Lab 8: Implement Your Machine Learning Project Plan

In this lab assignment, you will implement the machine learning project plan you created in the written assignment. You will:

1. Load your data set and save it to a Pandas DataFrame.
2. Perform exploratory data analysis on your data to determine which feature engineering and data preparation techniques you will use.
3. Prepare your data for your model and create features and a label.
4. Fit your model to the training data and evaluate your model.
5. Improve your model by performing model selection and/or feature selection techniques to find best model for your problem.

Import Packages

Before you get started, import a few packages.

In [1]:
```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
```

Task: In the code cell below, import additional packages that you have used in this course that you will need for this task.

In [2]:
```
# these are the additional packages that will ne needed for this task
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import plot_roc_curve, accuracy_score, roc_auc_score
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.pipeline import Pipeline
```

Part 1: Load the Data Set

You have chosen to work with one of four data sets. The data sets are located in a folder named "data." The file names of the three data sets are as follows:

- The "adult" data set that contains Census information from 1994 is located in file `adultData.csv`
- The airbnb NYC "listings" data set is located in file `airbnbListingsData.csv`
- The World Happiness Report (WHR) data set is located in file `WHR2018Chapter2OnlineData.csv`
- The book review data set is located in file `bookReviewsData.csv`

Task: In the code cell below, use the same method you have been using to load your data using `pd.read_csv()` and save it to DataFrame `df` .

In [3]:
```
# File names of the book review data set
adultDataSet_filename = os.path.join(os.getcwd(), "data", "adultData.csv")

df = pd.read_csv(adultDataSet_filename, header=0)

df.head()
```

Out[3]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex_selfID	capital-gain	capital-loss	hours-per-week	native-country	income_binary
0	39.0	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Non-Female	2174	0	40.0	United-States	<=50K
1	50.0	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Non-Female	0	0	13.0	United-States	<=50K
2	38.0	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Non-Female	0	0	40.0	United-States	<=50K
3	53.0	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Non-Female	0	0	40.0	United-States	<=50K
4	28.0	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40.0	Cuba	<=50K

Part 2: Exploratory Data Analysis

The next step is to inspect and analyze your data set with your machine learning problem and project plan in mind.

This step will help you determine data preparation and feature engineering techniques you will need to apply to your data to build a balanced modeling data set for your problem and model. These data preparation techniques may include:

- addressing missingness, such as replacing missing values with means
- renaming features and labels
- finding and replacing outliers
- performing winsorization if needed
- performing one-hot encoding on categorical features
- performing vectorization for an NLP problem
- addressing class imbalance in your data sample to promote fair AI

Think of the different techniques you have used to inspect and analyze your data in this course. These include using Pandas to apply data filters, using the Pandas `describe()` method to get insight into key statistics for each column, using the Pandas `dtypes` property to inspect the data type of each column, and using Matplotlib and Seaborn to detect outliers and visualize relationships between features and labels. If you are working on a classification problem, use techniques you have learned to determine if there is class imbalance.

Task: Use the techniques you have learned in this course to inspect and analyze your data.

Note: You can add code cells if needed by going to the **Insert** menu and clicking on **Insert Cell Below** in the drop-drown menu.

In [4]:
```
summary = df.describe()
print(summary)
```

```
                age        fnlwgt  education-num  capital-gain  capital-loss  \
count  32399.000000  3.256100e+04   32561.000000  32561.000000  32561.000000   
mean      38.589216  1.897784e+05      10.080679    615.907773     87.303830   
std       13.647862  1.055500e+05       2.572720   2420.191974    402.960219   
min       17.000000  1.228500e+04       1.000000      0.000000      0.000000   
25%       28.000000  1.178270e+05       9.000000      0.000000      0.000000   
50%       37.000000  1.783560e+05      10.000000      0.000000      0.000000   
75%       48.000000  2.370510e+05      12.000000      0.000000      0.000000   
max       90.000000  1.484705e+06      16.000000  14084.000000   4356.000000   

       hours-per-week  
count    32236.000000  
mean        40.450428  
std         12.353748  
min          1.000000  
25%         40.000000  
50%         40.000000  
75%         45.000000  
max         99.000000  
```

In [5]:
```
# check missingness, find null values
df.isnull().values.any()
nan_count = np.sum(df.isnull(), axis = 0)
nan_count
```

Out[5]:
```
age                162
workclass         1836
fnlwgt               0
education            0
education-num        0
marital-status       0
occupation        1843
relationship         0
race                 0
sex_selfID           0
capital-gain         0
capital-loss         0
hours-per-week     325
native-country     583
income_binary        0
dtype: int64
```

In [7]:

```
# drop columns that will not be necessary
exclude = ['fnlwgt','education-num','education','marital-status','relationship']
newdf = df.drop(columns=exclude, axis=0)

print(newdf)
```

```
        age         workclass         occupation   race  sex_selfID  \
0      39.0         State-gov       Adm-clerical  White  Non-Female
1      50.0  Self-emp-not-inc    Exec-managerial  White  Non-Female
2      38.0           Private  Handlers-cleaners  White  Non-Female
3      53.0           Private  Handlers-cleaners  Black  Non-Female
4      28.0           Private     Prof-specialty  Black      Female
...     ...               ...                ...    ...         ...
32556  27.0           Private       Tech-support  White      Female
32557  40.0           Private  Machine-op-inspct  White  Non-Female
32558  58.0           Private       Adm-clerical  White      Female
32559  22.0           Private       Adm-clerical  White  Non-Female
32560  52.0      Self-emp-inc    Exec-managerial  White      Female

       capital-gain  capital-loss  hours-per-week native-country income_binary
0              2174             0            40.0  United-States         <=50K
1                 0             0            13.0  United-States         <=50K
2                 0             0            40.0  United-States         <=50K
3                 0             0            40.0  United-States         <=50K
4                 0             0            40.0           Cuba         <=50K
...             ...           ...             ...            ...           ...
32556             0             0            38.0  United-States         <=50K
32557             0             0            40.0  United-States          >50K
32558             0             0            40.0  United-States         <=50K
32559             0             0            20.0  United-States         <=50K
32560         14084             0            40.0  United-States          >50K

[32561 rows x 10 columns]
```

In [8]:

```
newdf.head()
```

Out[8]:

	age	workclass	occupation	race	sex_selfID	capital-gain	capital-loss	hours-per-week	native-country	income_binary
0	39.0	State-gov	Adm-clerical	White	Non-Female	2174	0	40.0	United-States	<=50K
1	50.0	Self-emp-not-inc	Exec-managerial	White	Non-Female	0	0	13.0	United-States	<=50K
2	38.0	Private	Handlers-cleaners	White	Non-Female	0	0	40.0	United-States	<=50K
3	53.0	Private	Handlers-cleaners	Black	Non-Female	0	0	40.0	United-States	<=50K
4	28.0	Private	Prof-specialty	Black	Female	0	0	40.0	Cuba	<=50K

Part 3: Implement Your Project Plan

Task: Use the rest of this notebook to carry out your project plan. You will:

1. Prepare your data for your model and create features and a label.
2. Fit your model to the training data and evaluate your model.
3. Improve your model by performing model selection and/or feature selection techniques to find best model for your problem.

Add code cells below and populate the notebook with commentary, code, analyses, results, and figures as you see fit.

Step 1: Data Preparation and Features:

In [11]:

```
# 'income' is the target variable we want to predict

# Separate features (X) and the target variable (y)
X = df.drop('income_binary', axis=1)
y = df['income_binary']

# Perform one-hot encoding on categorical features
X_encoded = pd.get_dummies(X)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_encoded, y, test_size=0.2, random_state=42)
```

In [13]:

```
# Calculate the mean of each column and store it in a variable
mean_values = df.mean()

# Replace missing values in each column with the corresponding mean value
df.fillna(mean_values, inplace=True)
```

Step 2: Model Fitting and Evaluation:

In [15]:

```python
# Initialize and train the Logistic Regression model
lr = LogisticRegression()
lr.fit(X_train, y_train)

# Predict on the test set
y_pred = lr.predict(X_test)

# Evaluate model performance
accuracy = accuracy_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_pred)

print("Accuracy:", accuracy)
print("ROC AUC:", roc_auc)

# Plot ROC curve
plot_roc_curve(lr, X_test, y_test)
plt.title('ROC Curve')
plt.show()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-15-3ce0acd39d13> in <module>()
      1 # Initialize and train the Logistic Regression model
      2 lr = LogisticRegression()
----> 3 lr.fit(X_train, y_train)
      4 
      5 # Predict on the test set

/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/_logistic.py in fit(self, X, y, sample_weight)
   1525 
   1526         X, y = check_X_y(X, y, accept_sparse='csr', dtype=_dtype, order="C",
-> 1527                          accept_large_sparse=solver != 'liblinear')
   1528         check_classification_targets(y)
   1529         self.classes_ = np.unique(y)

/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py in check_X_y(X, y, accept_sparse, accept_large_sparse, dtype, o
rder, copy, force_all_finite, ensure_2d, allow_nd, multi_output, ensure_min_samples, ensure_min_features, y_numeric, warn_on_dtyp
e, estimator)
    753                     ensure_min_features=ensure_min_features,
    754                     warn_on_dtype=warn_on_dtype,
--> 755                     estimator=estimator)
    756     if multi_output:
    757         y = check_array(y, 'csr', force_all_finite=True, ensure_2d=False,

/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py in check_array(array, accept_sparse, accept_large_sparse, dtyp
e, order, copy, force_all_finite, ensure_2d, allow_nd, ensure_min_samples, ensure_min_features, warn_on_dtype, estimator)
    576         if force_all_finite:
    577             _assert_all_finite(array,
--> 578                                allow_nan=force_all_finite == 'allow-nan')
    579 
    580     if ensure_min_samples > 0:

/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py in _assert_all_finite(X, allow_nan, msg_dtype)
     58                     msg_err.format
     59                     (type_err,
---> 60                      msg_dtype if msg_dtype is not None else X.dtype)
     61             )
     62     # for object dtype data, we only check for NaNs (GH-13254)

ValueError: Input contains NaN, infinity or a value too large for dtype('float64').
```

In [16]:

```
param_grid = {
    'C': [0.001, 0.01, 0.1, 1, 10],
    'penalty': ['l1', 'l2']
}

grid_search = GridSearchCV(lr, param_grid, cv=5, scoring='roc_auc')
grid_search.fit(X_train, y_train)

# Get the best model and its parameters
best_logreg = grid_search.best_estimator_
best_params = grid_search.best_params_

# Re-train the best model on the entire training set
best_logreg.fit(X_train, y_train)

# Evaluate the improved model
y_pred_best = best_logreg.predict(X_test)
accuracy_best = accuracy_score(y_test, y_pred_best)
roc_auc_best = roc_auc_score(y_test, y_pred_best)

print("Best Model Accuracy:", accuracy_best)
print("Best Model ROC AUC:", roc_auc_best)

# Plot ROC curve for the best model
plot_roc_curve(best_logreg, X_test, y_test)
plt.title('ROC Curve (Best Model)')
plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penalty.

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penalty.

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penalty.

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penalty.

  FitFailedWarning)
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penalty.

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_validation.py:536: FitFailedWarning: Estimator fit failed. The sco
re on this train-test partition for these parameters will be set to nan. Details:
ValueError: Input contains NaN, infinity or a value too large for dtype('float64').

  FitFailedWarning)

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-16-53bc0792f786> in <module>()
      5 
      6 grid_search = GridSearchCV(lr, param_grid, cv=5, scoring='roc_auc')
----> 7 grid_search.fit(X_train, y_train)
      8 
      9 # Get the best model and its parameters

/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_search.py in fit(self, X, y, groups, **fit_params)
    737             refit_start_time = time.time()
    738             if y is not None:
--> 739                 self.best_estimator_.fit(X, y, **fit_params)
    740             else:
    741                 self.best_estimator_.fit(X, **fit_params)

/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/_logistic.py in fit(self, X, y, sample_weight)
   1486         The SAGA solver supports both float64 and float32 bit arrays.
   1487         """
-> 1488         solver = _check_solver(self.solver, self.penalty, self.dual)
   1489 
   1490         if not isinstance(self.C, numbers.Number) or self.C < 0:

/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/_logistic.py in _check_solver(solver, penalty, dual)
    443     if solver not in ['liblinear', 'saga'] and penalty not in ('l2', 'none'):
    444         raise ValueError("Solver %s supports only 'l2' or 'none' penalties, "
--> 445                          "got %s penalty." % (solver, penalty))
    446     if solver != 'liblinear' and dual:
    447         raise ValueError("Solver %s supports only "

ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penalty.
```

```
In [ ]:
```