ImplementMLProjectPlan

August 17, 2023

1 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.

1.0.1 Import Packages

Before you get started, import a few packages.

[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.

[54]:
```
#  import the scikit-learn LogisticRegression, the train_test_split() function␣
↪for splitting the data into training and test sets, and the function␣
↪roc_auc_score to evaluate the model.
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import roc_auc_score
from sklearn.feature_extraction.text import TfidfVectorizer
import gensim
from gensim.utils import simple_preprocess
from sklearn.preprocessing import LabelEncoder
from sklearn.svm import SVC
from sklearn.metrics import classification_report, accuracy_score
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from collections import Counter
import matplotlib.pyplot as plt
```

1.1 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`.

[3]:
```
# YOUR CODE HERE
bookReviewDataSet_filename = os.path.join(os.getcwd(), "data", "bookReviewsData.
 ↪csv")

df = pd.read_csv(bookReviewDataSet_filename)

df
```

[3]:
```
                                                Review  Positive Review
0     This was perhaps the best of Johannes Steinhof...             True
1     This very fascinating book is a story written ...             True
2     The four tales in this collection are beautifu...             True
3     The book contained more profanity than I expec...            False
4     We have now entered a second time of deep conc...             True
...                                                 ...              ...
1968  I purchased the book with the intention of tea...             True
1969  There are so many design books, but the Graphi...             True
1970  I am thilled to see this book being available ...             True
1971  As many have stated before me the book starts ...            False
1972  I love this book! It is a terrific blend of ha...             True

[1973 rows x 2 columns]
```

1.2 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.

[4]:
```
#data exploration:
#printing out the first few lines
df.head()
```

[4]:
```
                                              Review  Positive Review
0  This was perhaps the best of Johannes Steinhof...             True
1  This very fascinating book is a story written ...             True
2  The four tales in this collection are beautifu...             True
3  The book contained more profanity than I expec...            False
4  We have now entered a second time of deep conc...             True
```

[5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1973 entries, 0 to 1972
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Review           1973 non-null   object
 1   Positive Review  1973 non-null   bool
dtypes: bool(1), object(1)
memory usage: 17.5+ KB
```

1.3 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.

[6]:
```
df.describe()
```

[6]:
```
                                                   Review Positive Review
count                                                1973            1973
unique                                               1865               2
top     I have read several of Hiaasen's books and lov...           False
freq                                                    3             993
```

[7]:
```
df
```

[7]:
```
                                                Review  Positive Review
0     This was perhaps the best of Johannes Steinhof...             True
1     This very fascinating book is a story written ...             True
2     The four tales in this collection are beautifu...             True
3     The book contained more profanity than I expec...            False
4     We have now entered a second time of deep conc...             True
...                                                 ...              ...
1968  I purchased the book with the intention of tea...             True
1969  There are so many design books, but the Graphi...             True
1970  I am thilled to see this book being available ...             True
1971  As many have stated before me the book starts ...            False
1972  I love this book! It is a terrific blend of ha...             True

[1973 rows x 2 columns]
```

[8]:
```
class_distribution = df['Positive Review'].value_counts()
print(class_distribution)

#since there is around the same number of false/true reviews, it can be␣
 ↪concluded that there is NOT!!!an imbalanced data sample.
```

```
False    993
True     980
Name: Positive Review, dtype: int64
```

[9]:
```
y = df['Positive Review']
X = df['Review']

X.shape
```

[9]:
```
(1973,)
```

[10]:
```
X.head()
```

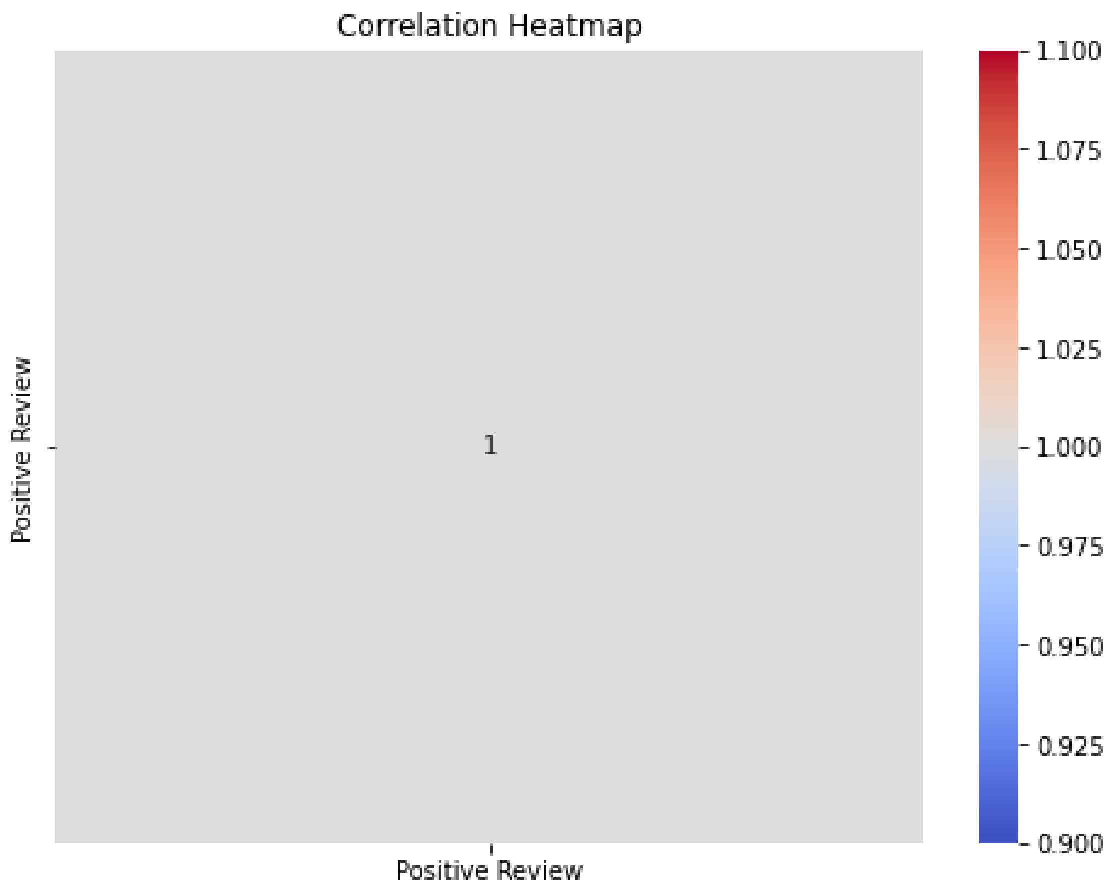
[10]:
```
0    This was perhaps the best of Johannes Steinhof...
1    This very fascinating book is a story written ...
2    The four tales in this collection are beautifu...
3    The book contained more profanity than I expec...
4    We have now entered a second time of deep conc...
Name: Review, dtype: object
```

[14]:
```
# Check for missing values
print(df.isnull().sum())
```

```
#when you run the program, there are no missing values
```

```
Review             0
Positive Review    0
dtype: int64
```

[19]:
```
#correlation heatmap

corr_matrix = df.corr()
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



[26]:
```
# Text preprocessing and vectorization
df['Review'] = df['Review'].str.lower()  # Convert to lowercase
df['Review'] = df['Review'].str.replace('[^\w\s]', '')  # Remove punctuation

tfidf_vectorizer = TfidfVectorizer(max_features=5000, stop_words='english')
```

```
X = tfidf_vectorizer.fit_transform(df['Review'])

# Encoding labels
label_encoder = LabelEncoder()
df['Encoded_Labels'] = label_encoder.fit_transform(df['Positive Review'])

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, df['Encoded_Labels'],␣
↪test_size=0.2, random_state=42)
```

[27]:
```
df
```

[27]:
```
                                                 Review  Positive Review  \
0     this was perhaps the best of johannes steinhof...             True
1     this very fascinating book is a story written ...             True
2     the four tales in this collection are beautifu...             True
3     the book contained more profanity than i expec...            False
4     we have now entered a second time of deep conc...             True
...                                                 ...              ...
1968  i purchased the book with the intention of tea...             True
1969  there are so many design books but the graphis...             True
1970  i am thilled to see this book being available ...             True
1971  as many have stated before me the book starts ...            False
1972  i love this book it is a terrific blend of han...             True

      Encoded_Labels
0                  1
1                  1
2                  1
3                  0
4                  1
...              ...
1968               1
1969               1
1970               1
1971               0
1972               1

[1973 rows x 3 columns]
```

[67]:
```
# remove the stopwords from the reviews

# Define your list of stopwords
stop_words = {
    'the', 'thee', 'i', 'book', 'was', 'not', 'but', 'are', 'you', 'have',␣
↪'be', 'he', 'his', 'or',
    'about', 'one', 'an', 'all', 'read', 'if', 'my', 'of', 'and', 'a', 'in',␣
↪'to', 'is', 'it', 'this',
    'that', 'for', 'with', 'on', 'at', 'by', 'from', 'as', 'we', 'so', 'will',␣
 ↪'can', 'your', 'our',
    'their', 'there', 'they', 'which', 'who', 'what', 'when', 'where', 'why',␣
 ↪'how', 'while', 'also',
    'than', 'more', 'much', 'over', 'under', 'through', 'between', 'among',␣
 ↪'before', 'after', 'during',
    'since', 'until', 'while', 'very', 'too', 'such', 'just', 'only', 'both',␣
 ↪'neither', 'either', 'some',
    'any', 'no', 'none', 'each', 'every', 'all', 'most', 'few', 'many', 'some',␣
 ↪'other', 'another',
    'has', 'her', 'its', 'like', 'would', 'me', 'out', 'she', 'do', 'had',␣
 ↪'up', 'even', 'into', 'people',
    'were', 'story'
}

# Remove stopwords from the reviews in the dataframe
def remove_stopwords(review):
    words = review.split()
    meaningful_words = [word for word in words if word.lower() not in␣
 ↪stop_words]
    return ' '.join(meaningful_words)

# Apply the remove_stopwords function to the 'Review' column
df['Review'] = df['Review'].apply(remove_stopwords)

# Now the 'Review' column in the dataframe contains reviews with stopwords␣
 ↪removed
print(df['Review'])
```

```
0       perhaps best johannes steinhoffs books does de...
1       fascinating written form numerous letters pers...
2       four tales collection beautifully composed art...
3       contained profanity expected rita rudner expec...
4       now entered second time deep concern science m...
                              ...
1968    purchased intention teaching myself core mater...
1969    design books graphis series always asure best ...
1970    am thilled see being available hardcover paper...
1971    stated starts off great promise historical fic...
1972    love terrific blend handholding concrete advic...
Name: Review, Length: 1973, dtype: object
```

[30]:
```
# Initialize and train the SVM classifier
svm_classifier = SVC(kernel='linear', random_state=42)
svm_classifier.fit(X_train, y_train)

# Make predictions on the test set
```

```
y_pred = svm_classifier.predict(X_test)

# Convert boolean labels to string labels for classification_report
class_names = ['Negative Review', 'Positive Review']
y_test_str = np.array([class_names[label] for label in y_test])
y_pred_str = np.array([class_names[label] for label in y_pred])

# Evaluate the model
accuracy = accuracy_score(y_test_str, y_pred_str)
classification_rep = classification_report(y_test_str, y_pred_str,␣
 ↪target_names=class_names)

print(f'Accuracy: {accuracy:.2f}')
print('Classification Report:\n', classification_rep)
```

```
Accuracy: 0.82
Classification Report:
                  precision    recall  f1-score   support

Negative Review       0.83      0.81      0.82       195
Positive Review       0.82      0.84      0.83       200

       accuracy                           0.82       395
      macro avg       0.82      0.82      0.82       395
   weighted avg       0.82      0.82      0.82       395
```

[31]:
```
#how to further improve and optimize model's performance:
```

[35]:
```
#hyperparameter tuning:

# Define the parameter grid to search
param_grid = {
    'C': [0.1, 1, 10],
    'kernel': ['linear', 'rbf'],
}

# Initialize the SVM classifier
svm_classifier = SVC(random_state=42)

# Perform grid search with cross-validation
grid_search = GridSearchCV(svm_classifier, param_grid, cv=5)
grid_search.fit(X_train, y_train)

# Print the best hyperparameters
print("Best hyperparameters:", grid_search.best_params_)

# Evaluate the model with the best hyperparameters
```

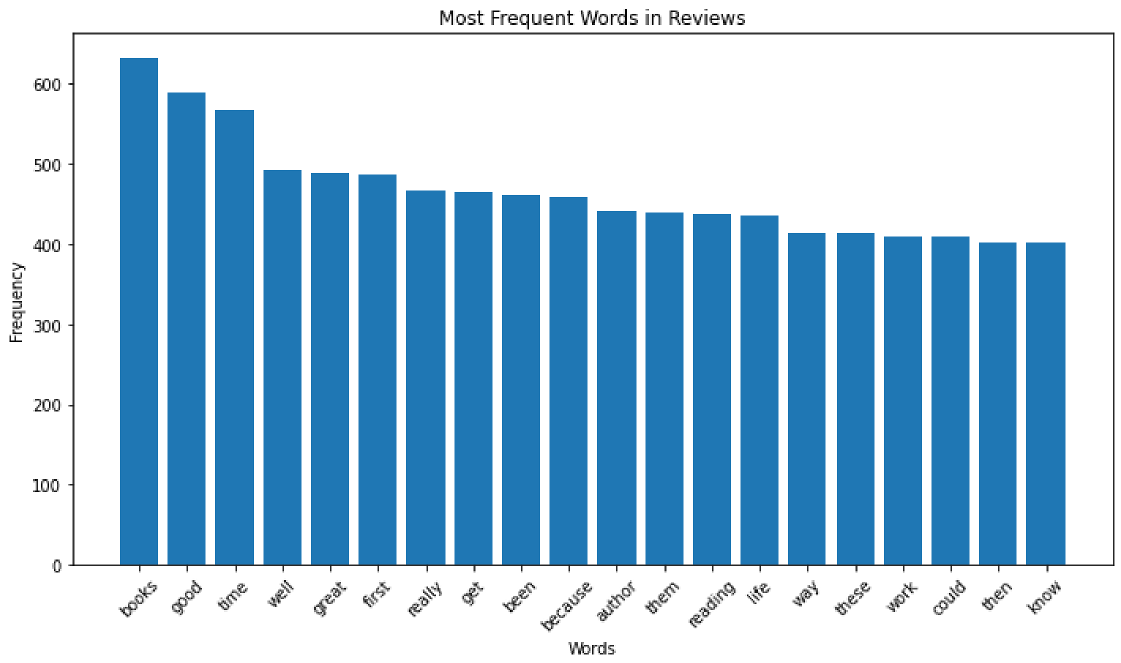
```
best_svm = grid_search.best_estimator_
y_pred = best_svm.predict(X_test)

# Print accuracy and classification report
accuracy = accuracy_score(y_test_str, y_pred_str)
classification_rep = classification_report(y_test_str, y_pred_str,␣
 ↪target_names=class_names)

print(f'Accuracy: {accuracy:.2f}')
print('Classification Report:\n', classification_rep)
```

```
Best hyperparameters: {'C': 10, 'kernel': 'rbf'}
Accuracy: 0.82
Classification Report:
                  precision    recall  f1-score   support

Negative Review       0.83      0.81      0.82       195
Positive Review       0.82      0.84      0.83       200

       accuracy                           0.82       395
      macro avg       0.82      0.82      0.82       395
   weighted avg       0.82      0.82      0.82       395
```

[39]:
```
#we can also try different models to fine tune the model:

# Initialize the Random Forest classifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)

# Train the Random Forest classifier
rf_classifier.fit(X_train, y_train)

# Make predictions
y_pred_rf = rf_classifier.predict(X_test)

# Convert numerical labels to string labels for classification report
y_test_str_rf = np.array([class_names[label] for label in y_test])
y_pred_str_rf = np.array([class_names[label] for label in y_pred_rf])

# Evaluate the Random Forest model
accuracy_rf = accuracy_score(y_test_str_rf, y_pred_str_rf)
classification_rep_rf = classification_report(y_test_str_rf, y_pred_str_rf,␣
 ↪target_names=class_names)

print(f'Random Forest Accuracy: {accuracy_rf:.2f}')
print('Random Forest Classification Report:\n', classification_rep_rf)
```

```
Random Forest Accuracy: 0.77
Random Forest Classification Report:
                  precision    recall  f1-score   support

Negative Review       0.76      0.79      0.77       195
Positive Review       0.79      0.76      0.77       200

       accuracy                           0.77       395
      macro avg       0.77      0.77      0.77       395
   weighted avg       0.77      0.77      0.77       395
```

[68]:
```
#Create a visualization of the most frequently used words:

# Combine all the reviews into a single string
all_reviews = ' '.join(df['Review'])

# Tokenize the text into words
words = all_reviews.split()

# Calculate the frequency of each word
word_freq = Counter(words)

# Get the most common words and their frequencies
top_words = word_freq.most_common(20)  # Change the number as needed

# Separate words and frequencies for plotting
labels, counts = zip(*top_words)

# Create a bar chart
plt.figure(figsize=(10, 6))
plt.bar(labels, counts)
plt.xticks(rotation=45)
plt.xlabel('Words')
plt.ylabel('Frequency')
plt.title('Most Frequent Words in Reviews')
plt.tight_layout()

#display the plot
plt.show()
```


Most Frequent Words in Reviews
Frequency
Words
0
100
200
300
400
500
600
books
good
time
well
great
first
really
get
been
because
author
them
reading
life
way
these
work
could
then
know

[ ]: