About

Enacted by Congress in 2020 to respond to the economic impact of the COVID-19 pandemic, the Paycheck Protection Program provided nearly $800 billion in loans to small businesses in order to retain payrolls. The Small Business Administration has oversight over the PPP program, although the loans are administered by private lenders, who then submit application information to the government. The loans could be fully forgiven if certain conditions were met.

The SBA has periodically released data on the more than 11.5 million approved applications, but it also has removed applications that had been previously present in the dataset.

Importing Libraries

In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from sklearn.model_selection import train_test_split
from scipy.optimize import curve_fit
from sklearn import metrics
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
```

Reading Data

In [2]:
```
df=pd.read_excel("PPP_updated.xlsx")
```

Data Preprocessing and EDA

In [3]:
```
df.head()
```

Out[3]:

	Unnamed: 0	name	amount	state	address	city	zip	naics_code	NAICS_Num	business_type	...	originating_lender_city	originating_lender_state	loan_status
0	0	LUMMUS CORPORATION	2000000.0	GA	225 Bourne Blvd	Savannah	31408-9586	333249.0	33.0	Corporation	...	CHICAGO	IL	
1	1	COLIANT SOLUTIONS INC.	1294555.0	GA	2703 Brickton North Dr	Buford	30518-9101	541519.0	54.0	Corporation	...	COLUMBUS	GA	
2	2	YOHE PLUMBING INC	729509.0	GA	1120 Franke Court N/A	Augusta	30909	238220.0	23.0	Corporation	...	PHOENIXVILLE	PA	
3	3	LEWIS COLOR LITHOGRAPHERS INC	571193.4	GA	30 Joe Kennedy Blvd	Statesboro	30458-3417	323111.0	32.0	Corporation	...	COLUMBUS	GA	
4	4	ALMA PAK INTERNATIONAL, LLC	472700.0	GA	230 PINEVIEW RD	ALMA	31510-4326	445230.0	44.0	Partnership	...	ALMA	GA	2021-

5 rows × 46 columns

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 579664 entries, 0 to 579663
Data columns (total 46 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   Unnamed: 0                     579664 non-null  int64
 1   name                           579661 non-null  object
 2   amount                         579664 non-null  float64
 3   state                          579664 non-null  object
 4   address                        579663 non-null  object
 5   city                           579664 non-null  object
 6   zip                            579664 non-null  object
 7   naics_code                     577324 non-null  float64
 8   NAICS_Num                      577324 non-null  float64
 9   business_type                  579623 non-null  object
 10  jobs_retained                  579664 non-null  int64
 11  date_approved                  579664 non-null  datetime64[ns]
 12  lender                         579664 non-null  object
 13  congressional_district         579659 non-null  object
 14  loan_number                    579664 non-null  int64
 15  sba_office_code                579664 non-null  int64
 16  processing_method              579664 non-null  object
 17  loan_status                    579664 non-null  object
 18  term                           579664 non-null  int64
 19  sba_guaranty_percentage        579664 non-null  int64
 20  initial_approval_amount        579664 non-null  float64
 21  current_approval_amount        579664 non-null  float64
 22  undisbursed_amount             579548 non-null  float64
 23  servicing_lender_location_id   579664 non-null  int64
 24  servicing_lender_name          579664 non-null  object
 25  servicing_lender_address       579664 non-null  object
 26  servicing_lender_city          579664 non-null  object
 27  servicing_lender_state         579664 non-null  object
 28  servicing_lender_zip           579664 non-null  object
 29  rural_urban_indicator          579664 non-null  object
 30  hubzone_indicator              579664 non-null  object
 31  business_age_description       579664 non-null  object
 32  project_city                   579664 non-null  object
 33  project_county_name            579658 non-null  object
 34  project_state                  579664 non-null  object
 35  project_zip                    579664 non-null  object
 36  originating_lender_city        579664 non-null  object
 37  originating_lender_state       579664 non-null  object
 38  loan_status_date               303920 non-null  datetime64[ns]
 39  originating_lender_location_id 579664 non-null  int64
 40  lmi_indicator                  579664 non-null  object
 41  forgiveness_amount             371197 non-null  float64
 42  forgiveness_date               371197 non-null  datetime64[ns]
 43  Status                         579664 non-null  object
 44  Unnamed: 43                    0 non-null       float64
 45  naics_code_def                 579664 non-null  object
dtypes: datetime64[ns](3), float64(8), int64(8), object(27)
memory usage: 203.4+ MB
```

After doing thorough Exploratory Data Analysis in Tableau came up with the below features that are important to build a model Subsetting the data with inly required columns

In [5]:
```
df1=df[["name","amount","zip","naics_code","NAICS_Num","business_type","jobs_retained","date_approved","lender","loan_status","initial_approval_amount",'
```

In [6]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 579664 entries, 0 to 579663
Data columns (total 17 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   name                      579661 non-null  object
 1   amount                    579664 non-null  float64
 2   zip                       579664 non-null  object
 3   naics_code                577324 non-null  float64
 4   NAICS_Num                 577324 non-null  float64
 5   business_type             579623 non-null  object
 6   jobs_retained             579664 non-null  int64
 7   date_approved             579664 non-null  datetime64[ns]
 8   lender                    579664 non-null  object
 9   loan_status               579664 non-null  object
 10  initial_approval_amount   579664 non-null  float64
 11  current_approval_amount   579664 non-null  float64
 12  undisbursed_amount        579548 non-null  float64
 13  hubzone_indicator         579664 non-null  object
 14  business_age_description  579664 non-null  object
 15  project_county_name       579658 non-null  object
 16  Status                    579664 non-null  object
dtypes: datetime64[ns](1), float64(6), int64(1), object(9)
memory usage: 75.2+ MB
```

In [7]:
```
df1.head()
```

Out[7]:

	name	amount	zip	naics_code	NAICS_Num	business_type	jobs_retained	date_approved	lender	loan_status	initial_approval_amount	current_approval_amount
0	LUMMUS CORPORATION	2000000.0	31408-9586	333249.0	33.0	Corporation	294	2021-02-20	CIBC Bank USA	Exemption 4	2000000.0	2000000.0
1	COLIANT SOLUTIONS INC.	1294555.0	30518-9101	541519.0	54.0	Corporation	63	2021-01-31	Synovus Bank	Exemption 4	1294555.0	1294555.0
2	YOHE PLUMBING INC	729509.0	30909	238220.0	23.0	Corporation	105	2021-03-12	Customers Bank	Exemption 4	729509.0	729509.0
3	LEWIS COLOR LITHOGRAPHERS INC	571193.4	30458-3417	323111.0	32.0	Corporation	49	2021-02-02	Synovus Bank	Exemption 4	571194.0	571193.4
4	ALMA PAK INTERNATIONAL, LLC	472700.0	31510-4326	445230.0	44.0	Partnership	37	2020-04-10	FNB South	Paid in Full	472700.0	472700.0

Removing the null data

In [8]:
```
df1.isna().sum(axis=0)
```

Out[8]:
```
name                           3
amount                         0
zip                            0
naics_code                  2340
NAICS_Num                   2340
business_type                 41
jobs_retained                  0
date_approved                  0
lender                         0
loan_status                    0
initial_approval_amount        0
current_approval_amount        0
undisbursed_amount           116
hubzone_indicator              0
business_age_description       0
project_county_name            6
Status                         0
dtype: int64
```

In [9]:
```
df1=df1.dropna()
```

In [10]:
```
df1.groupby(["Status"])["name"].count()
```

Out[10]:
```
Status
Not removed    551328
Removed         25835
Name: name, dtype: int64
```

In [11]:
```
df1['Is_Prestamos_CDFI_LLC']=df1['lender'].apply(lambda x: 1 if(x=='Prestamos CDFI, LLC') else 0)
df1['Is_Capital_Plus_Financial_LLC']=df1['lender'].apply(lambda x: 1 if(x=='Capital Plus Financial, LLC') else 0)

df1['Is_Benworth_Capital']=df1['lender'].apply(lambda x: 1 if(x=='Benworth Capital') else 0)

df1['BSD_Capital_LLC_dba_Lendistry']=df1['lender'].apply(lambda x: 1 if(x=='BSD Capital, LLC dba Lendistry') else 0)

df1['Harvest_Small_Business Finance_LLC']=df1['lender'].apply(lambda x: 1 if(x=='Harvest Small Business Finance, LLC') else 0)

df1['Is_Customers_Bank']=df1['lender'].apply(lambda x: 1 if(x=='Customers Bank') else 0)

df1['Other_Bank']=df1['lender'].apply(lambda x: 1 if((x!='Prestamos CDFI, LLC') and  (x!='Benworth Capital') and (x!='BSD Capital, LLC dba Lendistry') ar
df1['Status']=df1['Status'].apply(lambda x: 1 if(x=='Removed') else 0)
```

In [ ]:

In [12]:
```
df1.isna().sum(axis=0)
```

Out[12]:
```
name                                  0
amount                                0
zip                                   0
naics_code                            0
NAICS_Num                             0
business_type                         0
jobs_retained                         0
date_approved                         0
lender                                0
loan_status                           0
initial_approval_amount               0
current_approval_amount               0
undisbursed_amount                    0
hubzone_indicator                     0
business_age_description              0
project_county_name                   0
Status                                0
Is_Prestamos_CDFI_LLC                 0
Is_Capital_Plus_Financial_LLC         0
Is_Benworth_Capital                   0
BSD_Capital_LLC_dba_Lendistry         0
Harvest_Small_Business Finance_LLC    0
Is_Customers_Bank                     0
Other_Bank                            0
dtype: int64
```

In [13]:
```
df1.head()
```

Out[13]:

	name	amount	zip	naics_code	NAICS_Num	business_type	jobs_retained	date_approved	lender	loan_status	...	business_age_description	project_county_nar
0	LUMMUS CORPORATION	2000000.0	31408-9586	333249.0	33.0	Corporation	294	2021-02-20	CIBC Bank USA	Exemption 4	...	Existing or more than 2 years old	CHATH/
1	COLIANT SOLUTIONS INC.	1294555.0	30518-9101	541519.0	54.0	Corporation	63	2021-01-31	Synovus Bank	Exemption 4	...	Existing or more than 2 years old	GWINNE
2	YOHE PLUMBING INC	729509.0	30909	238220.0	23.0	Corporation	105	2021-03-12	Customers Bank	Exemption 4	...	Existing or more than 2 years old	RICHMOI
3	LEWIS COLOR LITHOGRAPHERS INC	571193.4	30458-3417	323111.0	32.0	Corporation	49	2021-02-02	Synovus Bank	Exemption 4	...	Existing or more than 2 years old	BULLO(
4	ALMA PAK INTERNATIONAL, LLC	472700.0	31510-4326	445230.0	44.0	Partnership	37	2020-04-10	FNB South	Paid in Full	...	Existing or more than 2 years old	BAC(

5 rows × 24 columns

In [14]:
```
df1['BSD_Capital_LLC_dba_Lendistry'].value_counts()
```

Out[14]:
```
0    551141
1     26022
Name: BSD_Capital_LLC_dba_Lendistry, dtype: int64
```

In [15]:
```
# creating the independent variables list for model building
feature=[feat for feat in df1.columns if feat not in ["Status","lender","name","initial_approval_amount","date_approved","zip","project_county_name"]]

# Subsetting the categorical columns for creating dummies
Cate_columns=['NAICS_Num','loan_status','business_type','hubzone_indicator','business_age_description']
```

In [16]:
```
df1[feature].info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 577163 entries, 0 to 579663
Data columns (total 17 columns):
 #   Column                              Non-Null Count   Dtype
---  ------                              --------------   -----
 0   amount                              577163 non-null  float64
 1   naics_code                          577163 non-null  float64
 2   NAICS_Num                           577163 non-null  float64
 3   business_type                       577163 non-null  object
 4   jobs_retained                       577163 non-null  int64
 5   loan_status                         577163 non-null  object
 6   current_approval_amount             577163 non-null  float64
 7   undisbursed_amount                  577163 non-null  float64
 8   hubzone_indicator                   577163 non-null  object
 9   business_age_description            577163 non-null  object
 10  Is_Prestamos_CDFI_LLC               577163 non-null  int64
 11  Is_Capital_Plus_Financial_LLC       577163 non-null  int64
 12  Is_Benworth_Capital                 577163 non-null  int64
 13  BSD_Capital_LLC_dba_Lendistry       577163 non-null  int64
 14  Harvest_Small_Business Finance_LLC  577163 non-null  int64
 15  Is_Customers_Bank                   577163 non-null  int64
 16  Other_Bank                          577163 non-null  int64
dtypes: float64(5), int64(8), object(4)
memory usage: 79.3+ MB
```

Model Building

Splitting the data into test and train

In [17]:
```
X_train, X_test, y_train, y_test = train_test_split(pd.get_dummies(df1[feature],columns=Cate_columns),df1["Status"],
                                                    test_size=0.3, random_state=20)
```

In [18]:
```
X_test.shape
```

Out[18]:
```
(173149, 72)
```

In [19]:
```
y_test.value_counts()
```

Out[19]:
```
0    165400
1      7749
Name: Status, dtype: int64
```

In [20]:
```
from sklearn import tree
```

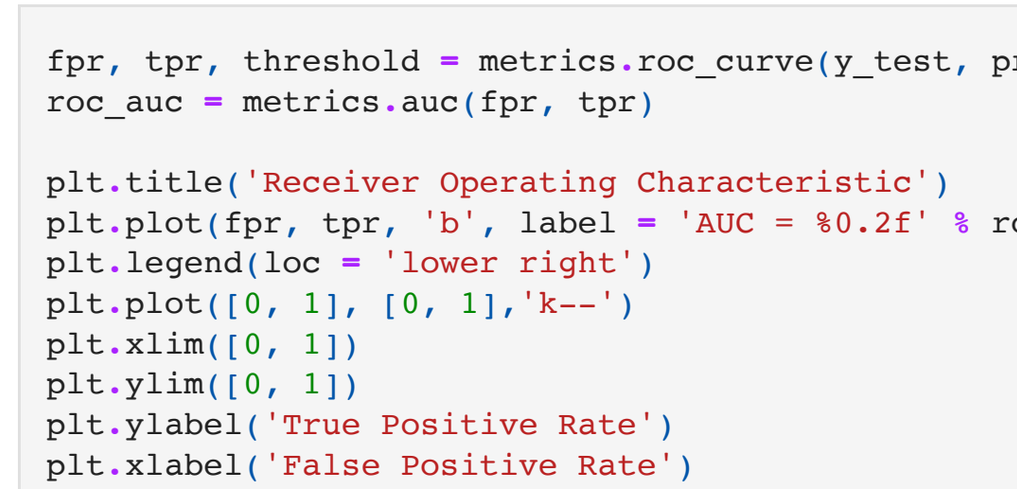
Building a Decision Tree Classifier

In [21]:
```
model=tree.DecisionTreeClassifier()
#model=RandomForestClassifier()
model.fit(X_train,y_train)
predic=model.predict(X_test)
X_predict=model.predict(X_train)
cm_DT=confusion_matrix(y_test,predic)
cm_DT
```

Out[21]:
```
array([[164919,    481],
       [   768,   6981]])
```

In [22]:
```
fpr, tpr, threshold = metrics.roc_curve(y_test, predic)
roc_auc = metrics.auc(fpr, tpr)

plt.title('Receiver Operating Characteristic')
plt.plot(fpr, tpr, 'b', label = 'AUC = %0.2f' % roc_auc)
plt.legend(loc = 'lower right')
plt.plot([0, 1], [0, 1],'k--')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```



In [23]:
```
def cal_accuracy(target_test, y_pred):

    print("Confusion Matrix: ",
        confusion_matrix(target_test, y_pred))

    print ("Accuracy : ",
    accuracy_score(target_test,y_pred)*100)

    print("Report : ",
    classification_report(target_test, y_pred))
```

In [24]:
```
#sen=0.908
cal_accuracy(y_test,predic)
```

```
Confusion Matrix:  [[164919    481]
 [   768   6981]]
Accuracy :  99.27865595527553
Report :                precision    recall  f1-score   support

           0       1.00      1.00      1.00    165400
           1       0.94      0.90      0.92      7749

    accuracy                           0.99    173149
   macro avg       0.97      0.95      0.96    173149
weighted avg       0.99      0.99      0.99    173149
```

Building Random Forest Classifier

In [25]:
```
#model=tree.DecisionTreeClassifier()
model=RandomForestClassifier()
model.fit(X_train,y_train)
predic=model.predict(X_test)
X_predict=model.predict(X_train)
cm_RF=confusion_matrix(y_test,predic)
cm_RF
```

Out[25]:
```
array([[165174,    226],
       [   723,   7026]])
```

In [26]:
```
# sen=0.902
cal_accuracy(y_test,predic)
```

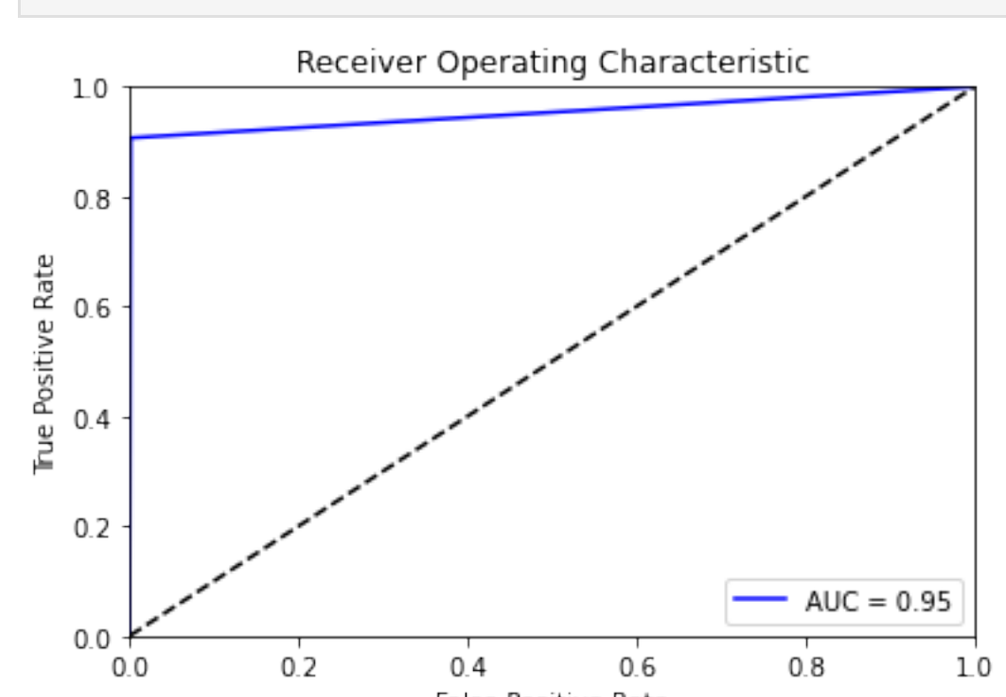
```
Confusion Matrix:  [[165174    226]
 [   723   7026]]
Accuracy :  99.45191713495313
Report :                precision    recall  f1-score   support

           0       1.00      1.00      1.00    165400
           1       0.97      0.91      0.94      7749

    accuracy                           0.99    173149
   macro avg       0.98      0.95      0.97    173149
weighted avg       0.99      0.99      0.99    173149
```

In [27]:
```
fpr, tpr, threshold = metrics.roc_curve(y_test, predic)
roc_auc = metrics.auc(fpr, tpr)

plt.title('Receiver Operating Characteristic')
plt.plot(fpr, tpr, 'b', label = 'AUC = %0.2f' % roc_auc)
plt.legend(loc = 'lower right')
plt.plot([0, 1], [0, 1],'k--')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```



Feature Importance

In [28]:
```
fi=pd.DataFrame({'feature':X_train.columns,
                 'value':model.feature_importances_})
```

In [29]:
```
fi.sort_values('value',ascending=False).head(10)
```

Out[29]:

	feature	value
4	undisbursed_amount	0.389019
37	loan_status_Active Un-Disbursed	0.358369
38	loan_status_Exemption 4	0.076924
39	loan_status_Paid in Full	0.068885
0	amount	0.023457
3	current_approval_amount	0.023241
1	naics_code	0.017005
11	Other_Bank	0.011177
5	Is_Prestamos_CDFI_LLC	0.007006
8	BSD_Capital_LLC_dba_Lendistry	0.003546

Trying to build the model without the features 'loan_status' and 'undisbursed_amount'

In [30]:
```
feature=[feat for feat in df1.columns if feat not in ["Status","lender","name","initial_approval_amount","date_approved","zip","naics_code","undisbursed_

Cate_columns=['NAICS_Num','business_type','hubzone_indicator','business_age_description']
```

In [31]:
```
df1[feature].info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 577163 entries, 0 to 579663
Data columns (total 14 columns):
 #   Column                              Non-Null Count   Dtype
---  ------                              --------------   -----
 0   amount                              577163 non-null  float64
 1   NAICS_Num                           577163 non-null  float64
 2   business_type                       577163 non-null  object
 3   jobs_retained                       577163 non-null  int64
 4   current_approval_amount             577163 non-null  float64
 5   hubzone_indicator                   577163 non-null  object
 6   business_age_description            577163 non-null  object
 7   Is_Prestamos_CDFI_LLC               577163 non-null  int64
 8   Is_Capital_Plus_Financial_LLC       577163 non-null  int64
 9   Is_Benworth_Capital                 577163 non-null  int64
 10  BSD_Capital_LLC_dba_Lendistry       577163 non-null  int64
 11  Harvest_Small_Business Finance_LLC  577163 non-null  int64
 12  Is_Customers_Bank                   577163 non-null  int64
 13  Other_Bank                          577163 non-null  int64
dtypes: float64(3), int64(8), object(3)
memory usage: 66.1+ MB
```

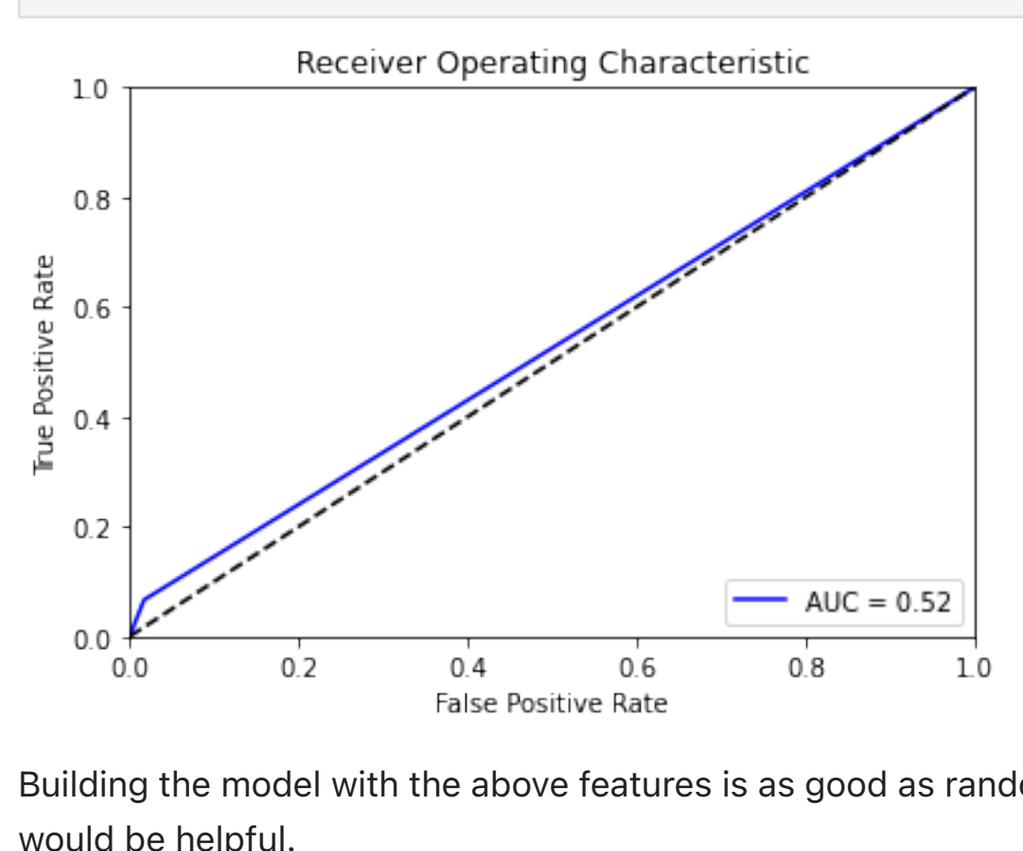
In [32]:
```
X_train, X_test, y_train, y_test = train_test_split(pd.get_dummies(df1[feature],columns=Cate_columns),df1["Status"],
                                                    test_size=0.3, random_state=20)

model=tree.DecisionTreeClassifier()
#model=RandomForestClassifier()
model.fit(X_train,y_train)
predic=model.predict(X_test)
X_predict=model.predict(X_train)
cm_DT=confusion_matrix(y_test,predic)
cm_DT
```

Out[32]:
```
array([[162545,   2855],
       [  7231,    518]])
```

In [33]:
```
fpr, tpr, threshold = metrics.roc_curve(y_test, predic)
roc_auc = metrics.auc(fpr, tpr)

plt.title('Receiver Operating Characteristic')
plt.plot(fpr, tpr, 'b', label = 'AUC = %0.2f' % roc_auc)
plt.legend(loc = 'lower right')
plt.plot([0, 1], [0, 1],'k--')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```



Building the model with the above features is as good as random guessing, to improve the predictability of the model in the future state inclusion of additional demographic indicators would be helpful.