IT Service Tickets Binary Classification Using Data Mining Techniques  
by  
Sriniketh Brahmandam (sb2582)

Introduction:

This document describes the working of 3-different ML algorithms in classifying IT service tickets based on text or description about the ticket. It is a binary classification problem and I have trained and evaluated Random Forest, SVM and LSTM on the dataset.

Objective :

The objective of this project is to train and evaluate 3 - different ML algorithms using k-fold technique . This helps in understanding various evaluation metrics, evaluation techniques in data mining.

Tools and Libraries :

* Python = 3.9
* Scikit-learn
* Pandas
* NLTK
* Tensorflow

Repository:

Github : <https://github.com/SrinikethBrahmandam/CS_634_Final_Project>

About Data :

The dataset has total 47,837 rows and 2 columns with following categories **'Hardware', 'HR Support', 'Access', 'Miscellaneous', 'Storage', 'Purchase', 'Internal Project', 'Administrative rights'.**  For this assignment I have selected two categories of data **'Purchase':0,"Internal Project":1.**

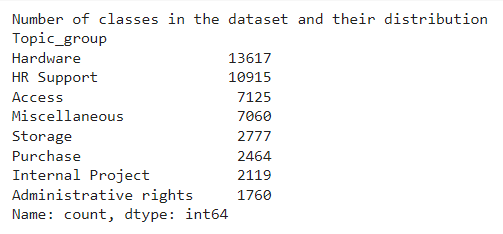


Fig 1: Data Distribution

Records count between **Purchase and Internal Project**  classes :

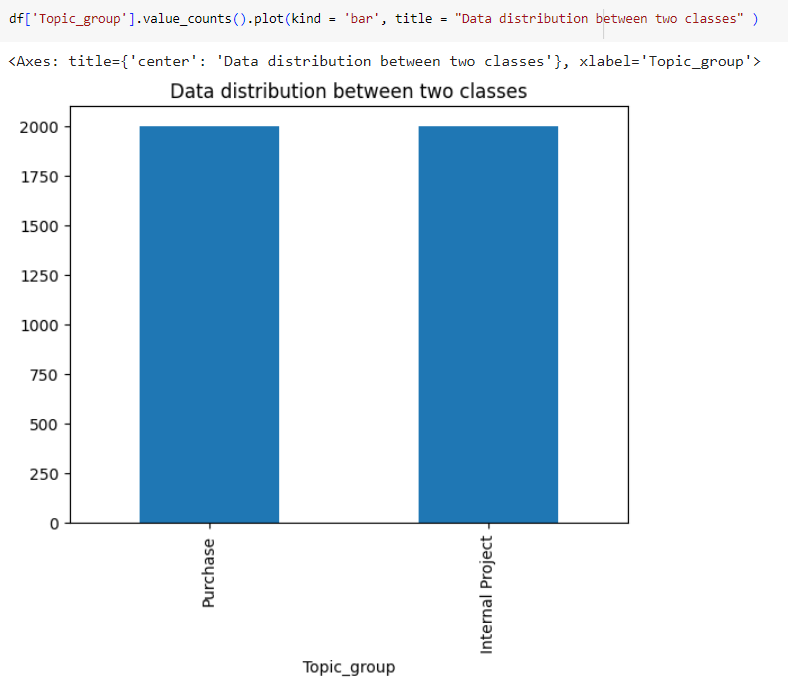


Fig 2: Data distribution between ‘purchase’ and ‘internal Project’ classes

Data Pre-processing :

Before training the data need to be cleaned , so I have employed following data cleaning techniques :

1. converted the text to lowercases
2. removed digits and numbers
3. Removed punctuations
4. Tokenized the text using word tokenizer
5. Removed stopword if any

Data Pre-preparation :

The text data need to be represented numerically which captures the relations between the words to train ML models. So I have used **TF-IDF** text vectorization technique to create text embeddings.

The data is split into train and test sets, the models are only trained on train set and evaluated on test set.

Models :

**Random forest** is a supervised machine learning algorithm which can be used for both classification and regression tasks. This algorithm creates a "forest" of decision trees, where data from multiple trees are merged to ensure accurate predictions. By selecting random subsets of features, random forest introduces diversity into the model, leading to more precise predictions. It is known for its ease of use, ability to handle both classification and regression problems, and its effectiveness in reducing overfitting compared to individual decision trees.

**SVM :**

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression tasks. SVMs work by finding an optimal hyperplane in an N-dimensional space to separate data points of different classes. This hyperplane aims to maximize the margin between the closest points of different classes, ensuring effective classification. SVMs can efficiently handle high-dimensional data and nonlinear problems, making itversatile and powerful tools in machine learning.

**LSTM :**

LSTM (Long Short-Term Memory) is a type of Recurrent Neural Network (RNN) that specializes in handling sequential data and learning long-term dependencies between time steps. It uses memory cells and gates to control the flow of information over time, allowing it to effectively learn and model complex sequential patterns. LSTM networks are often used for classification tasks, where they input sequence or time series data, learn long-term dependencies, map the output to a fixed-size vector, and output a probability distribution over the possible classes. LSTM is a powerful and versatile tool for working with sequential data, particularly in tasks involving long-term dependencies and classification.

Model Training and Evaluation :

The model is trained and evaluated using K-fold cross-validation technique. With K value as 10.

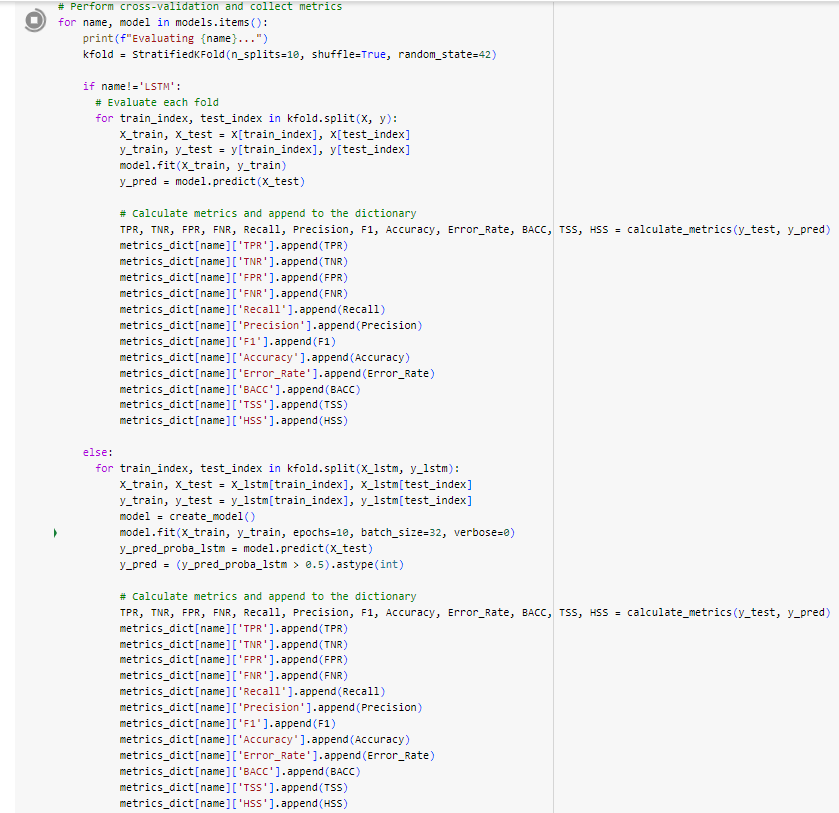


Fig 3 : K-fold cross validation code snippet

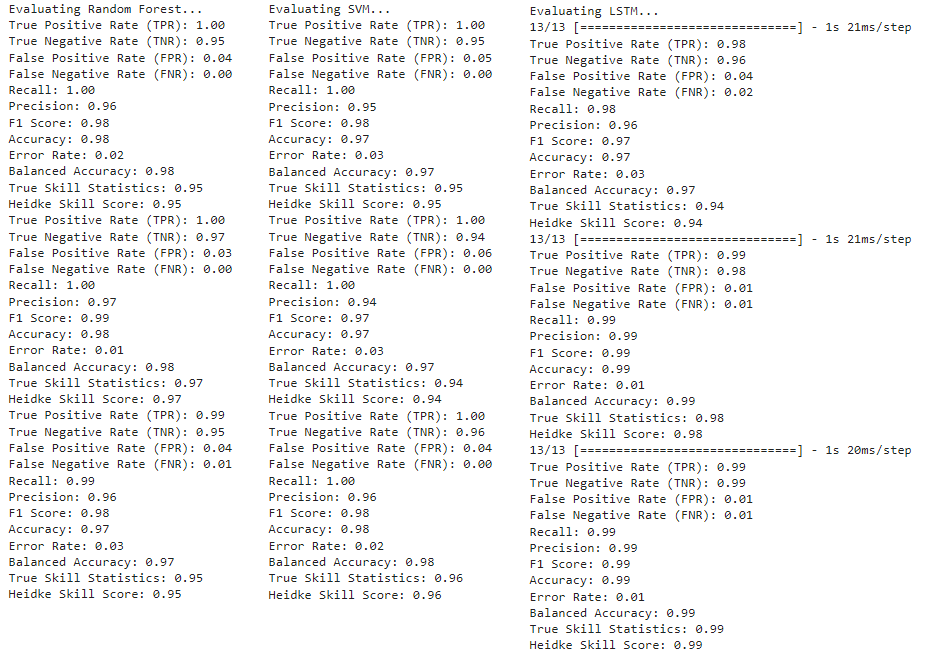


Fig 4 : Training

Evaluation metrics :

Models are evaluation using following metrics:

**True Positive Rate (TPR):** It is the proportion of true positive predictions among all actual positive instances. It indicates the model's ability to correctly identify positive cases. Higher TPR signifies better performance in correctly predicting positive instances.

**True Negative Rate (TNR):** It is the proportion of true negative predictions among all actual negative instances. It indicates the model's ability to correctly identify negative cases. Higher TNR signifies better performance in correctly predicting negative instances.

**False Positive Rate (FPR):** It is the proportion of false positive predictions among all actual negative instances. It indicates the model's tendency to incorrectly label negative cases as positive. Lower FPR indicates better performance, as it means fewer false alarms.

**False Negative Rate (FNR):** It is the proportion of false negative predictions among all actual positive instances. It indicates the model's tendency to incorrectly label positive cases as negative. Lower FNR indicates better performance, as it means fewer missed positive cases.

**Recall:** The proportion of true positive predictions among all actual positive instances. It measures the model's ability to capture positive instances. Higher recall indicates better performance in capturing positive cases.

**Precision:** It is the proportion of true positive predictions among all predicted positive instances. It measures the accuracy of positive predictions. Higher precision indicates better performance in making precise positive predictions.

**F1 Score:** It is the harmonic mean of precision and recall. It balances between precision and recall and is useful when classes are imbalanced. Higher F1 score indicates better overall performance in terms of both precision and recall.

**Accuracy:** It is the proportion of correct predictions among all predictions made by the model. It measures the overall correctness of the model's predictions. Higher accuracy indicates better overall performance.

Error Rate: It is the proportion of incorrect predictions among all predictions made by the model. It measures the overall incorrectness of the model's predictions. Lower error rate indicates better performance.

**Balanced Accuracy (BACC):** It is average of TPR and TNR. It provides a balanced measure of model performance across both positive and negative classes. Higher BACC indicates better overall performance in terms of correctly predicting both positive and negative instances.

**True Skill Statistics (TSS):** It combines TPR and TNR to assess model performance. It considers both sensitivity and specificity. Higher TSS indicates better discrimination ability of the model.

**Heidke Skill Score (HSS):** It is used to measure the model's skill in predicting the observed frequency of events compared to random chance. Higher HSS indicates better agreement between predicted and observed frequencies, beyond what would be expected by random chance.

**Results :**

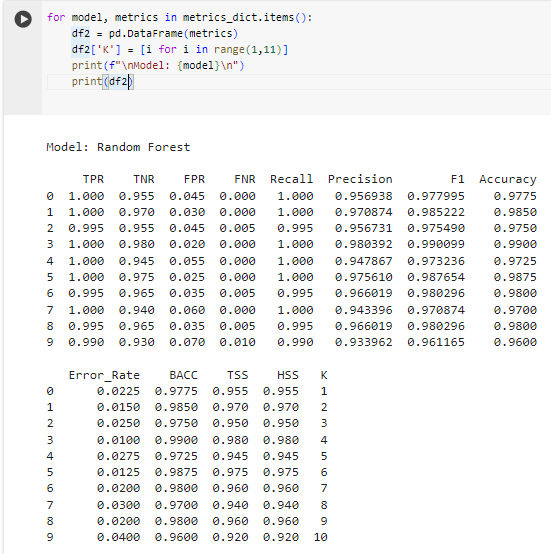


Fig 5 : Random Forest

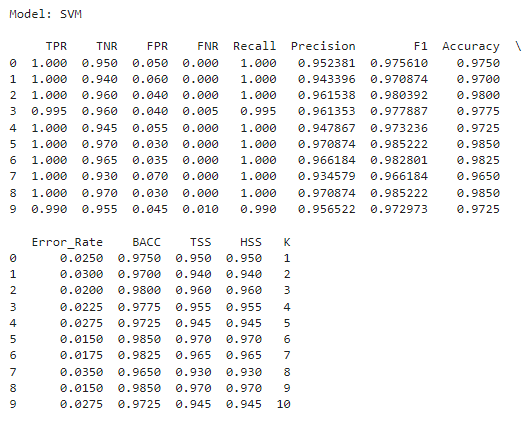


Fig 7 : SVM

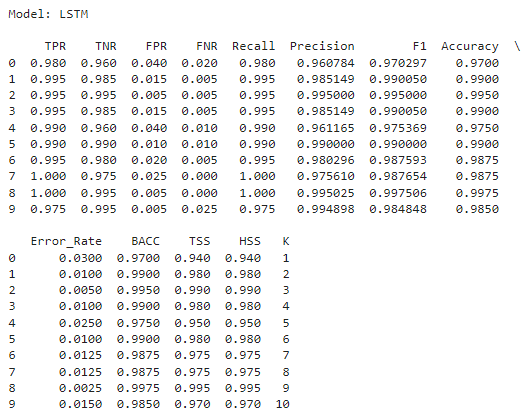


Fig 8 : LSTM

Steps to Run the code :

1. Clone the git repository using the command:

git clone <https://github.com/SrinikethBrahmandam/CS_634_Final_Project.git>

1. Open the jupyter notebook
2. Run all cells

Conclusion:

* Both Random forest and SVM had shown similar results. Their True Positive Rate (TPR), True Negative Rate (TNR), Precision, F1 Score, Accuracy, Error Rate, Balanced Accuracy, True Skill Statistics, and Heidke Skill Score. Their TPR, TNR, Recall, Precision, F1 Score, Accuracy, Balanced Accuracy, True Skill Statistics, and Heidke Skill Score values are consistently high, indicating robust performance.
* False Positive Rate (FPR): Both Random Forest and SVM demonstrate a low FPR, indicates their ability to minimize the misclassification of negative instances as positive.
* LSTM Performance: The LSTM model also performed well, with high TPR, TNR, Recall, Precision, F1 Score, Accuracy, Balanced Accuracy, True Skill Statistics, and Heidke Skill Score values. However, it had shown slightly higher FPR compared to Random Forest and SVM.

All three models—Random Forest, SVM, and LSTM—demonstrated strong performance in terms of classification accuracy and predictive capability. The Random Forest and SVM models exhibit similar performance levels, while the LSTM model also performed well but with a slight higher false positive rate. All three modes perfromed very well on the task.