6211 Assignment 1

Question 1.)

```
In [4]: ap_df.info()
executed in 85ms, finished 13:33:01 2023-02-07
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 91482 entries, 0 to 91481
Data columns (total 29 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   ETHNICITY            71772 non-null  object
 1   TERRITORY            91481 non-null  object
 2   ACADEMIC_INTEREST_1  44283 non-null  object
 3   ACADEMIC_INTEREST_2  24879 non-null  object
 4   Enroll               91482 non-null  int64
 5   CONTACT_DATE         91482 non-null  object
 6   TOTAL_CONTACTS       91482 non-null  int64
 7   SELF_INIT_CNTCTS     91482 non-null  int64
 8   TRAVEL_INIT_CNTCTS   91482 non-null  int64
 9   SOLICITED_CNTCTS     91482 non-null  int64
 10  REFERRAL_CNTCTS      91482 non-null  int64
 11  CAMPUS_VISIT         91482 non-null  int64
 12  CONTACT_CODE1        91339 non-null  object
 13  LEVEL_YEAR           91482 non-null  object
 14  IRSCHOOL             76172 non-null  object
 15  satscore             27003 non-null  float64
 16  sex                  87649 non-null  float64
 17  mailq                91482 non-null  int64
 18  telecq               20602 non-null  float64
 19  premiere             91482 non-null  int64
 20  interest             91482 non-null  int64
 21  stucar               91482 non-null  int64
 22  init_span            91482 non-null  int64
 23  int1rat              91482 non-null  float64
 24  int2rat              91482 non-null  float64
 25  hscrat               91482 non-null  float64
 26  avg_income           70553 non-null  float64
 27  distance             72014 non-null  float64
 28  Instate              91482 non-null  object
dtypes: float64(8), int64(12), object(9)
memory usage: 20.2+ MB
```

c)
- Variables 'ACADEMIC_INTEREST_1', 'ACADEMIC_INTEREST_2', 'IRSCHOOL', 'CONTACT_CODE1', 'CONTACT_DATE' are dropped as mentioned in the question.
- Column 'Level_Year' is to be dropped as the variable has only 1 unique value 'FR22' and has no impact on the predictive models.
- 'Satscore' and 'telecq' have more than 70% missing values and will not make a good feature. Hence, dropping these two columns.
- Variables Total_Contacts, Mailq have been for the regression model as they have **VIF** values greater than the threshold value of **10** during model building in the later steps.

d) Variable 'Enroll' is the target variable.

Question 2.)
- Categorical variables 'ETHNICITY', 'TERRITORY' and 'INSTATE' are to be dummied.
- Category 'A' from variable ETHNICITY is left out and similarly TERRITORY '0' along with INSTATE 'N' are left out.

Question 3.)
- Variable **Imputation** is required for model building but **Transformation** isn't required.
- Variable '**Ethnicity'** has been imputed with value **'C'** as it is the most frequently occurred value.
- Variable '**Territory'** has been imputed with value **'0'** as the missing value percentage is close to 0, and so filling it with '0' won't make any difference to the model.
- Variable '**sex'** has been imputed with value **1** as it is the most frequently occurred value.

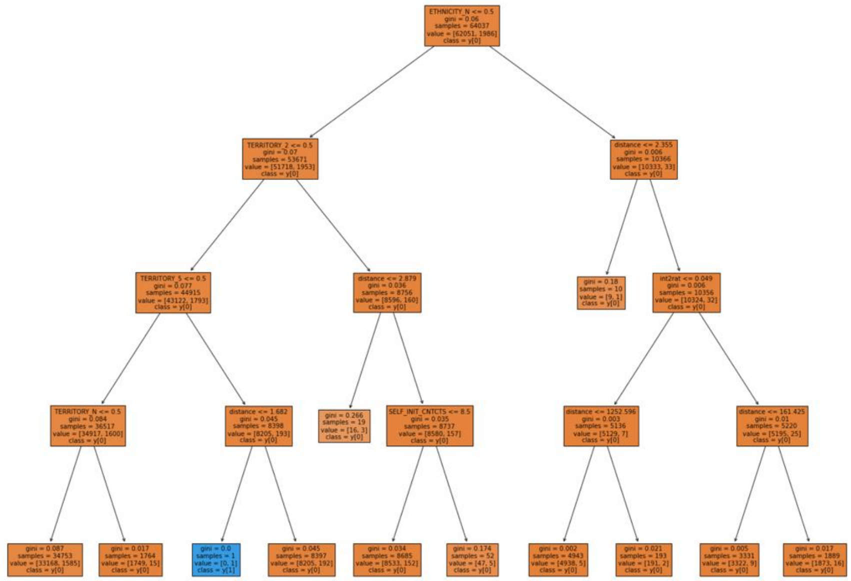
- Variable ‘**avg_income** has been imputed with its **mean value** as the variable is normally distributed**.**
- Variable ‘**distance’** has been imputed with its **mean value** as the variable is right skewed.
- Variables ‘Total_Contacts’ and ‘SELF_INIT_CNTCTS’ have been grouped into bins of 0 and 1. (if value greater than 0 then assigned value as 1 else 0).

Question 4.)

a) Logit Model Summary

```
Optimization terminated successfully.
         Current function value: 0.130450
         Iterations 10
                           Logit Regression Results
==============================================================================
Dep. Variable:                 Enroll   No. Observations:                64037
Model:                          Logit   Df Residuals:                    64005
Method:                           MLE   Df Model:                           31
Date:                Tue, 07 Feb 2023   Pseudo R-squ.:                 0.05640
Time:                        21:14:28   Log-Likelihood:                -8353.6
converged:                       True   LL-Null:                       -8852.9
Covariance Type:            nonrobust   LLR p-value:                 5.854e-190
======================================================================================
                         coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------
SELF_INIT_CNTCTS      -0.0074      0.053     -0.140      0.889      -0.111       0.096
TRAVEL_INIT_CNTCTS    -0.0716      0.048     -1.504      0.133      -0.165       0.022
SOLICITED_CNTCTS       0.0541      0.040      1.360      0.174      -0.024       0.132
REFERRAL_CNTCTS       -0.0357      0.105     -0.341      0.733      -0.241       0.169
CAMPUS_VISIT           0.0662      0.124      0.533      0.594      -0.177       0.309
sex                    0.0154      0.048      0.318      0.751      -0.079       0.110
premiere               0.0085      0.141      0.060      0.952      -0.269       0.286
interest               0.0792      0.093      0.852      0.394      -0.103       0.261
stucar                -0.0404      0.051     -0.798      0.425      -0.140       0.059
init_span             -0.0021      0.003     -0.723      0.470      -0.008       0.004
int1rat               -0.8614      1.350     -0.638      0.523      -3.507       1.784
int2rat                1.9032      1.195      1.593      0.111      -0.438       4.245
hscrat                 0.3373      0.395      0.854      0.393      -0.437       1.111
avg_income          -3.47e-06   1.38e-06     -2.510      0.012   -6.18e-06    -7.6e-07
distance              -0.0002   9.14e-05     -2.084      0.037      -0.000   -1.13e-05
ETHNICITY_B           -0.8502      0.116     -7.340      0.000      -1.077      -0.623
ETHNICITY_C           -0.5111      0.089     -5.746      0.000      -0.685      -0.337
ETHNICITY_H           -0.7468      0.109     -6.832      0.000      -0.961      -0.533
ETHNICITY_I           -0.9763      0.318     -3.073      0.002      -1.599      -0.354
ETHNICITY_N           -3.0555      0.194    -15.759      0.000      -3.435      -2.675
ETHNICITY_O           -0.8985      0.172     -5.222      0.000      -1.236      -0.561
TERRITORY_1           -2.0417      0.154    -13.290      0.000      -2.343      -1.741
TERRITORY_2           -3.1630      0.165    -19.162      0.000      -3.487      -2.839
TERRITORY_3           -2.0986      0.154    -13.641      0.000      -2.400      -1.797
TERRITORY_4           -2.2930      0.158    -14.508      0.000      -2.603      -1.983
TERRITORY_5           -2.9224      0.163    -17.962      0.000      -3.241      -2.604
TERRITORY_6           -2.2118      0.158    -13.984      0.000      -2.522      -1.902
TERRITORY_7           -2.6265      0.160    -16.375      0.000      -2.941      -2.312
TERRITORY_8           -1.8844      0.158    -11.919      0.000      -2.194      -1.575
TERRITORY_A           -3.4108      0.238    -14.304      0.000      -3.878      -2.943
TERRITORY_N           -3.9805      0.297    -13.416      0.000      -4.562      -3.399
Instate_Y             -0.0544      0.087     -0.628      0.530      -0.224       0.115
======================================================================================
```

b) Tree Plot



Question 5.)
I'll choose Logistic Regression as its ROC AUC value **(0.685)** is higher than Decision Tree ROC AUC value **(0.636).**

Question 6.)

- Given dataset is highly imbalanced and requires resampling for building better models.
- Based on the regression model, variables **AVG_INCOME, DISTANCE, ETHNICITY** and **TERRITORY** are the significant variables in the given dataset as their respective **p-values** are less than the threshold of **0.05** whereas the rest variables have higher p-values than 0.05. Keeping the rest variables constant, with every unit increase in either **AVG_INCOME or DISTANCE or ETHNICITY** or **TERRITORY**, the target value **Enroll** decreases.

Question 7.)

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


import pandas as pd
import numpy as np
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn.model_selection import train_test_split
import statsmodels.api as sm
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score
from matplotlib import pyplot as plt
from sklearn import tree
from sklearn.tree import export_text


# In[2]:


ap_df = pd.read_csv("inq2022.csv")


# In[3]:


ap_df.head()


# In[4]:


ap_df.info()


# In[5]:


ap_df.describe(include = 'all')
```

```
# In[6]:


ap_df.isnull().sum()/len(ap_df)


# In[7]:


ap_df = ap_df.drop(columns=['ACADEMIC_INTEREST_1','ACADEMIC_INTEREST_2','IRSCHOOL', 'CONTACT_CODE1','CONTACT_DATE'])


# In[8]:


ap_df.describe(include='all')


# In[9]:


ap_df.info()


# In[10]:


ap_df.isna().sum()/len(ap_df)


# In[11]:


ap_df = ap_df.drop(columns=['satscore','telecq'])


# In[12]:


ap_df = ap_df.drop(columns=['LEVEL_YEAR'])


# In[13]:
```

```
ap_df.Enroll.value_counts()


# In[14]:


cat_columns = ['ETHNICITY','TERRITORY','Instate']


# In[15]:


for col in cat_columns:
    print(ap_df[col].value_counts())
print(len(ap_df))
print(ap_df.isna().sum())


# In[16]:


ap_df['ETHNICITY'] = ap_df['ETHNICITY'].fillna('C') ## Most frequent occurence
ap_df['TERRITORY'] = ap_df['TERRITORY'].fillna('0') ## Null % is close to 0, so filling with least occurence value as it doesn't have much impact on the model
ap_df['sex'] = ap_df['sex'].fillna(1) ## Highest occurence
ap_df['avg_income'] = ap_df['avg_income'].fillna(ap_df['avg_income'].mean()) ## Right skewed so filling nulls with mean
ap_df['distance'] = ap_df['distance'].fillna(ap_df['distance'].mean()) ## Right skewed so filling nulls with mean


# In[17]:


ap_df =pd.get_dummies(ap_df, drop_first=True)

ap_df_tree = ap_df.copy() ## Making a replica for Decision Tree Model

ap_df.describe()


# In[18]:
```

```
ap_df.info()


# In[19]:


ap_df[['TOTAL_CONTACTS','SELF_INIT_CNTCTS','TRAVEL_INIT_CNTCTS',
       'SOLICITED_CNTCTS','REFERRAL_CNTCTS','CAMPUS_VISIT',
       'sex','mailq','premiere','interest','stucar','init_span','int1rat',
       'int2rat','hscrat','avg_income','distance']].skew(axis = 0, skipna = True)


# In[20]:


ap_df[['TOTAL_CONTACTS','SELF_INIT_CNTCTS','TRAVEL_INIT_CNTCTS',
       'SOLICITED_CNTCTS','REFERRAL_CNTCTS','CAMPUS_VISIT',
       'sex','mailq','premiere','interest','stucar','init_span','int1rat',
       'int2rat','hscrat','avg_income','distance']].hist(bins=30, figsize=(20, 15))


# In[21]:


ap_df['TOTAL_CONTACTS'] = np.where(ap_df['TOTAL_CONTACTS']>0,1,0)
ap_df['SELF_INIT_CNTCTS'] = np.where(ap_df['SELF_INIT_CNTCTS']>0,1,0)


# In[22]:


ap_df.drop(columns=['Enroll']).hist(bins=30, figsize=(20, 15))


# Considering VIF threshold as 10

# In[23]:


ap_vif = pd.DataFrame()
ap_vif["feature"] = ap_df.drop(columns=['Enroll']).columns

ap_vif["VIF"] = [variance_inflation_factor(ap_df.drop(columns=['Enroll']).values, i)
                 for i in range(len(ap_df.drop(columns=['Enroll']).columns))]

print(ap_vif)
```

```
# In[24]:


ap_df.corr()


# In[25]:


ap_df = ap_df.drop(columns=['TOTAL_CONTACTS']) ### Highest VIF value


# In[26]:


ap_vif = pd.DataFrame()
ap_vif["feature"] = ap_df.drop(columns=['Enroll']).columns

ap_vif["VIF"] = [variance_inflation_factor(ap_df.drop(columns=['Enroll']).values, i)
                 for i in range(len(ap_df.drop(columns=['Enroll']).columns))]

print(ap_vif)


# In[27]:


ap_df.corr()


# In[28]:


ap_df = ap_df.drop(columns=['mailq']) ### Highest VIF value


# In[29]:


ap_vif = pd.DataFrame()
ap_vif["feature"] = ap_df.drop(columns=['Enroll']).columns

ap_vif["VIF"] = [variance_inflation_factor(ap_df.drop(columns=['Enroll']).values, i)
                 for i in range(len(ap_df.drop(columns=['Enroll']).columns))]
```

```
print(ap_vif)


# ## Regression Model

# In[30]:


ap_X_train, ap_X_val, ap_y_train, ap_y_val = train_test_split(ap_df.drop(columns=['Enroll']),ap_df['Enroll'], test_size=0.3,random_state=0)
ap_log_reg = sm.Logit(ap_y_train, ap_X_train).fit()
print(ap_log_reg.summary())


# In[31]:


ap_prediction_prob = ap_log_reg.predict(ap_X_val)
ap_prediction = list(map(round, ap_prediction_prob))
confusion_matrix(ap_y_val,ap_prediction)


# In[32]:


lr_auc = roc_auc_score(ap_y_val, ap_prediction_prob)
print('Logistic: ROC AUC=%.3f' % (lr_auc))


# In[33]:


lr_fpr, lr_tpr, _ = roc_curve(ap_y_val, ap_prediction_prob)
plt.plot(lr_fpr, lr_tpr, marker='.')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.show()


# ## Tree Model

# In[34]:
```

```
ap_X_train, ap_X_val, ap_y_train, ap_y_val =
train_test_split(ap_df_tree.drop(columns=['Enroll']),ap_df_tree['Enroll'],
test_size=0.3,random_state=0)
ap_dtree = tree.DecisionTreeClassifier(max_depth=4,min_samples_split=30)
ap_dtree = ap_dtree.fit(ap_X_train, ap_y_train)


# In[35]:


ap_r = export_text(ap_dtree, feature_names=list(ap_X_train.columns.values))
print(ap_r)


# In[36]:


plt.figure(figsize=[25,20])
tree.plot_tree(ap_dtree,
            feature_names=list(ap_X_train.columns.values),
            class_names=True,
            filled=True)
plt.show()


# In[37]:


ap_prediction =ap_dtree.predict(ap_X_val)
confusion_matrix(ap_y_val,ap_prediction)


# In[38]:


ap_dtree.score(ap_X_val,ap_y_val)


# In[39]:


ap_prediction_prob = ap_dtree.predict_proba(ap_X_val)
ap_tree_auc = roc_auc_score(ap_y_val, ap_prediction_prob[:,1])
print('Decision Tree: ROC AUC=%.3f' % (ap_tree_auc))
```

In[40]:

```
tree_fpr, tree_tpr, _ = roc_curve(ap_y_val, ap_prediction_prob[:,1])
plt.plot(tree_fpr, tree_tpr, marker='.')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.show()
```