【TechGym】ゼロからはじめる機械学習入門講座「教師なし学習と自然言語処理入門」(テックジムオープン講座)
サンプルソースの公開場所 https://github.com/techgymjp/techgym_ai
実行環境がない場合は anaconda を install してください

■1-2：クラスタリング：G9xs.py
【問題】
分類データセット生成の関数を使ってデータを生成してクラスタリングをしてみよう。
(1) make_blobs 関数で random_state=5 として、生成したデータを散布図に書く。
■解答は Y7AM.py

(2)クラスタリングをしてクラスタ番号でデータを色分けする。
■解答は F9MH.py

■1-5：主成分分析(PCA)：Q3SR.py
分析対象のデータには多くの変数があり、一つ一つの変数間の関係を調べていくには大変な場合がある。そこで、変数の種類を圧縮しつつ元のデータの性質を残して分析する手法として、主成分分析(Principle Component Analysis)がある。また、数値のもつ意味も変数によって異なることがあり、平均値や標準偏差が異なる場合には単純に数値を比較することが出来ない。そこで、標準化という変換をする。こうすることで、平均値が 0 で標準偏差が 1 になり数値の比較が出来るようになる。

【問題】
◎サンプルデータを標準化してグラフ化してみよう。
さらに sp.stats.pearsonr によって相関係数を計算してみる。
■解答は Q9YY.py

■1-6：主成分分析(PCA)：E8SS.py
組み込みデータ(アヤメの計測データ)を用いて主成分分析を実行してみよう。アヤメの計測データの特徴量を 3 次元でプロットする。標準化して主成分分析(PCA)を実行する。主成分分析(PCA)を実行した前後でデータの次元が変化したかを確認する。第 1 成分と第 2 成分と縦軸と横軸にしてデータをプロットする。
■解答は T3JE.py

■1-8：形態素解析：JB65.py
自然言語処理をするときに、文章を分解して内容を判断する必要がある。このときに使われるのが形態素解析で、単語の最小単位に分割して品詞の種類や活用を分類する。ここでは Python で使える形態素解析ツールである「Janome」で解析をしてみる。以下のコマンドでインストールすることが出来る。
$ pip install janome

【問題】
(1) 例文'すももももももものうち'を形態素解析する。
■解答は FT5J.py

(2) 形態素解析をして基本形と品詞のみを表示する。
(Token オブジェクトの属性である token.base_form と token.part_of_speech によって表示できる)
■解答は MJ8E.py

(3)ファイル“techgym-AI.txt”から文章を読み込んで形態素解析をする。(このとき文章全体も表示してみる)
■解答は CT2P.py

■1-9：word2vec：SY7J.py
文章を形態素に分割することは出来たが、そのままでは文章の意味を扱う事ができない。そこで単語にベクトルを割り当てて数値化することで意味を持たせて扱いやすくする。word2vec はベクトル表現のための手法の一つで、抽象的な意味を表現出来るようなベクトルとして単語を扱える。
word2vec のモデルを作成するライブラリとして gensim を用いる。インストールは以下を実行する。

```
$ pip install gensim
```

この方法を使うメリットとして、単語同士の関係性を計算することが出来るようになる。
例えば、
「王様」-「男」+「女」=「女王」
「パリ」-「フランス」+「日本」=「東京」
のように、単語同士の意味を足したり引いたりすることが可能になる。
また、単語同士の意味の近さも計算出来るようになる。

◎与えられた単語を使用してモデルをつくって単語のベクトル表現とベクトルの次元を表示してみる。
■解答は I2NF.py
◎「猫」と「犬」のコサイン類似度を求める。
■解答は WG9Z.py

■1-11：word2vec：S6EY.py
さらに大きな文章で単語同士の関係性を確認してみる。青空文庫(https://www.aozora.gr.jp/)に小説があるので、これを使用して word2vec のモデルを作成して単語同士の関係性を分析してみる。

【問題】
◎青空文庫にある「老人と海」の最初と最後の文章を表示する。(不要な部分を取り除く処理は完了済)
■解答は S3YX.py
◎word2vec のモデルを作成して、「老人」と「海」のそれぞれに対して、
類似した単語の上位 5 個を表示する。さらに、「老人」と「海」のコサイン類似度を計算する。
■解答は WS2E.py

■1-12：word2vec：S3JN.py
単語をベクトル表現にすることは出来たが、ベクトルが多次元であるので、直感的に人間には理解しにくい形式になっている。そこで、単語同士の相関を分析するために、二次元平面に単語をプロットしてみる。
このとき、主成分分析(PCA)を用いることで次元を圧縮して二次元ベクトルにする。
◎老人」と「海」それぞれに類似している単語の上位 100 個を平面にプロットして可視化する。
(このとき可視化するための draw_2d_2groups メソッドを使用する)
■解答は GM8N.py

■1-15：word2vec：FP2F.py
word2vecのモデルでは類似度を計算出来るので、単語同士の類似性を活用したアプリケーションをつくりやすい。そこで、今回は、連想ゲームをするチャットボットを作ってみよう。
(HR 領域の単語ベクトルとして公開されている 200 次元のワードベクトルのデータを使用する)

【問題】
◎単語を自分で入力したら類似の単語を表示し、その単語から類似される単語をもう一度入力する。
これを繰り返して行う連想ゲームをするチャットボットをつくる。
■解答は W3MS.py