【TechGym】ゼロからはじめる機械学習入門講座「前処理と特徴量エンジニアリング」(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。(抜粋版なので問題番号は連番ではないです)

■2-1：ヒストグラム：4DSa.py
【問題】
　データの特徴を知る方法としてヒストグラム(度数分布図)がある。これは度数と階級をグラフにしたもので、データの分布状況を視覚的に捉えることが出来る。

(1) ヒストグラムを実際に表示してみよう。
　　scikit-learn に組み込まれている、乳がん細胞のデータの中に含まれている細胞核の半径(mean radius)をヒストグラムとして表示する。このとき、matplotlib の hist メソッドを用いて bins=16 にする。
　　■解答は Ok8C.py

(2)他の方法でもヒストグラムを表示してみよう。
　　seaborn ライブラリ(distplot)を使用してヒストグラムを表示する。データは(1)と同じものを用いる。
　　また、カーネル密度関数も描画する。これはサンプルデータから元の確率分布を推定するものです。
　　描画の設定は sns.set_style('whitegrid')とする。bins=16、label='mean radius'とする。
　　■解答は aWD4.py

■2-3：欠損値：xSD3.py
　扱うデータには欠損値があるデータがある。定義上値が存在しないことや、入力をすることをミスしていてい値がないことや、センサーデータのエラーによって取得出来なかったなどの様々な理由がある。
【問題】
乳がん細胞のデータの中に含まれている細胞核の半径(mean radius)を用いる。問題で欠損値 NaN を埋め込んでいる。
(1) 欠損値の確認をする。欠損値の数、データの数、平均値を表示する。また、以下の欠損値の処理をした後で欠損値の数、データの数、平均値を毎回表示する。
　　◎NaN がある行をすべて取り除く(リストワイズ削除)
　　◎NaN を 0 に置き換える
　　◎NaN を直前の行の値で置き換える。さらに、NaN を直後の行の値で置き換える
　　◎NaN を平均値で置き換える
　　■解答は V8Ay.py
(2)欠損値を補完値で置き換えてみる。
　　◎平均値から標準偏差でばらつきを考慮して補完(標準偏差の範囲内でランダムに補完値を作る)
　　◎interpolate()で線形補間をする(補完方法は linear、補完方向は forward、補完領域は inside)
　　■解答は SKx7.py

■2-5：正規化(スケーリング)：DOe4.py
　スケーリングをする方法として、変数の範囲を 0 から 1 の区間にする方法がある。この Min-Max スケーリングはあらかじめ変数の範囲が決まっているときには有効となる。ただし、この方法では外れ値の影響を受けることもあり、平均値がちょうど 0 にならない。
　データの最大値を max(x)、最小値を min(x)としたとき、各変数(x)から最小値 min(x)を引いた値を最大値 max(x)と最小値 min(x)の差で割ったものが新しい変数(x')になる。このとき、最小値が 1 で最小値が 0 になる。
【問題】
(1) scikit-learn に組み込まれている、乳がん細胞のデータで目的変数を target、説明変数を data としてロジスティック回帰でモデルをつくり、陽性・陰性を予測してみる。テストデータと訓練データの分割には train_test_split を使用する。訓練データとテストデータの正解率を計算し表示する。
　　データを Min-Max スケーリングして、同様にロジスティック回帰で予測をする。このときに、正解率がどの程度変化するかを調べてみる。(回帰の条件は random_state=0,solver='liblinear')
　　■解答は pJ0C.py

■2-6：カテゴリ変数：PZ6b.py
　数値変数以外にも文字列などのカテゴリ変数がある。このカテゴリ変数はそのままでは分析に使えずに変換する必要がある。数値のデータでも値に意味がない場合はカテゴリ変数として扱うこともある。
【問題】
(1) one-hot-encoding はカテゴリの列を作って、その列の 1 つだけを「1」 にして、その他の列を「0」にする。与えられたデータフレームのカテゴリ変数を one-hot-encoding する。このとき、get_dummies を使用する。さらに、one-hot の vector を自分で作成して表示する。

【TechGym】ゼロからはじめる機械学習入門講座「前処理と特徴量エンジニアリング」(テックジムオープン講座)

■解答は BCh6.py

(2) scikit-learn の OneHotEncoder を使用してカテゴリ変数を one-hot-encoding を実行する。

■解答は VWs2.py

(3) one-hot-encoding ではカテゴリ変数が非常に多い場合には変換後の変数は 0 が多い特徴量になる。余計な 0 を入れないようにするために、Feature Hashing という変換方法がある。これは、ハッシュ関数を用いてカテゴリ変数を変換する。この方法では、変換後の特徴量の数(ベクトル次元)を自分で決めることが出来る。
与えられたデータフレーム(df_D)で sklearn.feature_extraction の FeatureHasher を使用して変換を実行してみる。(このとき特徴量のベクトル次元は 5 とする)

■解答は Vw4L.py

■2-8：特徴量エンジニアリング：CDh3.py

【問題】

実際のデータを使用してこれまでに学んできたことを実践してみよう。サッカーゲームに使用する選手のデータを分析する。FIFA19 の選手データベースがあるのでこれを用いる。(https://www.kaggle.com/karangadiya/fifa19)

(1) 欠損値があるデータはそのままでは使用出来ないので、場所を確認して欠損値を補完する。文字列のデータの箇所は特定の値を指定して欠損値を補完する。

Club → No_Club
Preferred Foot → Right
International Reputation → 1
Weak Foot → 3
Work Rate → Medium/ Medium
Body Type → Normal
Position → ST
Jersey Number → 8
Joined → Jul 1, 2018'
Loaned From → **None**
Contract Valid Until → 2019
Height → 5'11
Weight → 200lbs
'Wage' → €200k
'Release Clause' → €4.2M

その他の数値のデータは平均値で補完する。
'Skill Moves'は整数値で平均値では補完出来ないので中央値で補完する。
ポジションについては補間せずにそのままにしておく('LS'～'RB')
また、その他のNaNがある場合に備えて、残りのNaNは「0」に置き換える。
さらに、URLなどの以下の不要な行を取り除く。('Unnamed: 0','Photo','Flag','Club Logo')

■解答は X2Fm.py

(2) 分析に使用する新しい特徴量を作成してみる。
まずは、特徴量の操作をする関数を自分で定義して、その関数を apply メソッドによってデータに適用させて、新しい列を追加する。
Defending : 'Marking','StandingTackle','SlidingTackle'のすべてを平均値した値
General : 'HeadingAccuracy','Dribbling','Curve','BallControl'のすべてを平均値した値
Mental : 'Aggression','Interceptions','Positioning','Vision','Composure'のすべてを平均値した値
Passing : 'Crossing', 'ShortPassing','LongPassing'のすべてを平均値した値
Mobility : 'Acceleration','SprintSpeed','Agility','Reactions'のすべてを平均値した値
Power : 'Balance','Jumping','Stamina','Strength'のすべてを平均値した値
Rating : 'Potential','Overall'のすべてを平均値した値
Shooting : 'Finishing','Volleys','FKAccuracy','ShotPower','LongShots','Penalties'のすべてを平均値した値
さらに、すべての定義する関数では計算したものに対して round メソッドによって値を丸める。
また、作成したすべての特徴量のヒストグラムを表示する。
後の課題で使用するために、'FIFA_data_pre.csv'としてファイルに書き出す。

■解答は O9Wq.py

##フランチャイズ校を募集しております。

【TechGym】ゼロからはじめる機械学習入門講座「前処理と特徴量エンジニアリング」(テックジムオープン講座)
■2-9：特徴量エンジニアリング：iT8G.py

【問題】
サッカーゲームに使用する選手のデータを分析する。前の課題で保存したデータを使用する。
(1) ‘Weight’(体重)には単位である lbs が数値データについている。そこで、lbs を取り除く処理をする。
lbs を取り除く関数を定義して、apply メソッドでデータに適用する。‘Weight’(体重)のヒストグラムを表示する。

(2) 選手の年俸などの金額が入っているデータはユーロ(€)が単位としてついている。そこで、このユーロを取り除く、さらに、単位が M や K になっているので数値データとして扱えるように変換する。対象となるのは、'Value'、'Wage'、'Release Clause'とする。‘Wage’(収入)のヒストグラムを表示する。
変換したデータフレームを’FIFA_data_pre2.csv’として保存する。
■解答は kSI4.py

■2-11：特徴量エンジニアリング：i5UD.py
【問題】
サッカーゲームに使用する選手のデータを分析する。前の課題で保存したデータを使用する。
(1) 前の課題で作成した特徴量で重回帰分析を実施する。
'Defending','General','Mental','Passing','Mobility','Power','Shooting'を説明変数とする。また、'Overall'を目的変数として、重回帰分析を行い、決定係数と RMSE を計算する。
データ分割は train_test_split を使用して、test_size=0.2 とする。
■解答は TD0x.py

(2) 作成した特徴量のみでは決定係数が低いため、すべての変数を用いて予測をする。カテゴリ変数を変換して特徴量を作る。まずは、どの変数がカテゴリ変数かを確認するために型が object である列のみを表示する。

以下の変換をする関数を定義して、apply メソッドですべてデータに適用して、新しい列を追加する。
◎‘Real Face'は Yes のときは「1」それ以外の No のときは「0」と変換する。
◎‘Preferred Foot’は右利きのときは「1」それ以外の左利きのときは「0」と変換する。
◎サッカーのポジションは分類を変更する。’Simple_Position’として列を追加する。
GK → GK
DF → RB,LB,CB,LCB,RCB,RWB,LWB
DM → LDM,CDM,RDM
MF → LM,LCM,CM,RCM,RM
AM → LAM,CAM,RAM,LW,RW
ST → RS,ST,LS,CF,LF,RF
◎‘Nationality’は選手が 250 人より多い国のときは「1」それ以外の 250 以下のときは「0」とする。

(3) ‘Work Rate’は「/」の文字で区切られているので、それぞれの文字を分離して、新しい列として加える。新しい列は’WorkRate1’と’WorkRate2’とする。

(4) ‘Height’は feet と inch で表記されている。「'」の文字で区切られているので、それぞれの文字を分離して新しい列とする。さらに、小数点の数値データとするため型変換をする。
feet に 30.48 を掛けて、inch に 2.54 を掛けてそれぞれを加えることで cm(センチメートル)の単位にする。これを’Height_cm’として列に加える。さらに、変換前のカテゴリ変数の列が残っているので、これを取り除く。

(5) ’Simple_Position’, ’WorkRate1’,’WorkRate2’はまだカテゴリ変数として残っているので、get_dummies メソッドを使って one-hot-encoding で変換する。

(6) 前問題と同様に、'Overall'を目的変数として、重回帰分析を行い、決定係数と RMSE を計算する。説明変数としては’Overall’以外のすべての変数を使用する。
■解答は X5Xy.py