MACHINE LEARNING WITH PYTHON

Employment Turnover prediction

Training project report

Submitted by:

NAME	UNIVERSITY ROLL NO.	COLLEGE ROLL NO.
Saurabh Shaw	12616001147	1651140
Prity varma	10400217009	142
Nidhi kumari	10400216104	117

B.TECH IN INFORMATION TECHNOLOGY

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY,WEST BENGAL

JUNE,2018

CANDIDATE'S DECLARATION

I hereby declare that we have undertaken industrial training at "WEBTEK LABS" during a period from 14 JUNE to 14 JULY in partial fulfilment of requirements for the award of degree of B.TECH(COMPUTER SCIENCE AND TECHNOLOGY AND INFORMATION TECHNOLOGY) in HERITAGE INSTITUTE OF TECHNOLOGY, KOLKATA and INSTITUTE OF ENGINEERING MANAGEMENT,KOLKATA. The work which is being presented in the training report submitted to Department of COMPUTER SCIENCE AND TECHNOLOGY at HERITAGE INSTITUTE OF TECHNOLOGY, KOLKATA is an authentic record of training work.

Signature of the students

The FOUR weeks industrial training Viva-voice examination of
Has been held on and accepted.

Signature of Internal Examiner

Signature of External Examiner

CERTIFICATE OF APPROVAL

The project "**EMPLOYMENT TURNOVER PREDICTION**" made by **SAURABH SHAW ,NIDHI KUMARI AND PRITY VARMA** is hereby approved as a creditable study for the Bachelor of Technology in COMPUTER SCIENCE AND TECHNOLOGY and INFORMATION TECHNOLOGY only for the purpose for which it is submitted.

MS. MOUSITA DHAR
(Project In-charge)

ACKNOWLEDGEMENT

WE would like to express our special thanks of gratitude to our trainer MS.MOUSITA DHAR who gave us the golden opportunity to do this wonderful project on the topic EMPPLOYMENT TURNOVER PREDICTION BY MACHINE LEARNING USING PYTHON which also helped us in doing a lot of Research and we came to know about so many new things. We are really thankful to them.

Finally, we would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited time frame.

1.INTRODUCTION

1.1 PYTHON

■ Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

▪ PYTHON FEATURES

✓ **Easy** to Learn and Use. Python is **easy** to learn and use

✓ Expressive Language

✓ Interpreted Language

✓ Cross-**platform** Language

✓ Free and Open Source

✓ **Object-Oriented** Language

✓ Extensible

✓ Large Standard Library

- **APPLICATIONS OF PYTHON**
  - ✓ Web and internet develpoment
  - ✓ Scientific and numeric computing
  - ✓ Data Analysis
  - ✓ Desktop GUIs
  - ✓ Machine Learning
  - ✓ Data visualization
  - ✓ Game Deveplopment
  - ✓ Software Develpoment
  - ✓ Business Application

1.2 ANACONDA

Anaconda is a free and open distribution of Python programming languages for data science and machine learning related applications (large-scale data processing, predictive analytics, scientific computing), that aims to simplify package management and deployment. Package versions are managed by the package management system *conda* . Conda is an open source, cross platform, language-agnostic package manager and environment management system that installs, runs, and updates packages and their dependencies.The Anaconda distribution is used by over 6 million users, and it includes more than 250 popular data science packages suitable for Windows, Linux, and MacOS.

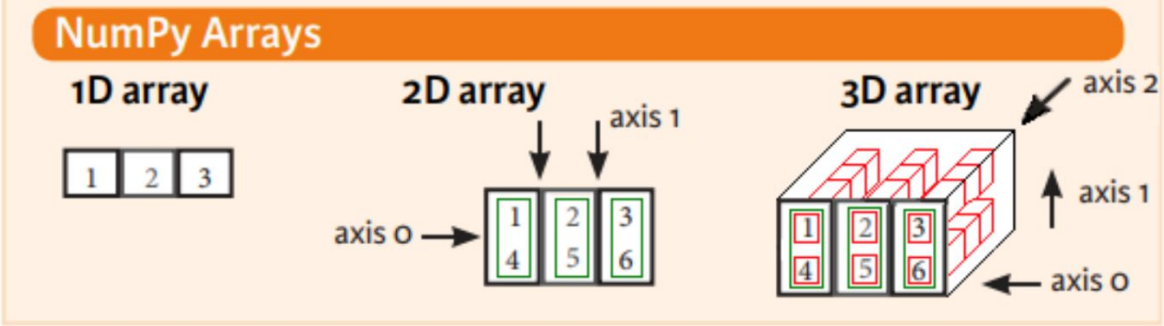
1.3 PYTHON PACKAGES

- **NUMPY**

✓ NumPy is the fundamental package for scientific computing with Python. It contains among other things:
✓ a powerful N-dimensional array object
✓ sophisticated (broadcasting) functions
✓ tools for integrating C/C++ and Fortran code
✓ useful linear algebra, Fourier transform, and random number

capabilities

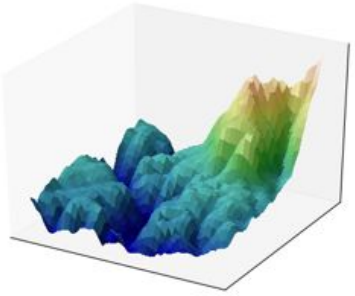
Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

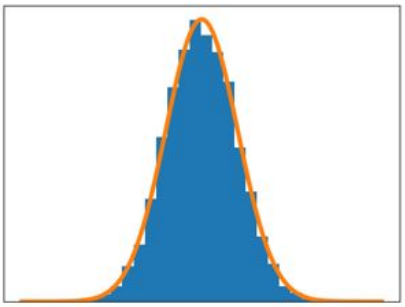


- **Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, the jupyter notebook, web application servers, and four graphical user interface toolkits.

Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code. For simple plotting the pyplot module provides a MATLAB-like interface, particularly when
combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.





- **Pandas**

Pandas is an open source, BSD-licensed library providing high-performance, easy- to-use data structures and data analysis tools for the *Python* programming language. Pandas library is well suited for data manipulation and analysis using python. In particular, it offers data structures and operations for manipulating numerical tables and time series.

- **Scikit-learn**

Scikit-learn provides machine learning libraries for python some of the features of Scikit- learn includes:

✓ Simple and efficient tools for data mining and data analysis
✓ Accessible to everybody, and reusable in various contexts
✓ Built on NumPy, SciPy, and matplotlib
✓ Open source, commercially usable - BSD license

TRAINING WORK UNDERTAKEN

- **COLLECTING DATA FROM KAGGLE**

Kaggle is a platform for predictive modelling and analytics competitions in which statisticians and data miners compete to produce the best models for predicting and describing the datasets uploaded by companies and users. This crowd sourcing approach relies on the fact that there are countless strategies that can be applied to any predictive modelling task and it is impossible to know beforehand which technique or analyst will be most effective.

On 8 March 2017, Google announced that they were acquiring Kaggle. They will join the Google Cloud team and continue to be a distinct brand. In January 2018, Booz Allen and Kaggle launched Data Science Bowl, a machine learning competition to analyze cell images and identify nuclei.

- **DATA SCIENCE**

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from data in various forms, both structured and unstructured, similar to data mining. Data science is a "concept to unify statistics, data analysis, machine learning and their related methods" in order to "understand and analyze actual phenomena" with data. It employs techniques and theories drawn from many fields within the context of mathematics, statistics, information science, and computer science.

Turing award winner Jim Gray imagined data science as a "fourth paradigm" of science (empirical, theoretical, computational and now data-driven) and asserted that "everything about science is changing because of the impact of information technology" and the data deluge.

When Harvard Business Review called it "The Sexiest Job of the 21st Century"the term became a buzzword, and is now often applied to business analytics, busines intelligence, predictive modeling, or any arbitrary use of data, or used as a glamorized term for statistics. In many cases, earlier approaches and solutions are now simply rebranded as "data science" to be more attractive, which can cause the term to become "dilute[d] beyond usefulness." While many university programs now offer a data science degree, there exists no consensus on a definition or suitable curriculum contents. Because of the current popularity of this term, there are many "advocacy efforts" surrounding the field. To its discredit, however, many data science and big data projects fail to deliver useful results, often as a result of poor management and utilization of resources.

- **DATASET**

The Datasets contains 14,999 rows and 13 columns.

▪ SOURCE CODE AND OUTPUT

Import Modules and load the data:-

```
import pandas as pd
hr=pd.read_csv('C:\\Users\\Shaw\\Desktop\\HR.csv')
col_names=hr.columns.tolist()
print("column names:")
print(col_names)
print("\nsample data:")
hr.head()
```

```
 print(hr.shape)
(14999, 10)
```

```
In [10]: hr
```

Out[10]:

satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	department	salary
0.38	0.53	2	157	3	0	1	0	sales	low
0.80	0.86	5	262	6	0	1	0	sales	medium
0.11	0.88	7	272	4	0	1	0	sales	medium
0.72	0.87	5	223	5	0	1	0	sales	low
0.37	0.52	2	159	3	0	1	0	sales	low
0.41	0.50	2	153	3	0	1	0	sales	low
0.10	0.77	6	247	4	0	1	0	sales	low
0.92	0.85	5	259	5	0	1	0	sales	low
0.89	1.00	5	224	5	0	1	0	sales	low
0.42	0.53	2	142	3	0	1	0	sales	low
0.45	0.54	2	135	3	0	1	0	sales	low

In this project, I checked and there is no missing value.
the columns of salary and department were converted to integer values which contained string values. These are then stored as

dummy columns .
The column which is not required were dropped.

ALGORITHMS USED:-

LINEAR REGRESSION

In **simple linear regression** a single independent variable is **used to** predict the value of a dependent variable. In multiple **linear regression** two or more independent variables are **used to** predict the value of a dependent variable. The difference between the two is the number of independent variables.

INPUT:-

In [21]:
```
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
rfe=RFE(model,10)
rfe=rfe.fit(hr[X],hr[y])
print(rfe.support_)
print(rfe.ranking_)
```

```
C:\Users\Shaw\Anaconda3\lib\site-packages\sklearn\utils\validation.py:578: DataConversionWarning: A column-vector y was passed
when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

```
[ True  True False False  True  True  True  True False  True  True False
 False False False  True  True False]
[1 1 3 9 1 1 1 1 5 1 1 6 8 7 4 1 1 2]
```

In [22]:
```
cols=['satisfaction_level','last_evaluation','time_spend_company','Work_accident','promotion_last_5years','department_RandD','dep
X=hr[cols]
y=hr['left']
```

In [23]:
```
from sklearn.cross_validation import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=0)

from sklearn.linear_model import LogisticRegression
from sklearn import metrics
logreg=LogisticRegression()
logreg.fit(X_train,y_train)
```

```
C:\Users\Shaw\Anaconda3\lib\site-packages\sklearn\cross_validation.py:41: DeprecationWarning: This module was deprecated in ver
sion 0.18 in favor of the model_selection module into which all the refactored classes and functions are moved. Also note that
the interface of the new CV iterators are different from that of this module. This module will be removed in 0.20.
  "This module will be removed in 0.20.", DeprecationWarning)
```

Out[23]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
```

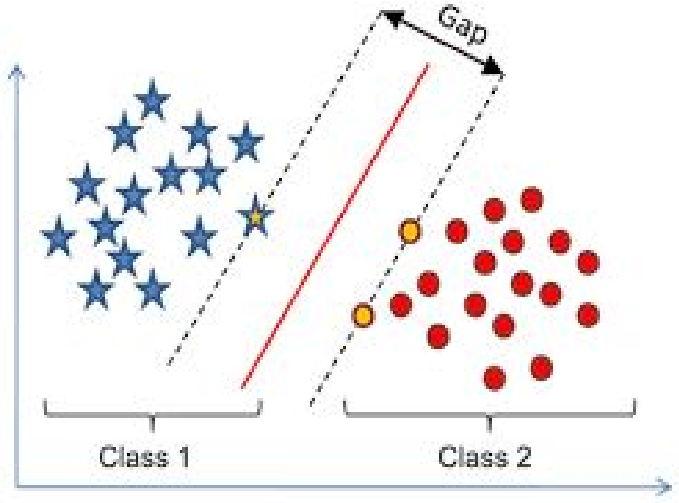
OUTPUT:-

```
In [24]: from sklearn.metrics import accuracy_score
         print('Logistic regression accuracy:{:.3f}'.format(accuracy_score(y_test,logreg.predict(X_test))))

         Logistic regression accuracy:0.771
```

- **SUPPORT VECTOR MACHINE** :

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (*supervised learning*), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimentional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.



INPUT:-

```
In [26]: from sklearn.svm import SVC
         svc = SVC()
         svc.fit(X_train, y_train)

Out[26]: SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
           decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
           max_iter=-1, probability=False, random_state=None, shrinking=True,
           tol=0.001, verbose=False)
```

OUTPUT:-

```
In [27]: print('Support vector machine accuracy: {:.3f}'.format(accuracy_score(y_test, svc.predict(X_test))))

         Support vector machine accuracy: 0.909
```

RESULTS AND DISCUSSION

- **RESULT**
  By using linear regression 77.1% prediction is obtain and 90.9% by using SVM.

CONCLUSION

As we used both linear regression and SVM algorithm, the better prediction result is opted by SVM algorithm which is 90.9% .
We can say that our prediction in most cases will give the accurate result.

REFERENCES

https://www.python.org/
https://anaconda.org/anaconda/python
http://www.numpy.org/
https://matplotlib.org/
http://scikit-learn.org/
https://pandas.pydata.org/
https://pandas.pydata.org/
https://ipython.org/