THE IMPLICATION OF STATISTICAL ANALYSIS AND FEATURE ENGINEERING FOR MODEL BUILDING USING MACHINE LEARNING ALGORITHMS

Swayanshu Shanti Pragnya and Shashwat Priyadarshi

Fellow of Computer Science Research, Global Journals
Sr. Python Developer, Accenture, Hyderabad

ABSTRACT

Scrutiny for presage is the era of advance statistics where accuracy matter the most. Commensurate between algorithms with statistical implementation provides better consequence in terms of accurate prediction by using data sets. Prolific usage of algorithms lead towards the simplification of mathematical models, which provide less manual calculations. Presage is the essence of data science and machine learning requisitions that impart control over situations. Implementation of any dogmas require proper feature extraction which helps in the proper model building that assist in precision. This paper is predominantly based on different statistical analysis which includes correlation significance and proper categorical data distribution using feature engineering technique that unravel accuracy of different models of machine learning algorithms.

KEYWORDS:

Correlation, Feature engineering, Feature selection, PCA, K nearest neighbour, logistic regression, RFE

1. INTRODUCTION

Statistical analysis is performed just to analyse the data little bit more by using statistical conventions. But only analysing a data is not sufficient when it comes to analysis that too by using statistics only. So at this point predictive analysis comes which is nothing but a part of inferential statistics. Here we try to infer any outcome based on analysing patterns from previous data just to predict for the next dataset when it comes to prediction first buzzword came i.e. machine learning. Machine learning is the way to train the machine for required task completion. Here machine learning is used to predict the survival of the passengers in the titanic disaster. But prediction of the survival depends upon how effectively we can reform the dataset. For enhancement or reform of the data set feature extraction is required. By using Logistic regression technique [9] the prediction accuracy increased to 80.756%. The actual Titanic disaster which was a ship voyage sunk in the Northern Atlantic on 15th Apr,1912 where 1502 passengers crewed out of 2224 [1]. The reason behind sinking, which data impacted more upon the analysis of survival is continuing [2], [3]. For analysing the data set more effectively is already available in the Kaggle website [4]. Kaggle has given the platform for data analysis and machine learning [4]. The persons who are able to predict to the most accurate Kaggle provides cash prize for encouragement. [1]. This paper comprises of explaining the importance and higher usability of extracting feature from data sets and how these accurate extraction will help in the accurate prediction using machine learning algorithms.

DOI:10.5121/ijcses.2019.10301

Before going through the topic we need to understand data. Generally through our study we collect different type of information which is known as data. Data can be numerical (discrete and continuous), categorical and ordinal. **Numerical data** represents different type of measurements like person's age, height or length of any train. These numerical data also known as quantitative data.

Discrete data are ought to counted. For example if we flip a coin for 100 times then the result can be determined in a generalized manner of 2^n, where n = number of times to flip. So here the number of outcome is finite so this data is discrete by nature.

Continuous data are not finite as the name itself defines its continuing. For example the value of pi i.e. 3.14159265358979323. And so on. That's the reason for calculating such continuous data we have to take an approximation.

Categorical data represents the nature of the data like a person's gender or answer of any question which is yes or no. Though these are characteristics of the data so we need to convert these data to numeric format. Example if probability of a question is 'yes' then we need to assign 'yes' as 1 or any integer so that machine will understand.

Ordinal data is the amalgamation of numeric and categorical data. It means data will fall into different categories but whatever numbers are placed on the category has some meaning. For example if in a survey of 1000 people and will ask them to give the rate of hospitality they got at the hospital from nurses on the scale of 0 to 5, then by taking the average of 1000 rate of responses will have meaning. Here this scenario or data would not be considered as categorical data.

Here we got the brief idea about different type of data and how we are going to recognise through examples. Though the reason behind knowing the feature extraction is to implement in machine learning process so we need to know about machine learning processes for both train and test data as given below.

Process to train data is given below-

Data collection → Data Pre-processing → **Feature extraction** → Model building → Model evaluation →Deployment → Model

Machine learning workflow for test data set i.e. given below-

Data collection → Data Pre-processing → **Feature Extraction** → Model → Predictions

Training a data and then gain testing the data is the steps towards implementing any model in machine learning towards prediction or regression and classification as these two are the main functionality of machine learning algorithms.

2. DATA PREPARATION PIPELINE

Here the aim is to show a Machine learning (ML) project work flow to build data preparation pipeline which transforms Pandas data frame to numpy array for training ML models with Scikit-Learn.

This process includes the following steps.

1. Splitting data into labels and predictors.
2. Mapping of data frame and selecting variables.
3. Categorical variable encoding
4. Filling missing data
5. Scaling numeric data
6. Assemble final pipeline
7. Test the final pipeline.

```
// Step 1 Splitting data into labels and predictor
import pandas as pd
train_data = pd.read_csv('data/housing_train.csv')
X = train_data.drop(['median_house_value'], axis=1)
y = train_data['median_house_value']
X.info()
X.head(5)
// Step 2 Mapping of data frame
import numpy as np
from sklearn.base import BaseEstimator, TransformerMixin
from sklearn.preprocessing import OneHotEncoder
// Step 3 selecting variables
class DataFrameAdapter(BaseEstimator, TransformerMixin)
def __init__(self, col_names):
self.col_names = list(col_names)
def fit(self, X, y=None):
return self
// Step 4
class CategoricalFeatureEncoder(BaseEstimator, TransformerMixin):
def __init__(self):
return None
def fit(self, X, y=None):
return self
// Step 5 Filling missing data
from sklearn.preprocessing import Imputer
num_data = X.drop(['ocean_proximity'], axis=1)
num_imputer = Imputer(strategy='median')
imputed_num_data = num_imputer.fit_transform(num_data)
// Step 6 Scaling numeric data
from sklearn.pipeline import Pipeline, FeatureUnion
numeric_cols = X.select_dtypes(exclude=['object']).columns
numeric_pipeline = Pipeline([
('var_selector', DataFrameAdapter(numeric_cols)),
('imputer', Imputer(strategy='median')),
('scaler', MinMaxScaler())
])
// Step 7 Assemble final pipeline
prepared_data = data_prep_pipeline.fit_transform(X)
```

```
print('prepared data has {} observations of {}
features'.format(*prepared_data.shape))
```

Fig 1. Steps for data preparation

Data pre-processing includes different type of data modification like dummy value replacement, data value replacement by using numeric values.

Dimensionality reduction is required in machine learning algorithm implementation as space complexity along with efficiency is the factor of any computation. It comprises of two factor i.e. feature selection and feature extraction.

Feature selection is comprises of Wrapper, Filter and embedded method.

Example- For improvising performance let's take a, b, c, d are different feature and create an equation as

$$a+b+c+d = e$$

$$\text{If } ab = a + b \text{ (Feature extraction)}$$

$$ab + c + d = e$$

$$\text{Let's take } c = 0 \text{ (As condition)}$$

$$ab + d = e \text{ (Feature selection)}$$

In the above example we came to know that how replacing few values and adding conditions in features completely changed and reduced the equation in terms of dimension. Initially there are five features and now it reduced to only three features.

3. METHODS OF FEATURE EXTRACTION

Any type of statistical model comprises of the following equation like,

$Y = \beta0 + \beta1X1 + \beta2X2 + \ldots. + \varepsilon$

Where X1 up to Xn are of different features.

Need of Feature Extraction:

It depends upon the number of features.

Less features:

1. Easy to interpret
2. Less likely to overfit
3. Low in prediction accuracy

More features:

1. Difficult to interpret as number of feature is high
2. More likely to overfit
3. High prediction accuracy

Feature Selection

It is also known as attribute or variable selection. The process to select attributes which are most relevant for the prediction. In other words feature selection is the way to select any subset of important feature to use in any model construction.

Difference between Dimensionality reduction and Feature selection:

Generally feature selection and dimensionality reduction seem hazy but both are different. Both has few similarity that too reducing number of attributes in the given data set is the work of feature selection process. But dimensionality reduction method also create new combination whereas feature selection method exclude and include feature or attributes present in the data set without changing them.

For example dimensionality reduction method includes singular value decomposition and Principal component analysis.

Feature Selection:

It is a process of selecting features in data set which has highest contribution for the out put column. Generally when we look at any data set those are consist of numerous type of data. All the columns are not vital for the processing. This is the reason to find features through selection method.

Another problem can be irrelevant feature may lead to decrease the accuracy of any model like linear regression.

Benefits of Feature Selection:

1. Improvement in Accuracy
2. Overfitting of data is very less
3. Time complexity (Less data which leads to faster execution)

Feature Selection for Machine Learning

There are different ways of feature selection in machine learning. Those are discussed below:-

1. Univariate Selection

Various statistical tests are performed for the selection of correlated features for the dependant column.

The library named selectKbest class by sci-kit library can perform statistical tests to select features.

The given example explains the chi squared statistical test for positive features. Model accuracy is used to identify the contributing target attribute.

The example below uses the chi squared (chi^2) statistical test for non-negative features to select 4 of the best features from the Pima Indians onset of diabetes data set.

```
import pandas
import numpy
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
# load data
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-
diabetes.data.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
dataframe = pandas.read_csv(url, names=names)
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
# feature extraction
test = SelectKBest(score_func=chi2, k=4)
fit = test.fit(X, Y)
# summarize scores
numpy.set_printoptions(precision=3)
print(fit.scores_)
features = fit.transform(X)
# summarize selected features
print(features[0:5,:])
you can see the scores for each attribute and the 4 attributes chosen (those with
the highest scores): plas, test, mass and age
O/p
[[ 148. 0. 33.6 50. ]
[ 85. 0. 26.6 31. ]
[ 183. 0. 23.3 32. ]
[ 89. 94. 28.1 21. ]
[ 137. 168. 43.1 33. ]]
```

Fig 2. Univariate selection

2. Recursive Feature Elimination

The Recursive Feature Elimination (or RFE) works recursively by removing attributes and Here logistic regression algorithm has been implemented to select to 3 features.

```
# Feature Extraction with RFE
from pandas import read_csv
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression
# load data
url="https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-
diabetes.data.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
dataframe = read_csv(url, names=names)
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
# feature extraction
model = LogisticRegression()
rfe = RFE(model, 3)
fit = rfe.fit(X, Y)
print("Num Features: %d") % fit.n_features_
print("Selected Features: %s") % fit.support_
print("Feature Ranking: %s") % fit.ranking_o/p
1 Num Features: 3
2 Selected Features:  [ True False False False False True True False]
3 Feature Ranking: [1 2 3 5 6 1 1 4]
```

Fig 3. Recursive feature selection using data set

3. Principal Component Analysis

PCA is uses algebra in linear format for the transformation of data set into compressed one. It is different from feature selection technique. Generally PCA is a dimension reduction technique. It can choose the number of dimension to reduce. The Fig. Below is the implication of PCA

```
# Principal Component Analysis
import numpy
from pandas import read_csv
from sklearn.decomposition import PCA
# load data
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-
diabetes.data.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
dataframe = read_csv(url, names=names)
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
# feature extraction
pca = PCA(n_components=3)
fit = pca.fit(X)
# summarize components
print("Explained Variance: %s") % fit.
```

Mathematics behind "import PCA" statement

The data set is resembles as a Vector of rows and columns. So the steps involved to implement PCA are as follows:-

1. Mean of the vector i.e. assuming we have N sample and we can compute the mean of vector as

$M = (M1 + M2 + \ldots\ldots + MN)/N$

2. Combine the mean adjusted matrix i.e. for every vector column 'p' the mean adjusted matrix will be

$\bar{Y} = Mp - M$ and Y mean = ($\bar{Y}1 \ldots\ldots \bar{Y}N$) (for column 'p')
$\bar{Y}'' = Mq - M$ (for row 'q')

3. Compute co variance matrix I.e.

$C(p,q) = \bar{Y}. \bar{Y}''$ (dot product of $\bar{Y}$ and $\bar{Y}''$)

4. Quantify Eigen values and Eigen vectors of co variance matrix.

5. Represent each combination of eigen value and vector as a linear combination of matrix

4. Feature Importance

A bagged decision tree for example random forest and extra umber of trees can be used to estimate the importance of features.

In the given example code we build ExtraTressClassifier which classifies for the data set named as Pima diabetes

```
# Feature Importance with Extra Trees Classifier
from pandas import read_csv
from sklearn.ensemble import
ExtraTreesClassifier
# load data
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class'] // selected columns from data set
dataframe = read_csv(url, names=names)
array = dataframe.values
X = array[:,0:8] // slicing method to select rows from 0 to 8
Y = array[:,8]
# feature extraction
model = ExtraTreesClassifier()
model.fit(X, Y) // particular rows and columns will be fitted for the training process
print(model.feature_importances_)
```

```
o/p- [
0.11070069  0.2213717  0.08824115  0.08068703  0.07281761  0.14548537  0.12654214
0.15415431]
```

Fig. 4 Live use of extra tree classifier

4. Model Implementation And Accuracy Analysis

In Module II and III, we explained the process of feature extraction, creation and selection. Along with we have provided the fully executable code. In this module we are going to discuss the change in accuracy by using the given techniques. The diabetes data consist of 768 data points with 9 features. Here we implemented logistic regression without correlation analysis.To know the correlation between each columns we need to find the correlation factor in data set.

Fig1. heat map shows that correlation between plasma glucose concentration and on set diabetes is high I.e. 0.8.

```
logreg001 = LogisticRegression(C=0.01).fit(X_train, y_train) print("Training set
accuracy: {:.3f}".format(logreg001.score(X_train, y_train))) print("Test set
accuracy: {:.3f}".format(logreg001.score(X_test, y_test)))
Training set accuracy: 0.700,
Test set accuracy: 0.703
//Less accuracy (Without correlation analysis)
```

Fig. 5. Logistic regression using diabetes data

After filling the missing values and selecting the high correlated columns now we can implement our algorithms to check the accuracy.

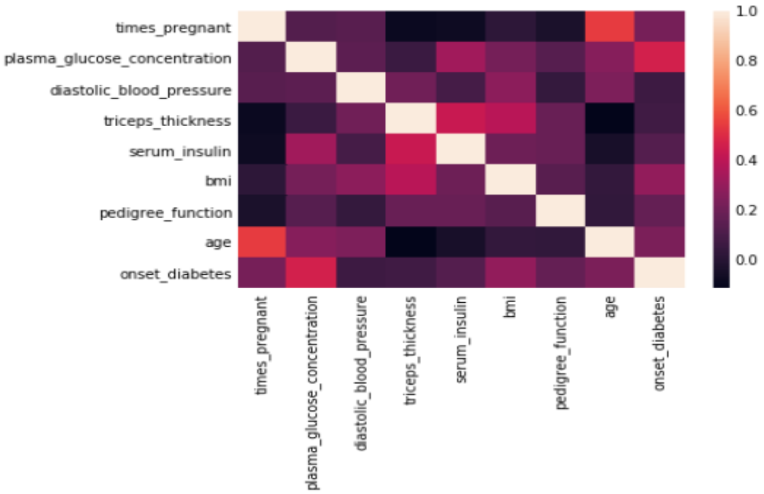

Fig 6. Correlation between features

After knowing the correlation factor we have modified the train and test data to implement K-NN as we got 9 features in our data set.

```
knn = KNeighborsClassifier(n_neighbors=9) knn.fit(X_train, y_train) // process of
training the data set assigned in x and y train
print('Accuracy of K-NN classifier on training set:
{:.2f}'.format(knn.score(X_train, y_train))) print('Accuracy of K-NN classifier on
test set: {:.2f}'.format(knn.score(X_test, y_test)))
Accuracy of K-NN classifier on training set: 0.79
Accuracy of K-NN classifier on test set: 0.78
// Improved
```

CONCLUSION

Here in all of the 4 subsection of the paper we discussed the following things i.e. types of data, steps involved to find correlation, feature engineering techniques, the difference between feature extraction and dimension reduction. In our final module we implemented logistic regression technique and got 0.70 as accuracy.

But after using a simple correlation function and Heat map visualization we sorted the data set with 9 features and by using K-nearest neighbour algorithm we are succeeded to get 0.78 as our model accuracy. Here we have shown the importance of selecting features and their impact on the improvisation of model accuracy.

We can visualize the importance of selecting proper feature by using statistical methods. Hence before experimenting on any algorithm we should vividly check the features as it clearly impacts on the accuracy.

The objective of our paper was to know which factors are important to improvise a model accuracy and which techniques can be helpful to achieve it. We got a conclusion that selecting proper feature along with reducing their dimension is correlated for enhancing model accuracy. But this is not the end as accuracy is increased only 11.42% which is not a major change that means there are other factors which we have to find and fix. So our next work will be on finding other factors which are merged in the process of upgradation.

FUTURE WORK:

In this experiment of implementing dimension reduction technique and selection method of feature are really helpful to increase a model accuracy. But the improvement is bit lesser. So we want to conduct another way to improvise the accuracy by using normalization techniques like min-max scaling, z score standardization and row normalization to develop model accuracy.

Along with this techniques we will implement different deep learning algorithms for better functionality. Understanding the factors which helps to improvise accuracy is really relevant to know as only few factors like selecting particular feature or even dimension reduction is not the only factor which we came to know in this paper. So more depth on each feature and development in training method is vital for improvisation.

Our next work will be fully focusing on normalization along with optimization of particular machine learning algorithm like **matrix notation** of **logistic regression** and random forest algorithm.

REFERENCES

[1] GE, “Flight Quest Challenge,” Kaggle.com. [Online]. Available: https://www.kaggle.com/c/flight2-final. [Accessed: 2-Jun-2017].

[2] “Titanic: Machine Learning from Disaster,” Kaggle.com. [Online]. Available: https://www.kaggle.com/c/titanic-gettingStarted. [Accessed: 2-Jun-2017].

[3] Wiki, “Titanic.” [Online]. Available: http://en.wikipedia.org/wiki/Titanic. [Accessed: 2-Jun-2017].

[4] Kaggle, Data Science Community, [Online]. Available: http://www.kaggle.com/ [Accessed: 2-Jun-2017].

[5] Multiple Regression, [Online] Available: https://statistics.laerd.com/spss-tutorials/multiple-regression-usingspss-statistics.php [Accessed: 2-Jun-2017].

[6] Logistic Regression, [Online] Available: https://en.wikipedia.org/wiki/Logistic_regression [Accessed: 2-Jun2017].

[7] Consumer Preferences to Specific Features in Mobile Phones: A Comparative Study [Online] Available: http://ermt.net/docs/papers/Volume_6/5_May2017/V6N5-107.pdf.

[8] Multiple Linear Regression, [Online] Available http://www.statisticssolutions.com/assumptions-of-multiplelinear-regression/ [Accessed: 3-Jun-2017]

[9] Prediction of Survivors in Titanic Dataset: A Comparative Study using Machine Learning Algorithms Tryambak Chatterjee* Department of Management Studies, NIT Trichy, Tiruchirappalli, Tamilnadu, India