# Assignment 3 – Linear Regression and Variable Selection on MOVIE DATASET

*The purpose of this assignment is to use Python to learn how to perform feature selection on the given dataset.*

This assignment provides you with an opportunity to demonstrate the achievement of the following course learning outcomes:

* Understand Linear regression
* Understand and apply the Python programming language
* Understand and apply feature selection principles

## Key Information

* **Type:** *Individual*
* **Weight:** 6.5%
* **Delivery:** Course website upload
* **Due Date:** End of lab session

## Expectations

You are expected to complete this assignment individually.

Respect for academic integrity is crucial to your success. Make sure you understand what constitutes acts of academic dishonesty in the page: [What is Academic Dishonesty?](http://mcmaster.ca/academicintegrity/students/whatis.html)

## Instructions

*Using Python, you are to complete the following questions.*

Please work on the following questions and tasks:

**Part I:**

1. **Import csv dataset (MOVIE DATASET)**
2. **Identify what features have missing values**
3. **Replace missing values with 0**
4. **Split dataset into training and test datasets in 0.7/0.3**
5. **Select all numerical features that can be used to predict ‘vote average’ target**
6. **Add three categorical features (for example, original language, status and genres) and transform them into numerical**
7. **Fill the training and test data with required information:** 
   1. **Use: train[features].dropna()**
8. **Import linear regression function from sklearn:** 
   1. **Use: linear\_model.LinearRegression() from sklearn**
9. **Use score to identify most accurate and predictive features. Justify the final choice of the features.**

**Part II:**

1. **Remove features with low variance and apply linear regression**:
   1. ***Use the same data set from step 8***
   2. Use VarianceThreshold from sklearn.feature\_selection

**VarianceThreshold** is a simple baseline approach to feature selection. It removes all features whose variance doesn’t meet some threshold. By default, it removes all zero-variance features, i.e. features that have the same value in all samples.

As an example, suppose that we have a dataset with boolean features, and we want to remove all features that are either one or zero (on or off) in more than 80% of the samples. Boolean features are Bernoulli random variables, and the variance of such variables is given by

mathrm{Var}[X] = p(1 - p)

so we can select using the threshold .8 \* (1 - .8)

Apply following example on the MOVIE dataset:

**>>> from** **sklearn.feature\_selection** **import** VarianceThreshold

**>>>** X = [[0, 0, 1], [0, 1, 0], [1, 0, 0], [0, 1, 1], [0, 1, 0], [0, 1, 1]]

**>>>** sel = VarianceThreshold(threshold=(.8 \* (1 - .8)))

**>>>** sel.fit\_transform(X) array([[0, 1], [1, 0], [0, 0], [1, 1], [1, 0], [1, 1]])

* 1. Calculate the score after applying linear regression on the new set

1. **Recursive feature elimination with linear regression:**

***Use the same data set from step 8.***

Sklearn provides RFE for recursive feature elimination and RFECV for finding the ranks together with optimal number of features via a cross validation loop.

* + Import: RFE from sklearn.feature\_selection
  + Import: LinearREgression from sklearn.linear\_model
  + Define features and target as X and Y
  + Use linear regression as the model
    - lr = LinearRegression()
  + Rank all features, i.e continue the elimination until the last one
    - rfe = RFE(lr, n\_features\_to\_select=1)
    - rfe.fit(X,Y)
  + Calculate the score after applying linear regression on the new set
  + Print Features sorted by their rank
  + What observations can you make based on obtained ranking?

## Rubric

To achieve full marks on this assignment, you must have answered all questions above correctly with code submitted that has no errors.

## Appendix

You can use the following code to add and extract information from the json column in the DataFrame

**import** json  
**def** get\_data(column, df, key):  
 df\_temp = pd.DataFrame([])  
 **for** i **in** column:  
 json\_data = json.loads(i)  
 df1 = pd.DataFrame.from\_dict(json\_data)  
 **if** len(i) > 2:  
 df2 = pd.DataFrame([df1[key][0]])  
 **else**:  
 df2 = df1  
 df\_temp = pd.concat([df\_temp, df2], ignore\_index=**True**)  
  
 df = pd.concat([df, df\_temp], axis=1)  
 **return** df

Example:  
The function can be used for example as the following:  
df = get\_data(df[**'genres'**], df, **'name'**)

*The genres are appended to the df as a column*