Student Name:_____Roll No:_____Section:_____

CS334 - Machine Learning

Lab 04 - Feature Selection Methods in ML (Part - 1)

Instructor: Ms. Maham Ashraf
E-mail: mashraf@uit.edu

Semester: Fall, 2023

Objective

The purpose of this lab session is to introduce feature selection methods for machine learning model. This lab is divided into two parts, in Part-1 we will use only Filter for feature selection. In Part-2, we will use Wrapper, Embedded and Hybrid methods for feature selections.

Instructions

You have to perform the following tasks yourselves. Raise your hand if you face any difficulty in understanding and solving these tasks. **Plagiarism** is an abhorrent practice and you should not engage in it.

How to Submit

- Submit lab work in a single .py file on Microsoft Teams. (No other format will be accepted)
- Lab work file name should be saved with your roll number (e.g. 19a-001-SE_LW01.py)
- Submit home work in a single .py file on Microsoft Teams. (No other format will be accepted)
- Home work file name should be saved with your roll number (e.g. 19a-001-SE_HW01.py)

1 Filter Methods for Features Selection

1.1 Information Gain

Information gain calculates the reduction in entropy from the transformation of a dataset. It can be used for feature selection by evaluating the Information gain of each variable in the context of the target variable.

```
# load data pima-indians-diabetes data set
import pandas as pd
from sklearn.feature_selection import mutual_info_classif
# X is a list of all features except target feature
# y is target feature
importance = mutual_info_classif(X,y)
feat_importance = pd.Series(importance, dataframe.columns[0: len(dataframe.
    columns)-1])
feat_importance.plot(kind='barh', color='teal')
```

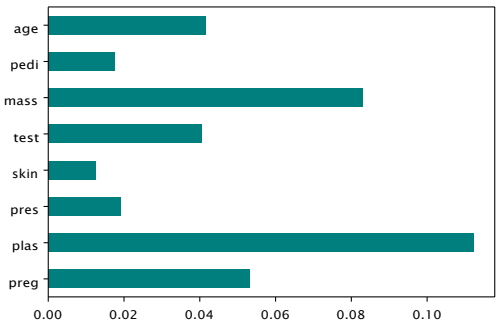

Figure 1: Feature importance histogram graph

1.2 χ^2 Test

This is a classification predictive modeling problem with categorical input variables. You can also use mutual information (information gain) from the field of information theory.

- Chi-Squared test (contingency tables).
- Mutual Information.

We calculate χ^2 between each feature and the target and select the desired number of features with the best χ^2 scores. We are using *Churn Modeling* data set for selecting features.

```
# Load churn_Modeling.csv file
# fit into pandas data frame
#          encode          categorical          variables
churn_df_sel=churn_df.loc[:,['Geography','Gender','HasCrCard', '
  IsActive Member','Exited']]
churn_df_sel.head()
 # use chi-square method for feature selection and see the chi-square
  statistics
 from sklearn.feature_selection import chi2
 X = churn_df_sel.drop('Exited',axis=1)
 y = churn_df_sel['Exited']
 # chi square test with 95% confidence interval
 chi_scores = chi2(X,y)
```

Select the features using the χ^2 table for 95% confidence interval.

Features of data set can be selected by observing the p-value of χ^2 test. The method to compute the p-values of χ^2 test is given below,

```
p_values = pd.Series(chi_scores[1],index = X.columns)
print(p_values[:])
p_values.sort_values(ascending = True , inplace = True)

# apply kbest or percentile method along with chi-square test
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import SelectPercentile
X_new.fit_transform(X,y)
feature_names = ['Geography','Gender','HasCrCard', 'IsActiveMember','
  Exited']
X_new = SelectPercentile(chi2, percentile=50) # use percentile method
# X_new = Select KBest ( chi2 , k = 3) # k best method
X_new.fit_transform(X,y)

# get the selected features
new_features = [] # The list of your K best features
mask = X_new.get_support()

# Print selected fatures
new_features = [] # The list of your K best features
for bool, feature in zip(mask, feature_names):
if      bool     :
new_features.append(feature)
new_features
```

Table entry for p is the critical value $(x^2)^*$ with probability p lying to its right.

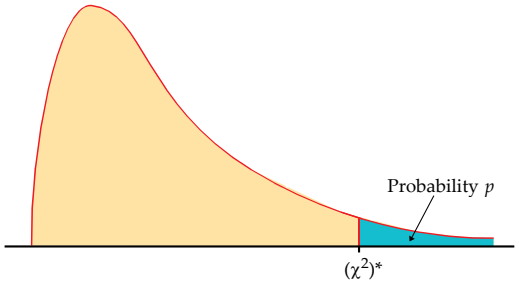

TABLE F

x^2 distribution critical values

	Tail probability p											
df	.25	.20	.15	.10	.05	.025	.02	.01	.005	.0025	.001	.0005
1	1.32	1.64	2.07	2.71	3.84	5.02	5.41	6.63	7.88	9.14	10.83	12.12
2	2.77	3.22	3.79	4.61	5.99	7.38	7.82	9.21	10.60	11.98	13.82	15.20
3	4.11	4.64	5.32	6.25	7.81	9.35	9.84	11.34	12.84	14.32	16.27	17.73
4	5.39	5.99	6.74	7.78	9.49	11.14	11.67	13.28	14.86	16.42	18.47	20.00
5	6.63	7.29	8.12	9.24	11.07	12.83	13.39	15.09	16.75	18.39	20.51	22.11
6	7.84	8.56	9.45	10.64	12.59	14.45	15.03	16.81	18.55	20.25	22.46	24.10
7	9.04	9.80	10.75	12.02	14.07	16.01	16.62	18.48	20.28	22.04	24.32	26.02
8	10.22	11.03	12.03	13.36	15.51	17.53	18.17	20.09	21.95	23.77	26.12	27.87
9	11.39	12.24	13.29	14.68	16.92	19.02	19.68	21.67	23.59	25.46	27.88	29.67
10	12.55	13.44	14.53	15.99	18.31	20.48	21.16	23.21	25.19	27.11	29.59	31.42
11	13.70	14.63	15.77	17.28	19.68	21.92	22.62	24.72	26.76	28.73	31.26	33.14
12	14.85	15.81	16.99	18.55	21.03	23.34	24.05	26.22	28.30	30.32	32.91	34.82
13	15.98	16.98	18.20	19.81	22.36	24.74	25.47	27.69	29.82	31.88	34.53	36.48
14	17.12	18.15	19.41	21.06	23.68	26.12	26.87	29.14	31.32	33.43	36.12	38.11
15	18.25	19.31	20.60	22.31	25.00	27.49	28.26	30.58	32.80	34.95	37.70	39.72
16	19.37	20.47	21.79	23.54	26.30	28.85	29.63	32.00	34.27	36.46	39.25	41.31
17	20.49	21.61	22.98	24.77	27.59	30.19	31.00	33.41	35.72	37.95	40.79	42.88
18	21.60	22.76	24.16	25.99	28.87	31.53	32.35	34.81	37.16	39.42	42.31	44.43
19	22.72	23.90	25.33	27.20	30.14	32.85	33.69	36.19	38.58	40.88	43.82	45.97
20	23.83	25.04	26.50	28.41	31.41	34.17	35.02	37.57	40.00	42.34	45.31	47.50
21	24.93	26.17	27.66	29.62	32.67	35.48	36.34	38.93	41.40	43.78	46.80	49.01
22	26.04	27.30	28.82	30.81	33.92	36.78	37.66	40.29	42.80	45.20	48.27	50.51
23	27.14	28.43	29.98	32.01	35.17	38.08	38.97	41.64	44.18	46.62	49.73	52.00
24	28.24	29.55	31.13	33.20	36.42	39.36	40.27	42.98	45.56	48.03	51.18	53.48
25	29.34	30.68	32.28	34.38	37.65	40.65	41.57	44.31	46.93	49.44	52.62	54.95
26	30.43	31.79	33.43	35.56	38.89	41.92	42.86	45.64	48.29	50.83	54.05	56.41
27	31.53	32.91	34.57	36.74	40.11	43.19	44.14	46.96	49.64	52.22	55.48	57.86
28	32.62	34.03	35.71	37.92	41.34	44.46	45.42	48.28	50.99	53.59	56.89	59.30
29	33.71	35.14	36.85	39.09	42.56	45.72	46.69	49.59	52.34	54.97	58.30	60.73
30	34.80	36.25	37.99	40.26	43.77	46.98	47.96	50.89	53.67	56.33	59.70	62.16
40	45.62	47.27	49.24	51.81	55.76	59.34	60.44	63.69	66.77	69.70	73.40	76.09
50	56.33	58.16	60.35	63.17	67.50	71.42	72.61	76.15	79.49	82.66	86.66	89.56
60	66.98	68.97	71.34	74.40	79.08	83.30	84.58	88.38	91.95	95.34	99.61	102.7
80	88.13	90.41	93.11	96.58	101.9	106.6	108.1	112.3	116.3	120.1	124.8	128.3
100	109.1	111.7	114.7	118.5	124.3	129.6	131.1	135.8	140.2	144.3	149.4	153.2

Figure 2: χ^2*Table*

1.3 Fisher Score

Fisher score is one of the most widely used supervised feature selection methods. The algorithm which we will use returns the ranks of the variables based on the fisher's score in descending order. We can then select the variables as per the case.

```
from skfeature.function.similarity_based import fisher_score
data = pd.read_csv('churn_Modelling.csv')
df = pd.DataFrame(data)
churn_df=df.loc[:,['Geography','Gender','HasCrCard', 'IsActiveMember','
  Exited']]

label_encoder = LabelEncoder()
churn_df['Geography'] = label_encoder.fit_transform(churn_df['Geography'])
churn_df['Gender'] = label_encoder.fit_transform(churn_df['Gender'])

X_train = churn_df.drop('Exited',axis=1)
y_train = churn_df['Exited']
#calculate score
rank = fisher_score.fisher_score(X_train.to_numpy(), y_train.to_numpy(),
  mode='rank')

# print histogram graph
feat_importances = pd.Series(rank,churn_df.columns[0:len(churn_df.columns)
  -1])
feat_importances.plot(kind='barh', color='teal')
plt.show
```

1.4 Correlation Coefficient & Variance method:

The correlation coefficient is a statistical measure of the strength of the relationship between the relative movements of two variables. The values range between -1.0 and 1.0. The logic behind using correlation for feature selection is that the good variables are highly correlated with the target. Furthermore, variables should be correlated with the target but should be uncorrelated among themselves.

If two variables are correlated, we can predict one from the other. Therefore, if two features are correlated, the model only really needs one of them, as the second one does not add additional information. We will use the Pearson Correlation here.

We need to set an absolute value, say 0.5 as the threshold for selecting the variables. If we find that the predictor variables are correlated among themselves, we can drop the variable which has a lower correlation coefficient value with the target variable. We can also compute multiple correlation coefficients to check whether more than two variables are correlated to each other. This phenomenon is known as multicollinearity.

```
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

# corelation matrix
corr = churn_df.corr()
plt.figure(figsize=(10,6))
sns.heatmap(corr, annot=True)
```

The variance threshold is a simple baseline approach to feature selection. It removes all features which 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. We assume that features with a higher variance may contain more useful information, but note that we are not taking the relationship between feature variables or feature and target variables into account, which is one of the drawbacks of filter methods.

```
from sklearn.feature_selection import VarianceThreshold
v_threshold = VarianceThreshold(threshold=0)
v_threshold.fit(churn_df)
v_threshold.get_support()
```

1.5 Mean Absolute Difference (MAD):

The mean absolute difference (MAD) computes the absolute difference from the mean value. The main difference between the variance and MAD measures is the absence of the square in the latter.

The MAD, like the variance, is also a scale variant. This means that higher the MAD, higher the discriminatory power.

```
# load data
import pandas as pd
import numpy as np
data = 'Iris.csv'
names = ['ID','sepalLength','sepalWidth','petalLength','petalWidth','
  species']
df = pd.read_csv(data, names=names)
df.head()
X = df.iloc[:,1:5]
Y = df.iloc[:,5]
mad = X.mad(axis=0);
print("Mean absolute deviation of columns:",mad);
```

1.6 Dispersion ratio:

Another measure of dispersion applies the arithmetic mean (AM) and the geometric mean (GM). For a given (positive) feature Xi on n patterns, the AM and GM are given by,

$$AM_i = \bar{X} = \frac{1}{n}\sum_{j=1}^{n} x_{ij}, \quad GM_i = \prod_{j=1}^{n} (x_{ij})^{\frac{1}{n}}$$

respectively; since $AM_i \geq GM_i$, with equality holding if and only if $x_{i1} = x_{i2} = \cdots = x_{in}$, then the ratio,

$$RM_i = \frac{AM_i}{GM_i} \in [1, +\infty)$$

can be used as a dispersion measure. Higher dispersion implies a higher value of R_i, thus a more relevant feature. Conversely, when all the feature samples have (roughly) the same value, R_i is close to 1, indicating a low relevance feature.

```
X = X+1
am = np.mean(X, axis=0)
gm = np.power(np.product(X, axis=0),1/X.shape[0])
# ratio of arithmatic mean and geometric mean
disp_ratio = am/gm
plt.bar(np.arange(X.shape[1]),disp_ratio,color='teal')
```

2 Homework

1. Apply *Information Gain (IG)* methods for feature selection on *Motor Insurance Fraud* data set and discover the selected features.
2. Apply χ^2 methods for feature selection on *diabetes* data set and discover the selected features.
3. Apply *Fisher score* methods for feature selection on *Motor Insurance Fraud* data set and discover the selected features.
4. Apply *Correlation & Variance* methods for feature selection on *Property* data set and discover the selected features.
5. Apply *Mean Absolute Difference (MAD)* method for feature selection on *diabetes* data set and discover the selected features.
6. Apply *Dispersion Ratio* method for feature selection on *Churn* data set and discover the selected features.