Name: Vaibhav Bora

NJIT ucid: vb22

Email Address: vb22@njit.edu

24 November 2024

Professor: Yasser Abduallah

CS 634-101 Data Mining

Final Project Report

Project Technical Overview

For this project, I got my dataset from one of the sources the professor provided which helped me to minimize my time needed to find a good dataset. Also, in the information of the source it mentioned that the datasets had no missing values which helped me to not have to make another algorithm to deal with that problem. The dataset is about direct marketing campaigns from a Portuguese banking institution which were based on different phone calls(sometimes more than one contact for the same person was needed and the reason was to know if the product(i.e. Bank term deposit) was subscribed or not.

Dataset Creation and Management

There were four different datasets which were given from the source but I picked the ‘bank.csv’ dataset because it was randomly selected with a few examples and not all. Moreover, it helped me calculate more computationally demanding algorithms, like SVM, more smoothly. The goal of the experiment was to predict if the clients were subscribed to the product or not using the variable y.

Development and Implementation Tools

The main and primary programming and editing I performed occurred in the Jupyter Notebook, which is a powerful tool that actively supports interactive data science and scientific computing. Jupyter Notebook provided an interactive approach to coding and allowed for immediate feedback and continuous refinement of the python code. For the final verification, I ran the code in IDLE, Python's integrated development and learning environment which is a very important step for ensuring the reliability of my code.

Key Technologies Used

Jupyter Notebook: Served as my main platform for developing, testing, and visualizing the algorithm.

Microsoft Excel: Generated and downloaded all the datasets and CSV files from here.

Python IDLE: Used in the final stages of the project to verify the completeness of the code.

ChatGPT: Used for helping understanding the python libraries and utilized for the verification.

Conclusion

After completing this project, it showed me the very useful applications of different algorithms, mainly Random Forest Classifier, SVM and LSTM. Not only that, learning how to implement them in real life examples and using a real technical dataset to experiment with gave me valuable knowledge and experience.

Screenshots

age;"job";"marital";"education";"default";"balance";"housing";"loan";"contact";"day";"month";"duration";"campaign";"pdays";"previous";"poutcome";"y"
30;"unemployed";"married";"primary";"no";1787;"no";"no";"cellular";19;"oct";79;1;-1;0;"unknown";"no"
33;"services";"married";"secondary";"no";4789;"yes";"yes";"cellular";11;"may";220;1;339;4;"failure";"no"
35;"management";"single";"tertiary";"no";1350;"yes";"no";"cellular";16;"apr";185;1;330;1;"failure";"no"
30;"management";"married";"tertiary";"no";1476;"yes";"yes";"unknown";3;"jun";199;4;-1;0;"unknown";"no"
59;"blue-collar";"married";"secondary";"no";0;"yes";"no";"unknown";5;"may";226;1;-1;0;"unknown";"no"
35;"management";"single";"tertiary";"no";747;"no";"no";"cellular";23;"feb";141;2;176;3;"failure";"no"
36;"self-employed";"married";"tertiary";"no";307;"yes";"no";"cellular";14;"may";341;1;330;2;"other";"no"
39;"technician";"married";"secondary";"no";147;"yes";"no";"cellular";6;"may";151;2;-1;0;"unknown";"no"
41;"entrepreneur";"married";"tertiary";"no";221;"yes";"no";"unknown";14;"may";57;2;-1;0;"unknown";"no"

Figure 1: Bank.csv

```
# Importing all the necessary libraries for the code to run smoothly
import pandas as pd
import numpy as np
from sklearn.model_selection import StratifiedKFold
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.metrics import confusion_matrix, roc_auc_score, brier_score_loss, f1_score
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout, Input
from tensorflow.keras.utils import to_categorical
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix, f1_score
import time
```

Figure 2: All the libraries necessary to run the program

In order to install these libraries such as pandas or numpy, just type ‘pip install pandas’ or ‘pip install numpy’ respectively in your command prompt.

```
# Defining the LSTM model using input layer alongside input shape
def create_lstm_model(shape_input):
    model = Sequential()
    model.add(Input(shape=shape_input)) # Defining the shape
    model.add(LSTM(50, activation='relu', return_sequences=True)) # Setting return sequence to True
    model.add(Dropout(0.2)) # Drouput helps out with regularization
    model.add(LSTM(20, activation='relu')) # Another layer
    model.add(Dense(1, activation='sigmoid')) # Helping with the binary classification
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy']) # Compilation
    return model
```

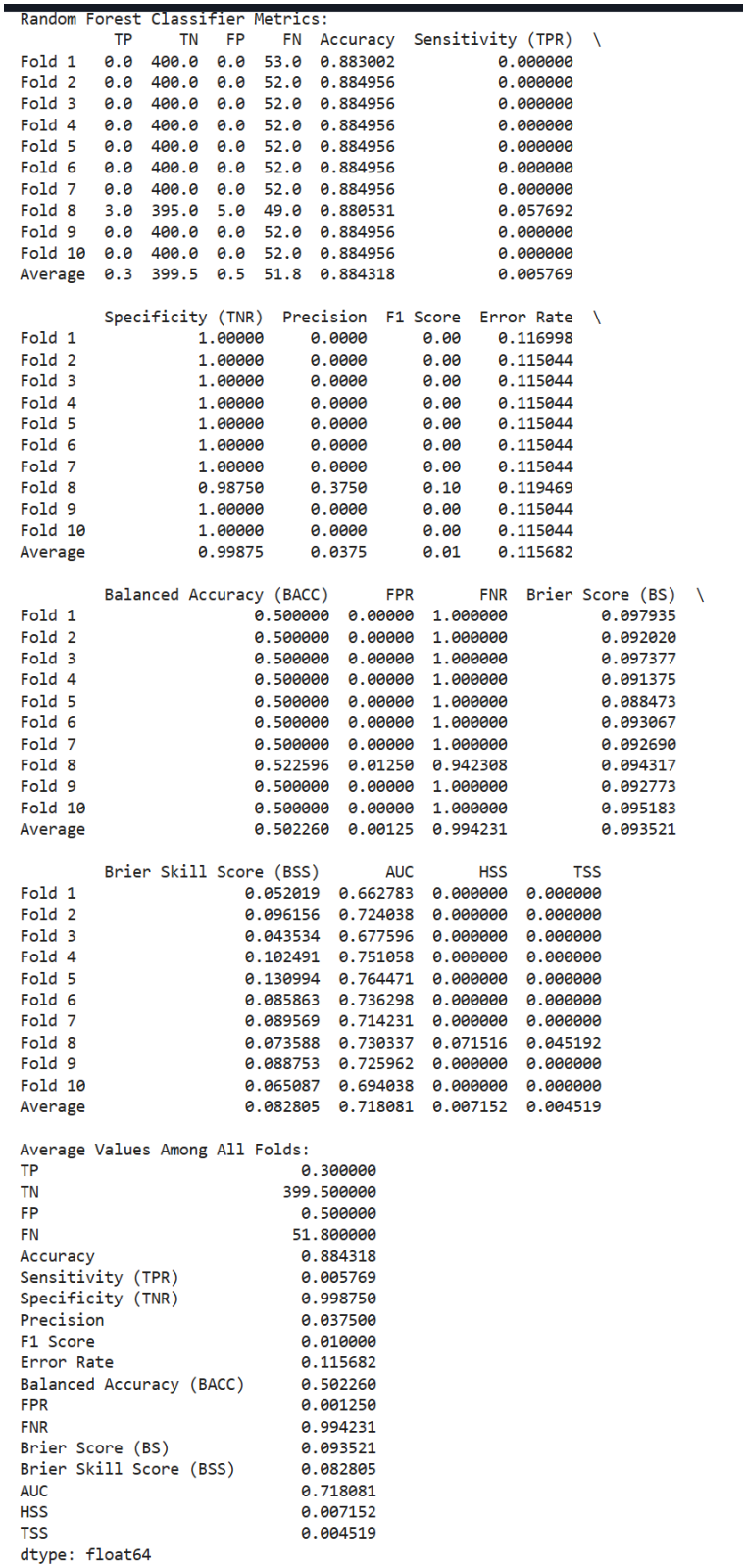
Figure 3: Creating a model for LSTM

```
# Cross-validation loop for each of the three algorithms
for train_index, test_index in kf.split(X_scaled, y):
    X_train, X_test = X_scaled[train_index], X_scaled[test_index] # Splitting into training and testing sets
    X_train_reshaped, X_test_reshaped = X_reshaped[train_index], X_reshaped[test_index] # LSTM exclusive
    y_train, y_test = y[train_index], y[test_index]

    # Random Forest Classifier training and prediction
    rf = RandomForestClassifier(n_estimators=100, random_state=42)
    rf.fit(X_train, y_train)
    y_pred_rf = rf.predict(X_test)
    y_pred_rf_proba = rf.predict_proba(X_test)[:, 1]

    # SVM Classifier training and prediction
    svm = SVC(kernel='rbf', class_weight='balanced', probability=True, random_state=42)
    svm.fit(X_train, y_train)
    y_pred_svm = svm.predict(X_test)
    y_pred_svm_proba = svm.predict_proba(X_test)[:, 1]

    # LSTM Classifier training and prediction
    lstm_model = create_lstm_model((X_train_reshaped.shape[1], X_train_reshaped.shape[2]))
    lstm_model.fit(X_train_reshaped, y_train, epochs=10, batch_size=64, verbose=0)
    y_pred_lstm = (lstm_model.predict(X_test_reshaped) > 0.5).astype(int)
    y_pred_lstm_proba = lstm_model.predict(X_test_reshaped).flatten()
```

Figure 4: Training and prediction for the three algorithms

```
# Jotting down all the important formulas needed for the output calculations for Random Forest
Accuracy = (TP+TN)/(TP+TN+FP+FN)
TPR = TP/(TP+FN)  # Sensitivity
TNR = TN/(TN+FP)  # Specificity
Precision = TP/(TP+FP)
F1_Score = 2*(Precision*TPR)/(Precision+TPR)
Error_Rate = (FP+FN)/(TP+TN+FP+FN)
BACC = (TPR+TNR)/2
FPR = FP/(FP+TN)
FNR = FN/(TP+FN)
TSS = TPR-FPR
BS = brier_score_loss(y_test, y_pred_rf_proba)
BSS = 1-BS/np.var(y_test)
AUC = roc_auc_score(y_test, y_pred_rf_proba)
HSS = 2*(TP*TN-FP*FN)/((TP+FN)*(FN+TN)+(TP+FP)*(FP+TN))
```

Figure 5: Random Forest Classifer's Metric Formulas



```
Random Forest Classifier Metrics:
          TP     TN   FP    FN  Accuracy  Sensitivity (TPR)  \
Fold 1   0.0  400.0  0.0  53.0  0.883002           0.000000
Fold 2   0.0  400.0  0.0  52.0  0.884956           0.000000
Fold 3   0.0  400.0  0.0  52.0  0.884956           0.000000
Fold 4   0.0  400.0  0.0  52.0  0.884956           0.000000
Fold 5   0.0  400.0  0.0  52.0  0.884956           0.000000
Fold 6   0.0  400.0  0.0  52.0  0.884956           0.000000
Fold 7   0.0  400.0  0.0  52.0  0.884956           0.000000
Fold 8   3.0  395.0  5.0  49.0  0.880531           0.057692
Fold 9   0.0  400.0  0.0  52.0  0.884956           0.000000
Fold 10  0.0  400.0  0.0  52.0  0.884956           0.000000
Average  0.3  399.5  0.5  51.8  0.884318           0.005769

         Specificity (TNR)  Precision  F1 Score  Error Rate  \
Fold 1             1.00000     0.0000      0.00    0.116998
Fold 2             1.00000     0.0000      0.00    0.115044
Fold 3             1.00000     0.0000      0.00    0.115044
Fold 4             1.00000     0.0000      0.00    0.115044
Fold 5             1.00000     0.0000      0.00    0.115044
Fold 6             1.00000     0.0000      0.00    0.115044
Fold 7             1.00000     0.0000      0.00    0.115044
Fold 8             0.98750     0.3750      0.10    0.119469
Fold 9             1.00000     0.0000      0.00    0.115044
Fold 10            1.00000     0.0000      0.00    0.115044
Average            0.99875     0.0375      0.01    0.115682

         Balanced Accuracy (BACC)      FPR       FNR  Brier Score (BS)  \
Fold 1                   0.500000  0.00000  1.000000          0.097935
Fold 2                   0.500000  0.00000  1.000000          0.092020
Fold 3                   0.500000  0.00000  1.000000          0.097377
Fold 4                   0.500000  0.00000  1.000000          0.091375
Fold 5                   0.500000  0.00000  1.000000          0.088473
Fold 6                   0.500000  0.00000  1.000000          0.093067
Fold 7                   0.500000  0.00000  1.000000          0.092690
Fold 8                   0.522596  0.01250  0.942308          0.094317
Fold 9                   0.500000  0.00000  1.000000          0.092773
Fold 10                  0.500000  0.00000  1.000000          0.095183
Average                  0.502260  0.00125  0.994231          0.093521

         Brier Skill Score (BSS)       AUC       HSS       TSS
Fold 1                  0.052019  0.662783  0.000000  0.000000
Fold 2                  0.096156  0.724038  0.000000  0.000000
Fold 3                  0.043534  0.677596  0.000000  0.000000
Fold 4                  0.102491  0.751058  0.000000  0.000000
Fold 5                  0.130994  0.764471  0.000000  0.000000
Fold 6                  0.085863  0.736298  0.000000  0.000000
Fold 7                  0.089569  0.714231  0.000000  0.000000
Fold 8                  0.073588  0.730337  0.071516  0.045192
Fold 9                  0.088753  0.725962  0.000000  0.000000
Fold 10                 0.065087  0.694038  0.000000  0.000000
Average                 0.082805  0.718081  0.007152  0.004519

Average Values Among All Folds:
TP                          0.300000
TN                        399.500000
FP                          0.500000
FN                         51.800000
Accuracy                    0.884318
Sensitivity (TPR)           0.005769
Specificity (TNR)           0.998750
Precision                   0.037500
F1 Score                    0.010000
Error Rate                  0.115682
Balanced Accuracy (BACC)    0.502260
FPR                         0.001250
FNR                         0.994231
Brier Score (BS)            0.093521
Brier Skill Score (BSS)     0.082805
AUC                         0.718081
HSS                         0.007152
TSS                         0.004519
dtype: float64
```

Figure 6: Random Forest Classifer's Metrics with Averages

SVM Metrics:

	TP	TN	FP	FN	Accuracy	Sensitivity (TPR)	\
Fold 1	0.0	400.0	0.0	53.0	0.883002	0.000000	
Fold 2	0.0	400.0	0.0	52.0	0.884956	0.000000	
Fold 3	0.0	400.0	0.0	52.0	0.884956	0.000000	
Fold 4	0.0	400.0	0.0	52.0	0.884956	0.000000	
Fold 5	0.0	400.0	0.0	52.0	0.884956	0.000000	
Fold 6	0.0	400.0	0.0	52.0	0.884956	0.000000	
Fold 7	0.0	400.0	0.0	52.0	0.884956	0.000000	
Fold 8	3.0	395.0	5.0	49.0	0.880531	0.057692	
Fold 9	0.0	400.0	0.0	52.0	0.884956	0.000000	
Fold 10	0.0	400.0	0.0	52.0	0.884956	0.000000	
Average	0.3	399.5	0.5	51.8	0.884318	0.005769	

	Specificity (TNR)	Precision	F1 Score	Error Rate	\
Fold 1	1.00000	0.0000	0.00	0.116998	
Fold 2	1.00000	0.0000	0.00	0.115044	
Fold 3	1.00000	0.0000	0.00	0.115044	
Fold 4	1.00000	0.0000	0.00	0.115044	
Fold 5	1.00000	0.0000	0.00	0.115044	
Fold 6	1.00000	0.0000	0.00	0.115044	
Fold 7	1.00000	0.0000	0.00	0.115044	
Fold 8	0.98750	0.3750	0.10	0.119469	
Fold 9	1.00000	0.0000	0.00	0.115044	
Fold 10	1.00000	0.0000	0.00	0.115044	
Average	0.99875	0.0375	0.01	0.115682	

	Balanced Accuracy (BACC)	FPR	FNR	Brier Score (BS)	\
Fold 1	0.500000	0.00000	1.000000	0.097935	
Fold 2	0.500000	0.00000	1.000000	0.092020	
Fold 3	0.500000	0.00000	1.000000	0.097377	
Fold 4	0.500000	0.00000	1.000000	0.091375	
Fold 5	0.500000	0.00000	1.000000	0.088473	
Fold 6	0.500000	0.00000	1.000000	0.093067	
Fold 7	0.500000	0.00000	1.000000	0.092690	
Fold 8	0.522596	0.01250	0.942308	0.094317	
Fold 9	0.500000	0.00000	1.000000	0.092773	
Fold 10	0.500000	0.00000	1.000000	0.095183	
Average	0.502260	0.00125	0.994231	0.093521	

	Brier Skill Score (BSS)	AUC	HSS	TSS
Fold 1	0.052019	0.662783	0.000000	0.000000
Fold 2	0.096156	0.724038	0.000000	0.000000
Fold 3	0.043534	0.677596	0.000000	0.000000
Fold 4	0.102491	0.751058	0.000000	0.000000
Fold 5	0.130994	0.764471	0.000000	0.000000
Fold 6	0.085863	0.736298	0.000000	0.000000
Fold 7	0.089569	0.714231	0.000000	0.000000
Fold 8	0.073588	0.730337	0.071516	0.045192
Fold 9	0.088753	0.725962	0.000000	0.000000
Fold 10	0.065087	0.694038	0.000000	0.000000
Average	0.082805	0.718081	0.007152	0.004519

Average Values Among All Folds:

TP	0.300000
TN	399.500000
FP	0.500000
FN	51.800000
Accuracy	0.884318
Sensitivity (TPR)	0.005769
Specificity (TNR)	0.998750
Precision	0.037500
F1 Score	0.010000
Error Rate	0.115682
Balanced Accuracy (BACC)	0.502260
FPR	0.001250
FNR	0.994231
Brier Score (BS)	0.093521
Brier Skill Score (BSS)	0.082805
AUC	0.718081
HSS	0.007152
TSS	0.004519

dtype: float64

Figure 7: SVM's Metrics with Averages

LSTM Metrics:

	TP	TN	FP	FN	Accuracy	Sensitivity (TPR)
Fold 1	0.0	400.0	0.0	53.0	0.883002	0.000000
Fold 2	0.0	400.0	0.0	52.0	0.884956	0.000000
Fold 3	0.0	400.0	0.0	52.0	0.884956	0.000000
Fold 4	0.0	400.0	0.0	52.0	0.884956	0.000000
Fold 5	0.0	400.0	0.0	52.0	0.884956	0.000000
Fold 6	0.0	400.0	0.0	52.0	0.884956	0.000000
Fold 7	0.0	400.0	0.0	52.0	0.884956	0.000000
Fold 8	3.0	395.0	5.0	49.0	0.880531	0.057692
Fold 9	0.0	400.0	0.0	52.0	0.884956	0.000000
Fold 10	0.0	400.0	0.0	52.0	0.884956	0.000000
Average	0.3	399.5	0.5	51.8	0.884318	0.005769

	Specificity (TNR)	Precision	F1 Score	Error Rate
Fold 1	1.00000	0.0000	0.00	0.116998
Fold 2	1.00000	0.0000	0.00	0.115044
Fold 3	1.00000	0.0000	0.00	0.115044
Fold 4	1.00000	0.0000	0.00	0.115044
Fold 5	1.00000	0.0000	0.00	0.115044
Fold 6	1.00000	0.0000	0.00	0.115044
Fold 7	1.00000	0.0000	0.00	0.115044
Fold 8	0.98750	0.3750	0.10	0.119469
Fold 9	1.00000	0.0000	0.00	0.115044
Fold 10	1.00000	0.0000	0.00	0.115044
Average	0.99875	0.0375	0.01	0.115682

	Balanced Accuracy (BACC)	FPR	FNR	Brier Score (BS)
Fold 1	0.500000	0.00000	1.000000	0.097935
Fold 2	0.500000	0.00000	1.000000	0.092020
Fold 3	0.500000	0.00000	1.000000	0.097377
Fold 4	0.500000	0.00000	1.000000	0.091375
Fold 5	0.500000	0.00000	1.000000	0.088473
Fold 6	0.500000	0.00000	1.000000	0.093067
Fold 7	0.500000	0.00000	1.000000	0.092690
Fold 8	0.522596	0.01250	0.942308	0.094317
Fold 9	0.500000	0.00000	1.000000	0.092773
Fold 10	0.500000	0.00000	1.000000	0.095183
Average	0.502260	0.00125	0.994231	0.093521

	Brier Skill Score (BSS)	AUC	HSS	TSS
Fold 1	0.052019	0.662783	0.000000	0.000000
Fold 2	0.096156	0.724038	0.000000	0.000000
Fold 3	0.043534	0.677596	0.000000	0.000000
Fold 4	0.102491	0.751058	0.000000	0.000000
Fold 5	0.130994	0.764471	0.000000	0.000000
Fold 6	0.085863	0.736298	0.000000	0.000000
Fold 7	0.089569	0.714231	0.000000	0.000000
Fold 8	0.073588	0.730337	0.071516	0.045192
Fold 9	0.088753	0.725962	0.000000	0.000000
Fold 10	0.065087	0.694038	0.000000	0.000000
Average	0.082805	0.718081	0.007152	0.004519

Average Values Among All Folds:

TP	0.300000
TN	399.500000
FP	0.500000
FN	51.800000
Accuracy	0.884318
Sensitivity (TPR)	0.005769
Specificity (TNR)	0.998750
Precision	0.037500
F1 Score	0.010000
Error Rate	0.115682
Balanced Accuracy (BACC)	0.502260
FPR	0.001250
FNR	0.994231
Brier Score (BS)	0.093521
Brier Skill Score (BSS)	0.082805
AUC	0.718081
HSS	0.007152
TSS	0.004519

dtype: float64

Figure 8: LSTM's Metrics with Averages

```
Time taken to output: 127.02 seconds
```

Figure 9: Total time it took to run the code in seconds

Github link: https://github.com/vaibhavbora11/Data-Mining-Final-Project

Dataset Source: https://archive.ics.uci.edu/dataset/222/bank+marketing