Zara F. Amer

Worked

1. Filling NA’s with 0’s -> GINI: 28.7%
2. XGBoost Classifier -> Gini: 37%
3. XGB with SMOTE -> Gini: 41%
4. 10 Generation Genetic Algorithm -> Gini: 82%
5. Balancing hyperparameters using Random Forest Classifier -> Gini: 73%
   - Bootstrap: false
   - Max_depth: 84
   - Max_feautures: sqrt
   - Min_samples_leaf: 1
   - Min_samples_split: 2
   - N_estimators: 255

Did not work

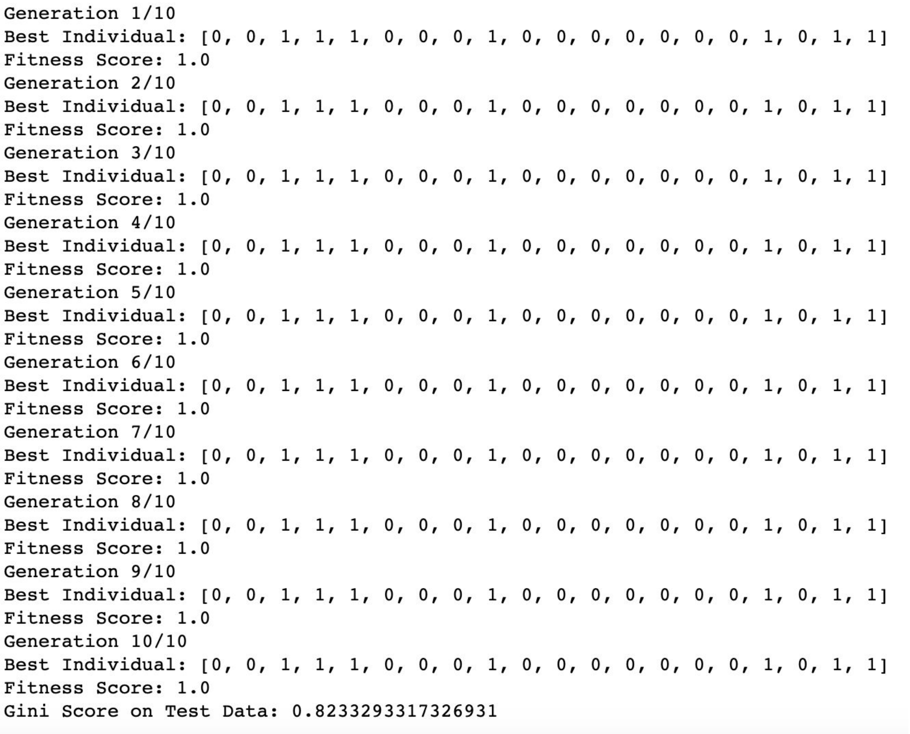
1. Logistic Regression
2. Applying Standard Scaling

Wish List

(things you wish to have tested but didn’t have time)

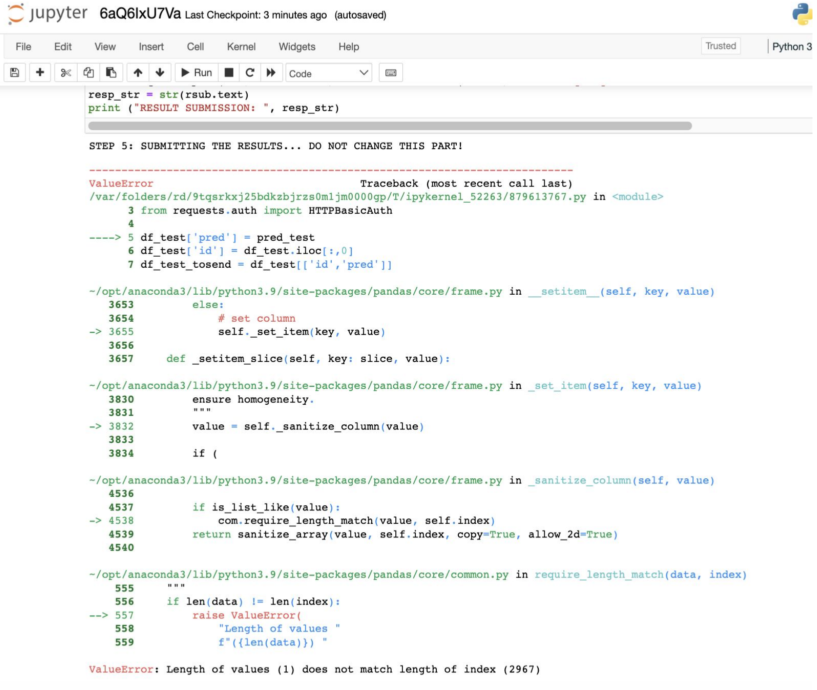
1. Bagging
2. Filling NA’s with Mean, Median
3. Chi-Squared
4. Support Vector Machines (SVM)
5. Neural Networks
6. Isolation Forests
7. K-nearest neighbor (KNN)

GENETIC ALGORITHM



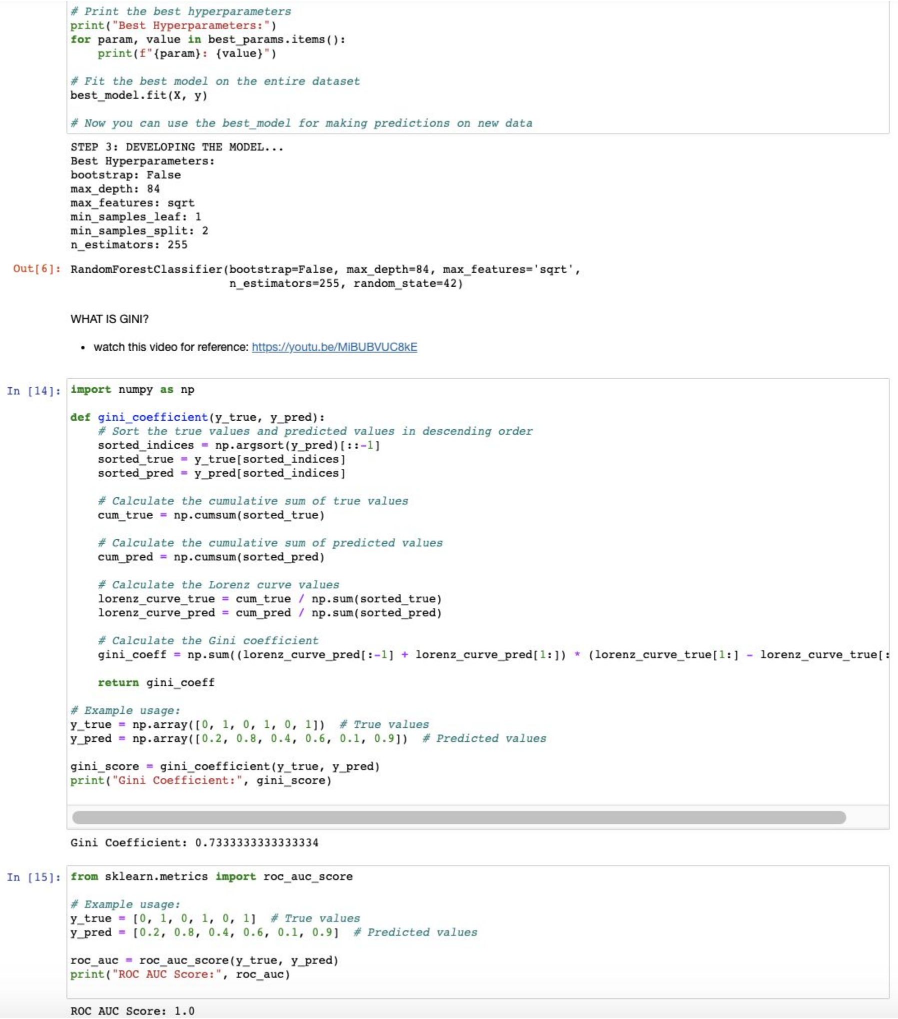
```
Generation 1/10
Best Individual: [0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1]
Fitness Score: 1.0
Generation 2/10
Best Individual: [0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1]
Fitness Score: 1.0
Generation 3/10
Best Individual: [0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1]
Fitness Score: 1.0
Generation 4/10
Best Individual: [0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1]
Fitness Score: 1.0
Generation 5/10
Best Individual: [0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1]
Fitness Score: 1.0
Generation 6/10
Best Individual: [0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1]
Fitness Score: 1.0
Generation 7/10
Best Individual: [0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1]
Fitness Score: 1.0
Generation 8/10
Best Individual: [0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1]
Fitness Score: 1.0
Generation 9/10
Best Individual: [0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1]
Fitness Score: 1.0
Generation 10/10
Best Individual: [0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1]
Fitness Score: 1.0
Gini Score on Test Data: 0.8233293317326931
```

I tried a Genetic Algorithm but could not figure out how to upload/submit the result to the challenge. It gave me a Value Error in Part 5 (see below). However, the code can be found in the notebook and the picture on the left shows the result of the algorithm in Part 3.



```
resp_str = str(rsub.text)
print ("RESULT SUBMISSION: ", resp_str)
```

```
STEP 5: SUBMITTING THE RESULTS... DO NOT CHANGE THIS PART!

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/rd/9tqsrkxj25bdkzbjrzs0m1jm0000gp/T/ipykernel_52263/879613767.py in <module>
      3 from requests.auth import HTTPBasicAuth
      4 
----> 5 df_test['pred'] = pred_test
      6 df_test['id'] = df_test.iloc[:,0]
      7 df_test_tosend = df_test[['id','pred']]

~/opt/anaconda3/lib/python3.9/site-packages/pandas/core/frame.py in __setitem__(self, key, value)
   3653         else:
   3654             # set column
-> 3655             self._set_item(key, value)
   3656 
   3657     def _setitem_slice(self, key: slice, value):

~/opt/anaconda3/lib/python3.9/site-packages/pandas/core/frame.py in _set_item(self, key, value)
   3830         ensure homogeneity.
   3831         """
-> 3832         value = self._sanitize_column(value)
   3833 
   3834         if (

~/opt/anaconda3/lib/python3.9/site-packages/pandas/core/frame.py in _sanitize_column(self, value)
   4536 
   4537         if is_list_like(value):
-> 4538             com.require_length_match(value, self.index)
   4539         return sanitize_array(value, self.index, copy=True, allow_2d=True)
   4540 

~/opt/anaconda3/lib/python3.9/site-packages/pandas/core/common.py in require_length_match(data, index)
    555     """
    556     if len(data) != len(index):
--> 557         raise ValueError(
    558             "Length of values "
    559             f"({len(data)}) "

ValueError: Length of values (1) does not match length of index (2967)
```



```
# Print the best hyperparameters
print("Best Hyperparameters:")
for param, value in best_params.items():
    print(f"{param}: {value}")

# Fit the best model on the entire dataset
best_model.fit(X, y)

# Now you can use the best_model for making predictions on new data
```

```
STEP 3: DEVELOPING THE MODEL...
Best Hyperparameters:
bootstrap: False
max_depth: 84
max_features: sqrt
min_samples_leaf: 1
min_samples_split: 2
n_estimators: 255
```

```
Out[6]: RandomForestClassifier(bootstrap=False, max_depth=84, max_features='sqrt',
                               n_estimators=255, random_state=42)
```

WHAT IS GINI?

- watch this video for reference: https://youtu.be/MiBUBVUC8kE

```
In [14]: import numpy as np

def gini_coefficient(y_true, y_pred):
    # Sort the true values and predicted values in descending order
    sorted_indices = np.argsort(y_pred)[::-1]
    sorted_true = y_true[sorted_indices]
    sorted_pred = y_pred[sorted_indices]

    # Calculate the cumulative sum of true values
    cum_true = np.cumsum(sorted_true)

    # Calculate the cumulative sum of predicted values
    cum_pred = np.cumsum(sorted_pred)

    # Calculate the Lorenz curve values
    lorenz_curve_true = cum_true / np.sum(sorted_true)
    lorenz_curve_pred = cum_pred / np.sum(sorted_pred)

    # Calculate the Gini coefficient
    gini_coeff = np.sum((lorenz_curve_pred[:-1] + lorenz_curve_pred[1:]) * (lorenz_curve_true[1:] - lorenz_curve_true[:

    return gini_coeff

# Example usage:
y_true = np.array([0, 1, 0, 1, 0, 1])  # True values
y_pred = np.array([0.2, 0.8, 0.4, 0.6, 0.1, 0.9])  # Predicted values

gini_score = gini_coefficient(y_true, y_pred)
print("Gini Coefficient:", gini_score)
```

```
Gini Coefficient: 0.7333333333333334
```

```
In [15]: from sklearn.metrics import roc_auc_score

# Example usage:
y_true = [0, 1, 0, 1, 0, 1]  # True values
y_pred = [0.2, 0.8, 0.4, 0.6, 0.1, 0.9]  # Predicted values

roc_auc = roc_auc_score(y_true, y_pred)
print("ROC AUC Score:", roc_auc)
```

```
ROC AUC Score: 1.0
```

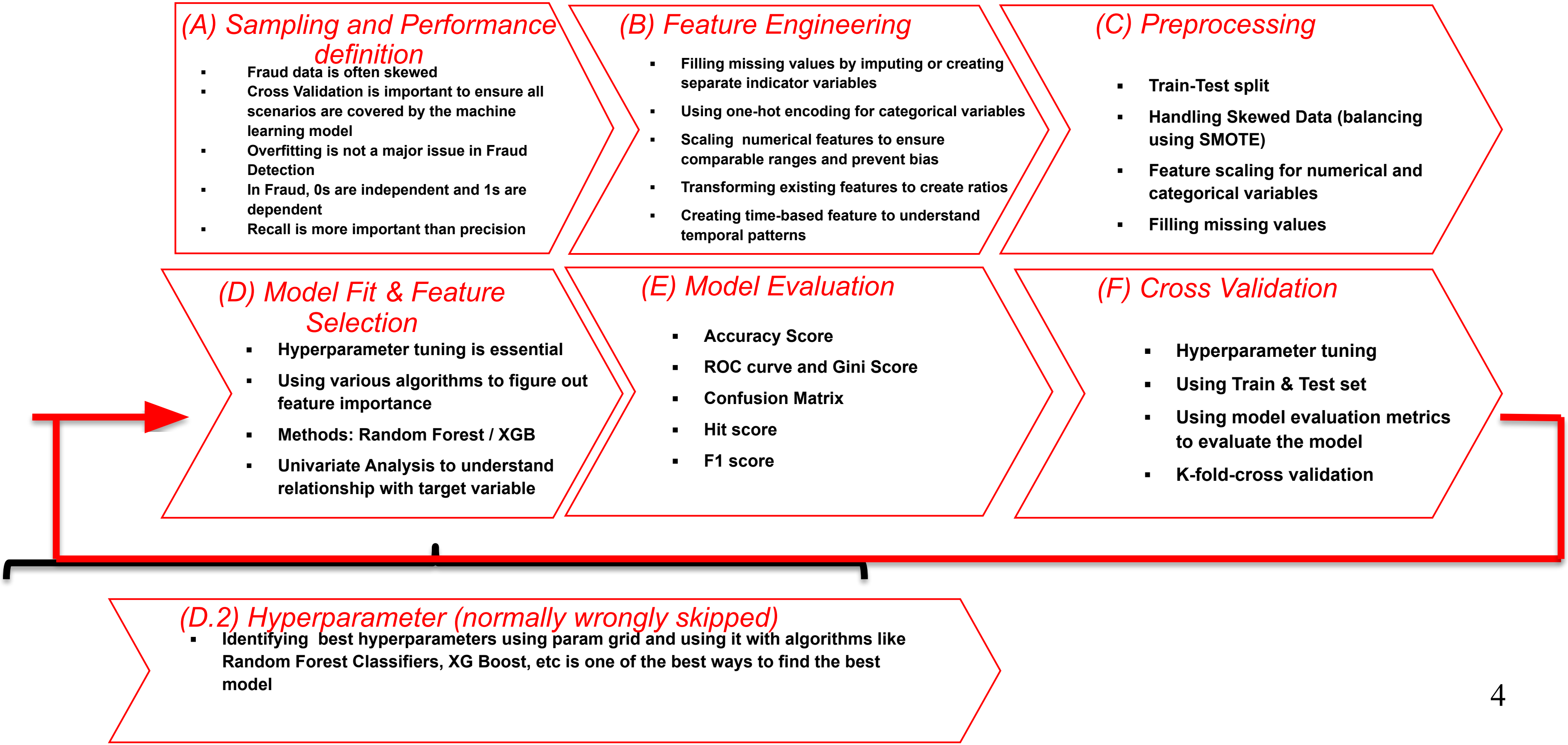
Random Forest Classifier with Hyperparameters as follows:

- Bootstrap: false
- Max_depth: 84
- Max_feautures: sqrt
- Min_samples_leaf: 1
- Min_samples_split: 2
- N_estimators: 255

However, I was unable to submit the result in Part 5 and got the same Value Error as in the previous slide on Genetic Algorithms.

SCHOOL OF
HUMAN SCIENCES
& TECHNOLOGY

Main steps in Machine Learning (role 3)



(A) Sampling and Performance definition

- Fraud data is often skewed
- Cross Validation is important to ensure all scenarios are covered by the machine learning model
- Overfitting is not a major issue in Fraud Detection
- In Fraud, 0s are independent and 1s are dependent
- Recall is more important than precision

(B) Feature Engineering

- Filling missing values by imputing or creating separate indicator variables
- Using one-hot encoding for categorical variables
- Scaling numerical features to ensure comparable ranges and prevent bias
- Transforming existing features to create ratios
- Creating time-based feature to understand temporal patterns

(C) Preprocessing

- Train-Test split
- Handling Skewed Data (balancing using SMOTE)
- Feature scaling for numerical and categorical variables
- Filling missing values

(D) Model Fit & Feature Selection

- Hyperparameter tuning is essential
- Using various algorithms to figure out feature importance
- Methods: Random Forest / XGB
- Univariate Analysis to understand relationship with target variable

(E) Model Evaluation

- Accuracy Score
- ROC curve and Gini Score
- Confusion Matrix
- Hit score
- F1 score

(F) Cross Validation

- Hyperparameter tuning
- Using Train & Test set
- Using model evaluation metrics to evaluate the model
- K-fold-cross validation

(D.2) Hyperparameter (normally wrongly skipped)

- Identifying best hyperparameters using param grid and using it with algorithms like Random Forest Classifiers, XG Boost, etc is one of the best ways to find the best model