|  |  |
| --- | --- |
| SEMESTER PROJECT  ITRI 616 - AI | Abstract  This project is focused on modelling data and training models. We will take a look at how to use Kaggle and Jupyter to train models using datasets from Kaggle.  Kristen Hoff  3429292 |

# **Table Of Contents**

|  |  |
| --- | --- |
| Topic | Page Number |
| Introduction | 2 |
| The Description Of The Dataset | 2 |
| Methodology | 2 |
| 3.1 Data Preprocessing | 2 |
| 3.2 Feature Engineering | 3 |
| 3.3 The Model Selection | 3 |
| 3.4 The Model Training | 3 |
| Evaluation Results | 3-4 |
| Discussion | 4 |
| Conclusion And Future Work | 4 |
| Bibliography | 4 |
| Appendix | 4 |
| Screenshots Of Output | 4-12 |

## **Introduction**

Customer churn is an extremely challenging concept for subscription-based businesses such as telecommunications providers as telco itself. By accurately identifying which customers are likely to cancel services, these businesses can now aid in the implementation of retention strategies. The aim of this project is to build a machine learning model that can predict churning by using customer data such as demographics, billing, and service usage. The model is built using publicly available data from the \*\*Telco Customer Churn\*\* dataset on Kaggle, and is evaluated using standard classification performance metrics.

## **2. The Description of The Dataset**

The dataset used in this project is from the Telco Customer Churn dataset from Kaggle, consisting of 7,043 entries. It also includes 20 features and a target variable or “Churn”, which indicates whether a customer has canceled their services or not. It also includes certain features such as numerical values for (e.g., tenure, MonthlyCharges), binary (e.g., gender, SeniorCitizen), and multi-class categorical variables (e.g., InternetService, PaymentMethod).

**The link to the dataset is inserted below:**

<https://www.kaggle.com/datasets/blastchar/telco-customer-churn>

## **3.Methodology**

## **3.1 Data Preprocessing**

The Data cleaning process involved converting the ‘TotalCharges’ column to numeric format and removing rows with missing values. The ‘customerID’ column was also removed as it holds no predictive value. Binary variables that are categorical were labelled and encoded, and multi-class variables were one-hot encoded using pandas” and ‘get\_dummies()’.

## **3.2 Feature Engineering**

For this section there were no additional features that were engineered. Only the existing features were encoded to make them suitable or more appropriate for machine learning algorithms to be used. Numerous categorical features were transformed into dummy variables to allow ML algorithms to handle them effectively.

## **3.3 The Model Selection**

Regarding the model selection process I have ensured that three models were selected and are as follows:

* Decision Tree
* Random Forest and
* Support Vector Machine (SVM).

I chose these because of their proven performance in binary classification and their complementary strengths in accuracy and interpretability.

## **3.4 The Model Training**

When training the dataset, it was split into training and testing sets using the ratio of 80/20. Default parameters using Scikit-learn was used to train the data models. The target variable or ‘Churn’ was encoded as 0 (No) and 1 (Yes).

## **4. Evaluation and Results**

The performance of the models was assessed using Accuracy, Precision, Recall, F1 Score, and ROC-AUC. The Random Forest model outperformed the others as seen below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1 Score** | **ROC AUC** |
| **Decision Tree** | 0.78 | 0.68 | 0.62 | 0.65 | 0.75 |
| **Random Forest** | 0.82 | 0.74 | 0.67 | 0.70 | 0.85 |
| **SVM** | 0.79 | 0.69 | 0.63 | 0.66 | 0.83 |

## **5. Discussion**

Random Forest provided the best overall performance in all evaluated metrics. Although SVM was slightly more accurate than the Decision Tree, it is less interpretable and computationally heavier. Some limitations include class imbalance and a lack of hyperparameter tuning.

## **6. Conclusion and Future Work**

The project should successfully demonstrate machine learning techniques for churn prediction. Random Forest emerged as the most effective model compared to SVM and the Decision Trees. Future work could involve applying ensemble techniques like XGBoost, and feature importance analysis to understand key drivers of churn better and improve its overall efficiency.

# **7. Bibliography**

(Blastchar, Telco Customer Churn, n.d.)

## **8. Appendix**

GitHub Repository: <https://github.com/Kristen785/ITRI_616_Project>

See the Jupyter Notebook for code, plots (e.g., ROC curves, Confusion Matrix), and read the requirements.txt on additional implementation details and steps to follow.

## **9.Screenshots of Output from using Jupyter in Kaggle**

