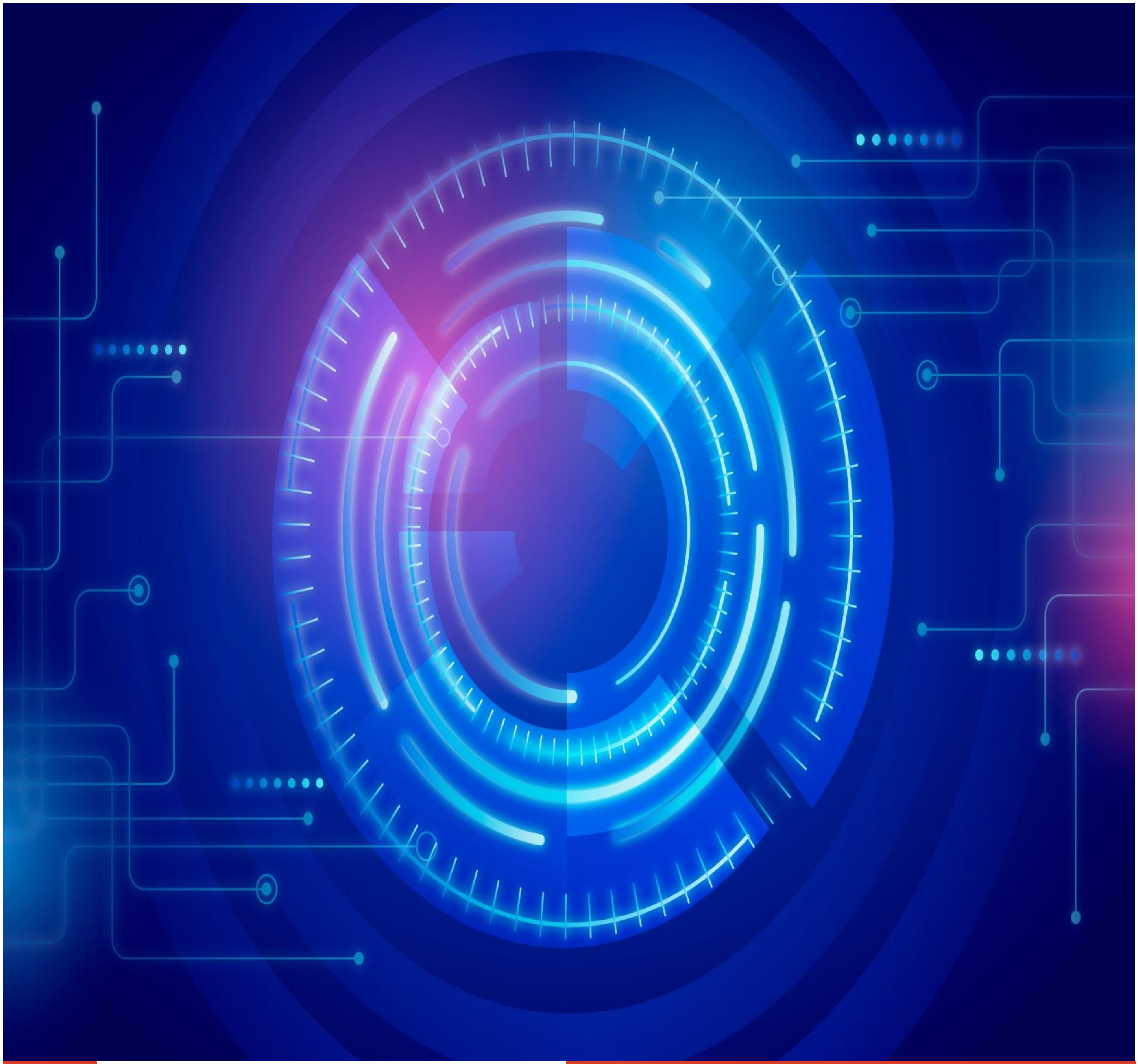
TOUR & TRAVEL CUSTOMER CHURN PREDICTION

PROJECT REPORT

COGNORISE INFOTECH

Authored by: Mahmoud Elmahdy



1. PROJECT OVERVIEW

Objective

The primary goal of this project was to develop a predictive model to identify potential customer churn in a travel company. By understanding and predicting which customers are likely to leave, the company can take proactive measures to retain them and optimize resource allocation.

2. DATASET CHARACTERISTICS

Data Description

```
# Assess the first 5 rows to explore the data
df.head()

   Age FrequentFlyer AnnualIncomeClass  ServicesOpted  \
0   34            No     Middle Income              6
1   34           Yes        Low Income              5
2   37            No     Middle Income              3
3   30            No     Middle Income              2
4   30            No        Low Income              1

  AccountSyncedToSocialMedia BookedHotelOrNot  Target
0                         No              Yes       0
1                        Yes               No       1
2                        Yes               No       0
3                         No               No       0
4                         No               No       0
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 954 entries, 0 to 953
Data columns (total 7 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Age                         954 non-null    int64
 1   FrequentFlyer               954 non-null    object
 2   AnnualIncomeClass           954 non-null    object
 3   ServicesOpted               954 non-null    int64
 4   AccountSyncedToSocialMedia  954 non-null    object
 5   BookedHotelOrNot            954 non-null    object
 6   Target                      954 non-null    int64
dtypes: int64(3), object(4)
memory usage: 52.3+ KB
```

The dataset contains 954 customer records with the following features:

- Age
- Frequent Flyer Status
- Annual Income Class
- Number of Services Opted
- Social Media Account Sync Status
- Hotel Booking Status
- Churn Target (Binary: 0 = No Churn, 1 = Churn)

Key Statistical Insights

```
# Check the summary statistics of the data
df.describe().T

                count       mean       std   min   25%   50%   75%
max
Age             954.0  32.109015  3.337388  27.0  30.0  31.0  35.0
38.0
ServicesOpted   954.0   2.437107  1.606233   1.0   1.0   2.0   4.0
6.0
Target          954.0   0.234801  0.424097   0.0   0.0   0.0   0.0
1.0
```

- Average Customer Age: 32 years (Range: 27-38)
- Churn Rate: 23% (224 out of 954 customers)

Feature Distribution

1. **Frequent Flyer**

```
FrequentFlyer
No           608
Yes          286
No Record     60
Name: count, dtype: int64
```

   - No: 608 customers
   - Yes: 286 customers
   - No Record: 60 customers

2. **Annual Income Class**

```
AnnualIncomeClass
Middle Income    409
Low Income       386
High Income      159
Name: count, dtype: int64
```

   - Middle Income: 409 customers
   - Low Income: 386 customers
   - High Income: 159 customers

3. **Services Opted**

```
ServicesOpted
1    404
2    176
3    124
4    117
5     69
6     64
Name: count, dtype: int64
```

- Range: 1-6 services
- Most customers opt for 1-2 services

3. DATA PREPROCESSING

Preprocessing Steps

1. **Scaling**:

```
numerical = ['Age', 'ServicesOpted']
```

```
scaler.fit_transform(features_raw[numerical])

# Show an example of a record with scaling applied
display(features_minmax_transform.head(n = 5))
```

```
        Age FrequentFlyer AnnualIncomeClass  ServicesOpted  \
0  0.636364            No     Middle Income            1.0
1  0.636364           Yes        Low Income            0.8
2  0.909091            No     Middle Income            0.4
3  0.272727            No     Middle Income            0.2
4  0.272727            No        Low Income            0.0

  AccountSyncedToSocialMedia BookedHotelOrNot
0                         No              Yes
1                        Yes               No
2                        Yes               No
3                         No               No
4                         No               No
```

- Used MinMaxScaler to normalize numerical features (Age, Services Opted)
- Scaled features to range [0, 1]

2. **Encoding**:

```
# One-hot encode the 'features_minmax_transform' data using
pandas.get_dummies()
features_final = pd.get_dummies(features_minmax_transform)

# Print the number of features after one-hot encoding
encoded = list(features_final.columns)
print("{} total features after one-hot
encoding.".format(len(encoded)))

12 total features after one-hot encoding.
```

- Applied one-hot encoding to categorical variables
- Transformed dataset from 7 to 12 features

3. **Data Split**

```
# Import train_test_split
from sklearn.model_selection import train_test_split

# Split the 'features' and 'income' data into training and testing
sets
X_train, X_test, y_train, y_test = train_test_split(features_final,
                                                    churn_raw,
                                                    test_size = 0.2,
                                                    random_state = 0)

# Show the results of the split
print("Training set has {} samples.".format(X_train.shape[0]))
print("Testing set has {} samples.".format(X_test.shape[0]))

Training set has 763 samples.
Testing set has 191 samples.
```

- Training Set: 763 samples (80%)
- Testing Set: 191 samples (20%)

4. MODEL DEVELOPMENT

Baseline (Naive Predictor)

```
TP = np.sum(churn_raw)
FP = churn_raw.count()
TN = 0
FN = 0
# Calculate accuracy, precision and recall
accuracy = (TP + TN )/ (TP + TN + FP + FN)
recall = TP / (TP + FN)
precision = TP / (TP + FP)

# Calculate F-score using the formula below for beta = 0.5 and correct
values for precision and recall.
beta = 0.5
fscore = (1+beta**2)*((precision*recall)/(((beta**2)*precision)
+recall))

# Print the results
print("Naive Predictor: [Accuracy score: {:.4f}, F-score:
{:.4f}]".format(accuracy, fscore))
```

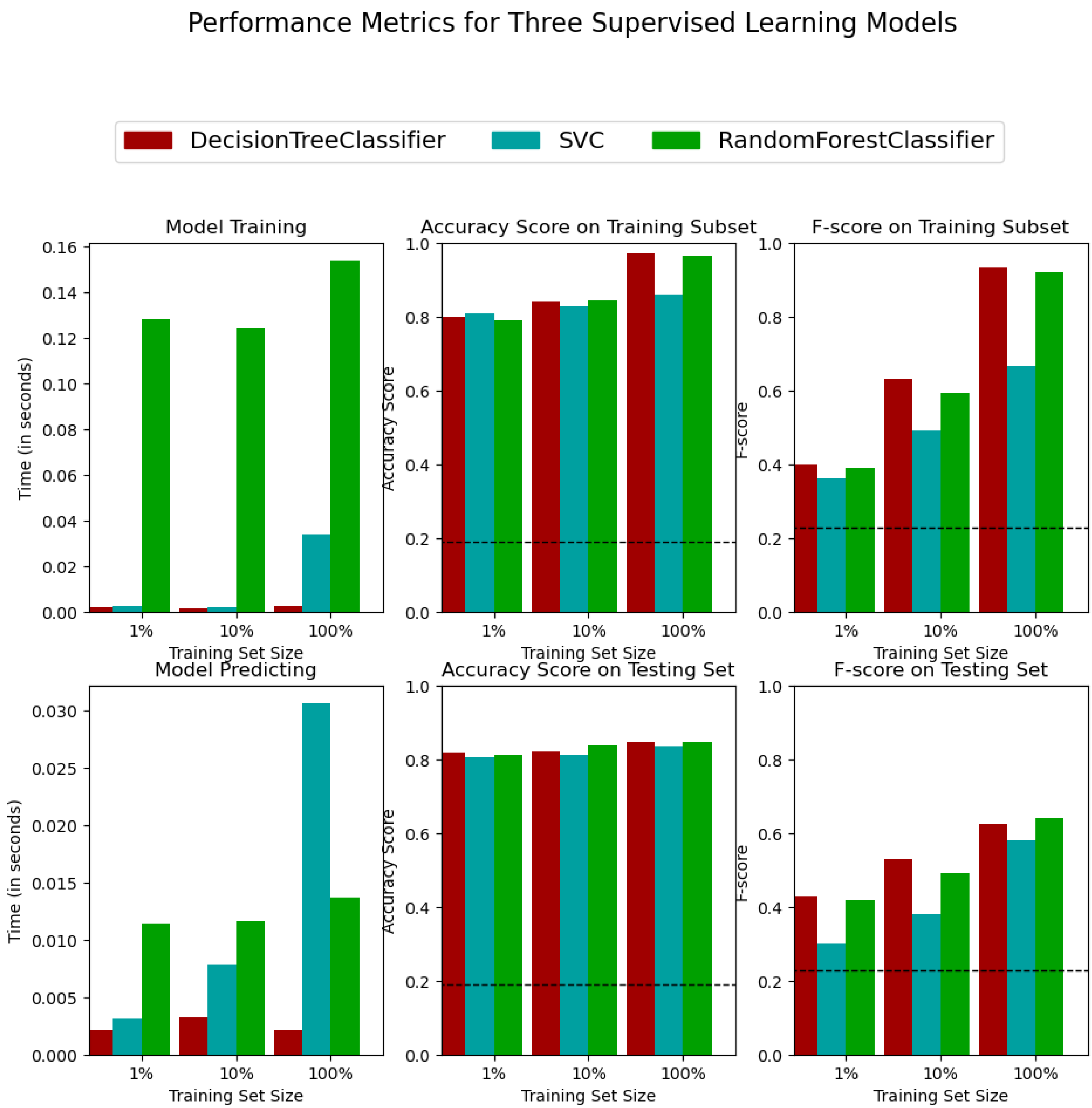
```
Naive Predictor: [Accuracy score: 0.1902, F-score: 0.2269]
```

- Accuracy: 0.1902
- F-score: 0.2269

Machine Learning Models Evaluated

1. Decision Tree Classifier
2. Support Vector Machine (SVM)
3. Random Forest Classifier

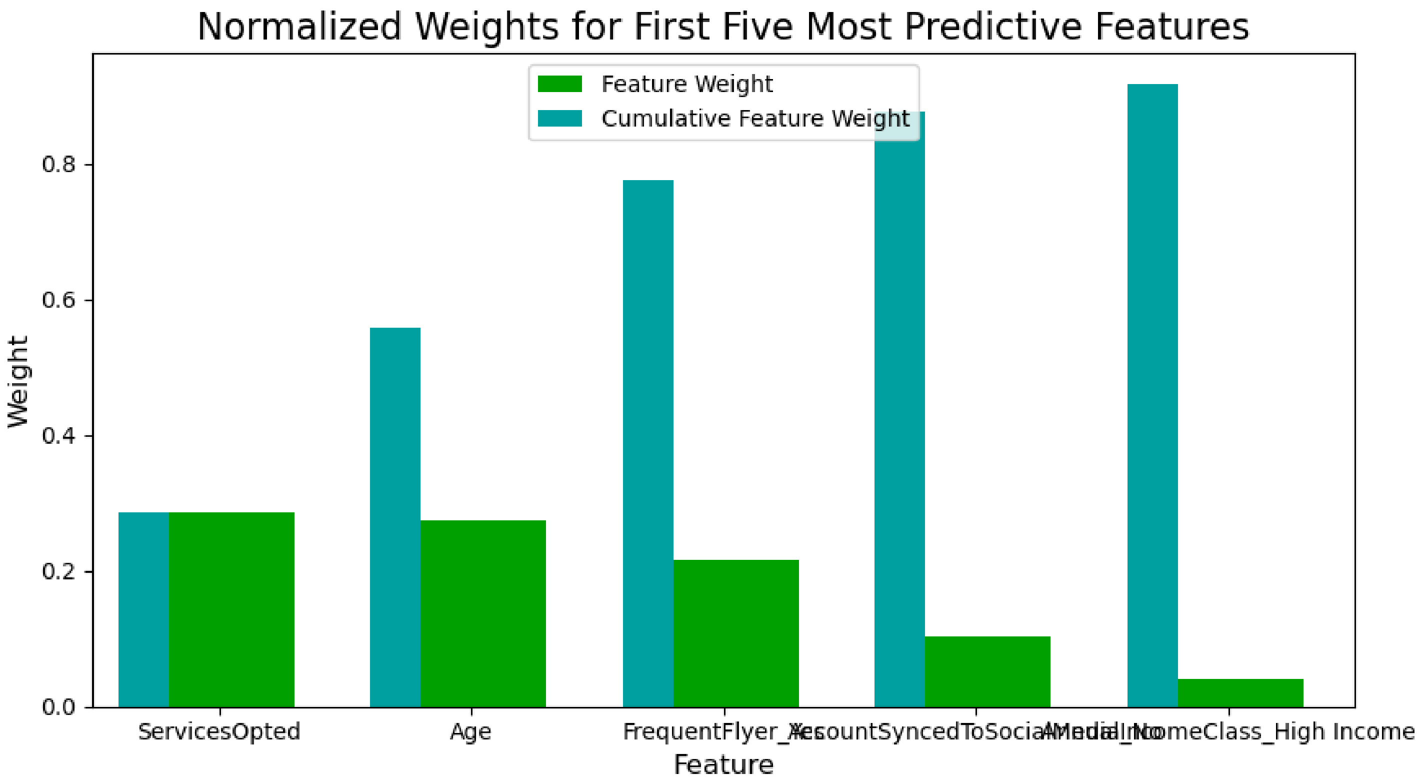
Model Performance Comparison



Decision Tree Classifier

- Best performing model
- Accuracy: 85%
- F-score: 62%

Feature Importance Analysis



The feature importance plot helped identify the most critical features for predicting customer churn.

5. MODEL VALIDATION

```
# Reduce the feature space
X_train_reduced = X_train[["ServicesOpted","Age"]]
X_test_reduced = X_test[["ServicesOpted","Age"]]

# Train on the "best" model found from grid search earlier
clf = (clone(clf_A)).fit(X_train_reduced, y_train)

# Make new predictions
reduced_predictions = clf.predict(X_test_reduced)

# Report scores from the final model using both versions of data
print("Final Model trained on full data\n------")
print("Accuracy on testing data: {:.4f}".format(accuracy_score(y_test,
clf_A.predict(X_test))))
print("F-score on testing data: {:.4f}".format(fbeta_score(y_test,
clf_A.predict(X_test), beta = 0.5)))
print("\nFinal Model trained on reduced data\n------")
print("Accuracy on testing data: {:.4f}".format(accuracy_score(y_test,
reduced_predictions)))
print("F-score on testing data: {:.4f}".format(fbeta_score(y_test,
reduced_predictions, beta = 0.5)))

Final Model trained on full data
------
Accuracy on testing data: 0.8482
F-score on testing data: 0.6593

Final Model trained on reduced data
------
Accuracy on testing data: 0.8010
F-score on testing data: 0.5000
```

Full Feature Model

- Accuracy: 0.8482
- F-score: 0.6593

Reduced Feature Model (Age, Services Opted)

- Accuracy: 0.8010
- F-score: 0.5000

Conclusion: The full feature model outperforms the reduced feature model, indicating that all features contribute significantly to churn prediction.

6. KEY FINDINGS AND RECOMMENDATIONS

Insights

1. The company has a notable churn rate of 23%
2. Most customers are middle-income, aged around 32
3. A majority of customers are not frequent flyers
4. Most customers opt for 1-2 services

Recommendations

1. **Targeted Retention Strategies**
    - Develop personalized retention programs for customers with high churn risk
    - Focus on understanding why customers with fewer services are more likely to churn
2. **Service Enhancement**
    - Investigate reasons behind low service adoption
    - Create bundled service packages to increase customer engagement
3. **Frequent Flyer Program**
    - Enhance frequent flyer benefits to improve customer loyalty
    - Create incentives for non-frequent flyers to increase travel frequency

7. LIMITATIONS AND FUTURE WORK

Limitations

- Relatively small dataset (954 records)
- Binary churn prediction without granular risk levels

Future Improvements

1. Collect more data to improve model accuracy
2. Implement more advanced models like gradient boosting
3. Add more features if possible
4. Develop a probability-based churn risk scoring system

8. CONCLUSION

The machine learning model, particularly the Decision Tree Classifier, provides a robust tool for predicting customer churn in the travel company. With an accuracy of 85% and an F-score of 62%, it offers valuable insights for proactive customer retention strategies.

TECHNICAL ENVIRONMENT

- Programming Language: Python
- Libraries: Pandas, NumPy, Scikit-learn
- Preprocessing: MinMax Scaling, One-Hot Encoding
- Models: Decision Tree, SVM, Random Forest