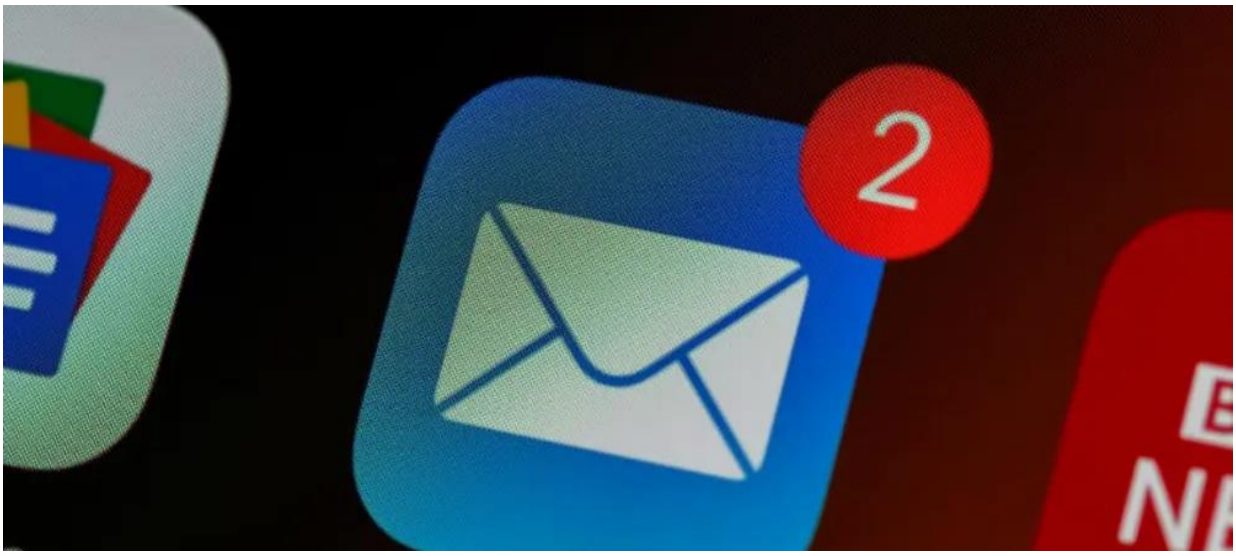
BUILDING A SMARTER AI-POWERED SPAM CLASSIFIER

Phase 3 Submission Document

Project Title : Development Part 1

Topic: section begin building your project by loading and preprocessing the dataset



INTRODUCTION:

In the realm of data-driven projects, success often hinges on the quality and readiness of the dataset under examination. Loading and preprocessing this data is a foundational step, setting the stage for robust analysis, modeling, and decision-making. In this section, we will delve into the critical processes of acquiring, loading, and preparing the dataset for our project.

- **Dataset Overview:** We will begin by providing a brief overview of the dataset under investigation. This includes its source, the context in which it was collected, and the primary objective of its utilization within the project.

- **Data Acquisition:** This section will discuss the methods employed to obtain the dataset. It may include data collection procedures, sources, and any ethical considerations associated with data gathering.

- **Data Loading:** Loading the dataset into our analysis environment is a pivotal task. We will discuss the tools and techniques used for importing the data, whether it be from a database, CSV file, API, or other sources.

- **Data Preprocessing:** Raw data seldom arrives in the perfect format for analysis. This subsection will cover data preprocessing steps such as handling missing values, dealing with outliers, and converting data types to ensure it is ready for analytical tasks.

- **Data Exploration:** While primarily an exploratory process, this phase is crucial in identifying initial patterns and trends within the data, which may inform subsequent project directions.

- **Data Quality Assurance:** Quality control is integral to ensuring the integrity of the dataset. We will discuss measures taken to validate and clean the data, maintaining its accuracy and reliability.

DATASET :

v1	v2
ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there go
ham	Ok lar... Joking wif u oni...
spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to rece
ham	U dun say so early hor... U c already then say...
ham	Nah I don't think he goes to usf, he lives around here though
spam	FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you u
ham	Even my brother is not like to speak with me. They treat me like aids patent.
ham	As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as y
spam	WINNER!! As a valued network customer you have been selected to receivea å£900 prize re
spam	Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles wi
ham	I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cr
spam	SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575. Cost 1
spam	URGENT! You have won a 1 week FREE membership in our å£100,000 Prize Jackpot! Txt the
ham	I've been searching for the right words to thank you for this breather. I promise i wont take
ham	I HAVE A DATE ON SUNDAY WITH WILL!!
spam	XXXMobileMovieClub: To use your credit, click the WAP link in the next txt message or click
ham	Oh k...i'm watching here:)
ham	Eh u remember how 2 spell his name... Yes i did. He v naughty make until i v wet.
ham	Fine if thatåÕs the way u feel. ThatåÕs the way its gota b
spam	England v Macedonia - dont miss the goals/team news. Txt ur national team to 87077 eg EN
ham	Is that seriously how you spell his name?
ham	I‰Û÷m going to try for 2 months ha ha only joking
ham	So Ì_ pay first lar... Then when is da stock comin...
ham	Aft i finish my lunch then i go str down lor. Ard 3 smth lor. U finish ur lunch already?
ham	Ffffffffff. Alright no way I can meet up with you sooner?
ham	Just forced myself to eat a slice. I'm really not hungry tho. This sucks. Mark is getting worrie
ham	Lol your always so convincing.
ham	Did you catch the bus ? Are you frying an egg ? Did you make a tea? Are you eating your mo
ham	I'm back & we're packing the car now, I'll let you know if there's room
ham	Ahhh. Work. I vaguely remember that! What does it feel like? Lol
ham	Wait that's still not all that clear, were you not sure about me being sarcastic or that that's v
ham	Yeah he got in at 2 and was v apologetic. n had fallen out and she was actin like spoilt child a
ham	K tell me anything about you.
ham	For fear of fainting with the of all that housework you just did? Quick have a cuppa
spam	Thanks for your subscription to Ringtone UK your mobile will be charged å£5/month Please
ham	Yup... Ok i go home look at the timings then i msg Ì_ again... Xuhui going to learn on 2nd ma
ham	Oops, I'll let you know when my roommate's done
ham	I see the letter B on my car
ham	Anything lor... U decide...
ham	Hello! How's you and how did saturday go? I was just texting to see if you'd decided to do a
ham	Pls go ahead with watts. I just wanted to be sure. Do have a great weekend. Abiola
ham	Did I forget to tell you ? I want you , I need you, I crave you ... But most of all ... I love you m
spam	07732584351 - Rodger Burns - MSG = We tried to call you re your reply to our sms for a free
ham	WHO ARE YOU SEEING?
ham	Great! I hope you like your man well endowed. I am <#> inches...
ham	No calls..messages..missed calls

Context:

The SMS Spam Collection is a set of SMS tagged messages that have been collected for SMS Spam research. It contains one set of SMS messages in English of 5,574 messages, tagged acording being ham (legitimate) or spam.

Content:

The files contain one message per line. Each line is composed by two columns: v1 contains the label (ham or spam) and v2 contains the raw text.

This corpus has been collected from free or free for research sources at the Internet:

- ✓ A collection of 425 SMS spam messages was manually extracted from the Grumbletext Web site. This is a UK forum in which cell phone users make public claims about SMS spam messages, most of them without reporting the very spam message received. The identification of the text of spam messages in the claims is a very hard and time-consuming task, and it involved carefully scanning hundreds of web pages.
- ✓ A subset of 3,375 SMS randomly chosen ham messages of the NUS SMS Corpus (NSC), which is a dataset of about 10,000 legitimate messages collected for research at the Department of Computer Science at the National University of Singapore. The messages largely originate from Singaporeans and mostly from students attending the University. These messages were collected from volunteers who were made aware that their contributions were going to be made publicly available.
- ✓ A list of 450 SMS ham messages collected from Caroline Tag's PhD Thesis available.
- ✓ Finally, we have incorporated the SMS Spam Corpus v.0.1 Big. It has 1,002 SMS ham messages and 322 spam messages and it is public available.
- ✓ This is an automatically-generated kernel with starter code demonstrating how to read in the data and begin exploring. Click the blue "Edit Notebook" or "Fork Notebook" button at the top of this kernel to begin editing.

Acknowledgements:

The original dataset can be found here. The creators would like to note that in case you find the dataset useful, please make a reference to previous paper and the web page: http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/ in your papers, research, etc. We offer a comprehensive study of this corpus in the following paper. This work presents a number of statistics, studies and baseline results for several machine learning methods.

Exploratory Analysis:

To begin this exploratory analysis, first use `matplotlib` to import libraries and define functions for plotting the data. Depending on the data, not all plots will be made. (Hey, I'm just a kerneling bot, not a Kaggle Competitions Grandmaster!)

In[1]:

```
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt # plotting
import numpy as np # linear algebra
import os # accessing directory structure
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```

There is 1 csv file in the current version of the dataset:

In[2]:

```
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

The next hidden code cells define functions for plotting data. Click on the "Code" button in the publis hed kernel to reveal the hidden code.

In[3]:

```
# Distribution graphs (histogram/bar graph) of column data
def plotPerColumnDistribution(df, nGraphShown, nGraphPerRow):
    nunique = df.nunique()
    df = df[[col for col in df if nunique[col] > 1 and nunique[col] < 50]] # For
displaying purposes, pick columns that have between 1 and 50 unique values
    nRow, nCol = df.shape
    columnNames = list(df)
    nGraphRow = (nCol + nGraphPerRow - 1) / nGraphPerRow
    plt.figure(num = None, figsize = (6 * nGraphPerRow, 8 * nGraphRow), dpi = 80,
facecolor = 'w', edgecolor = 'k')
    for i in range(min(nCol, nGraphShown)):
        plt.subplot(nGraphRow, nGraphPerRow, i + 1)
        columnDf = df.iloc[:, i]
        if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):
            valueCounts = columnDf.value_counts()
            valueCounts.plot.bar()
        else:
            columnDf.hist()
        plt.ylabel('counts')
        plt.xticks(rotation = 90)
        plt.title(f'{columnNames[i]} (column {i})')
    plt.tight_layout(pad = 1.0, w_pad = 1.0, h_pad = 1.0)
    plt.show()
```

ln[4]:

```
# Correlation matrix
def plotCorrelationMatrix(df, graphWidth):
    filename = df.dataframeName
```

```
    df = df.dropna('columns') # drop columns with NaN
    df = df[[col for col in df if df[col].nunique() > 1]] # keep columns where th
ere are more than 1 unique values
    if df.shape[1] < 2:
        print(f'No correlation plots shown: The number of non-NaN or constant col
umns ({df.shape[1]}) is less than 2')
        return
    corr = df.corr()
    plt.figure(num=None, figsize=(graphWidth, graphWidth), dpi=80, facecolor='w',
edgecolor='k')
    corrMat = plt.matshow(corr, fignum = 1)
    plt.xticks(range(len(corr.columns)), corr.columns, rotation=90)
    plt.yticks(range(len(corr.columns)), corr.columns)
    plt.gca().xaxis.tick_bottom()
    plt.colorbar(corrMat)
    plt.title(f'Correlation Matrix for {filename}', fontsize=15)
    plt.show()

ln[5]:

# Scatter and density plots
def plotScatterMatrix(df, plotSize, textSize):
    df = df.select_dtypes(include =[np.number]) # keep only numerical columns
    # Remove rows and columns that would lead to df being singular
    df = df.dropna('columns')
    df = df[[col for col in df if df[col].nunique() > 1]] # keep columns where th
ere are more than 1 unique values
    columnNames = list(df)
    if len(columnNames) > 10: # reduce the number of columns for matrix inversion
of kernel density plots
        columnNames = columnNames[:10]
    df = df[columnNames]
    ax = pd.plotting.scatter_matrix(df, alpha=0.75, figsize=[plotSize, plotSize],
diagonal='kde')
    corrs = df.corr().values
    for i, j in zip(*plt.np.triu_indices_from(ax, k = 1)):
        ax[i, j].annotate('Corr. coef = %.3f' % corrs[i, j], (0.8, 0.2), xycoords
='axes fraction', ha='center', va='center', size=textSize)
    plt.suptitle('Scatter and Density Plot')
    plt.show()
```

Now you're ready to read in the data and use the plotting functions to visualize the data.

```
ln[6]:

nRowsRead = 1000 # specify 'None' if want to read whole file
# spam.csv has 5572 rows in reality, but we are only loading/previewing the first
1000 rows
df1 = pd.read_csv('/kaggle/input/spam.csv', delimiter=',', nrows = nRowsRead)
df1.dataframeName = 'spam.csv'
nRow, nCol = df1.shape
print(f'There are {nRow} rows and {nCol} columns')
```

```
---------------------------------------------------------------------------
UnicodeDecodeError                        Traceback (most recent call last)
pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader._convert_tokens()

pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader._convert_with_dtype()

pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader._string_convert()

pandas/_libs/parsers.pyx in pandas._libs.parsers._string_box_utf8()

UnicodeDecodeError: 'utf-8' codec can't decode bytes in position 135-136: invalid
continuation byte

During handling of the above exception, another exception occurred:

UnicodeDecodeError                        Traceback (most recent call last)
<ipython-input-6-556be88e201a> in <module>
      1 nRowsRead = 1000 # specify 'None' if want to read whole file
      2 # spam.csv has 5572 rows in reality, but we are only loading/previewing t
he first 1000 rows
----> 3 df1 = pd.read_csv('/kaggle/input/spam.csv', delimiter=',', nrows = nRowsR
ead)
      4 df1.dataframeName = 'spam.csv'
      5 nRow, nCol = df1.shape

/opt/conda/lib/python3.6/site-packages/pandas/io/parsers.py in parser_f(filepath_
or_buffer, sep, delimiter, header, names, index_col, usecols, squeeze, prefix, ma
ngle_dupe_cols, dtype, engine, converters, true_values, false_values, skipinitial
space, skiprows, skipfooter, nrows, na_values, keep_default_na, na_filter, verbos
e, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_pars
er, dayfirst, cache_dates, iterator, chunksize, compression, thousands, decimal,
lineterminator, quotechar, quoting, doublequote, escapechar, comment, encoding, d
ialect, error_bad_lines, warn_bad_lines, delim_whitespace, low_memory, memory_map
, float_precision)
    683         )
    684
--> 685         return _read(filepath_or_buffer, kwds)
    686
    687     parser_f.__name__ = name

/opt/conda/lib/python3.6/site-packages/pandas/io/parsers.py in _read(filepath_or_
buffer, kwds)
    461
    462     try:
--> 463         data = parser.read(nrows)
    464     finally:
    465         parser.close()

/opt/conda/lib/python3.6/site-packages/pandas/io/parsers.py in read(self, nrows)
   1152     def read(self, nrows=None):
   1153         nrows = _validate_integer("nrows", nrows)
```

```
-> 1154         ret = self._engine.read(nrows)
   1155
   1156         # May alter columns / col_dict

/opt/conda/lib/python3.6/site-packages/pandas/io/parsers.py in read(self, nrows)
   2046     def read(self, nrows=None):
   2047         try:
-> 2048             data = self._reader.read(nrows)
   2049         except StopIteration:
   2050             if self._first_chunk:

pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader.read()

pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader._read_low_memory()

pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader._read_rows()

pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader._convert_column_data(
)

pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader._convert_tokens()

pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader._convert_with_dtype()

pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader._string_convert()

pandas/_libs/parsers.pyx in pandas._libs.parsers._string_box_utf8()

UnicodeDecodeError: 'utf-8' codec can't decode bytes in position 135-136: invalid
continuation byte
```

Let's take a quick look at what the data looks like:

```
ln[7]:

df1.head(5)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-7-e55bb665ba13> in <module>
----> 1 df1.head(5)

NameError: name 'df1' is not defined
```

Distribution graphs (histogram/bar graph) of sampled columns:

```
ln[8]:

plotPerColumnDistribution(df1, 10, 5)
```

```
---------------------------------------------------------------------------
```

```
NameError                                 Traceback (most recent call last)
<ipython-input-8-a0a199b2d778> in <module>
----> 1 plotPerColumnDistribution(df1, 10, 5)

NameError: name 'df1' is not defined
```

Conclusion:

This concludes your starter analysis! To go forward from here, click the blue "Edit Notebook" button at the top of the kernel. This will create a copy of the code and environment for you to edit. Delete, modify, and add code as you please. Happy Kaggling!