with us for further inquiries.