LOGIN / REGISTER (HTTPS://ID.ANALYTICSVIDHYA.COM/AUTH/LOGIN/?NEXT=HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/2020/11/DEMYSTIFYING-MODEL-INTERPRETATION-USING-ELI5/)

Analytics Vidhya (https://www.analyticsvidhya.com/blog/)

CLASSIFICATION (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/CLASSIFICATION/)
INTERMEDIATE (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/INTERMEDIATE/)
MACHINE LEARNING (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/MACHINE-LEARNING/)
PYTHON (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/PYTHON-2/)
STRUCTURED DATA (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/STRUCTURED-DATA/)
TECHNIQUE (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/TECHNIQUE/)

Demystifying Model Interpretation using ELI5

PRIYANKA870 (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/AUTHOR/PRIYANKA870/), NOVEMBER 10, 2020 LOGIN TO BOOKMARK TH…

Article 🔒 Video Book

This article was published as a part of the Data Science Blogathon (https://datahack.analyticsvidhya.com/contest/data-science-blogathon-ii/).

Introduction

Most of the machine learning and artificial intelligence models are popularly referred to as “**Black Boxes**”. Have you ever wondered why they are called so? Is it always good to have black box models? For the uninitiated, black-box models are those machines for which humans can only see the input and the output. We as humans are clueless about what the machine has done with the input to arrive at the said output.

In machine learning(ML) too, most of the time, we are not aware of how the machine has arrived at a particular solution but we are aware of the accuracy of the model. So, we are left with two choices: either we can just trust the machine and accept the solution or take a deep dive into figuring out why the machine is arriving at a particular solution? Let’s take the latter route and try to interpret a machine learning model.

But before we start, let’s answer a very critical question: why is it necessary to interpret a machine learning model when it already has high accuracy? Why not just trust it already?

Why is the interpretability of ML models so important?

In ML, there is always a **trade-off between model interpretation and accuracy**. And which one to choose always depends on: **What has to be predicted**? For example, we don’t care how a movie recommendation system makes recommendations given that it has good accuracy. However, things will be a bit different when a highly accurate machine learning model makes predictions on what kind of drug should be administered to a patient.

In the latter case, we have very little appetite for a wrong prediction when compared to the former. So it becomes necessary to check how the model is arriving at the solution. Model interpretation is also important for presenting our solution to business people, who don’t care about our code and the fancy loss functions that we have used. They just care about if the model solves the problem.

Having discussed why interpretability is so important, let’s go ahead and try to get some hands-on experience in a very popular model interpretation tool called ELI5.

What is ELI5?

In short, ELI5 is a python package that is used to inspect ML classifiers and explain their predictions. It is popularly used to **debug algorithms** such as sklearn regressors and classifiers, XGBoost, CatBoost, Keras, etc. You can read more about the package here (https://github.com/TeamHG-Memex/eli5).

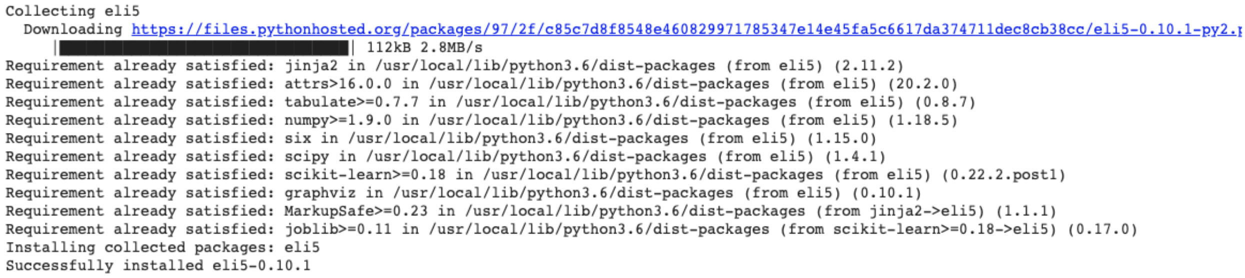
ELI5 Model Interpretation

To begin with, we need a dataset and a problem. For the purpose of this article, we will be using the social_network_ads (https://www.kaggle.com/akram24/social-network-ads) dataset. Here, we will try to predict whether a user has purchased a product by clicking on the advertisements shown to it on social networks, based on its gender, age, and estimated salary. So without further ado, let’s get started.

Step 1: Install ELI5

If you already do not have the ELI5 package installed on your system, install it using :

```
pip install eli5
```



```
Collecting eli5
  Downloading https://files.pythonhosted.org/packages/97/2f/c85c7d8f8548e460829971785347e14e45fa5c6617da374711dec8cb38cc/eli5-0.10.1-py2.p
     |████████████████████████████████| 112kB 2.8MB/s
Requirement already satisfied: jinja2 in /usr/local/lib/python3.6/dist-packages (from eli5) (2.11.2)
Requirement already satisfied: attrs>16.0.0 in /usr/local/lib/python3.6/dist-packages (from eli5) (20.2.0)
Requirement already satisfied: tabulate>=0.7.7 in /usr/local/lib/python3.6/dist-packages (from eli5) (0.8.7)
Requirement already satisfied: numpy>=1.9.0 in /usr/local/lib/python3.6/dist-packages (from eli5) (1.18.5)
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (from eli5) (1.15.0)
Requirement already satisfied: scipy in /usr/local/lib/python3.6/dist-packages (from eli5) (1.4.1)
Requirement already satisfied: scikit-learn>=0.18 in /usr/local/lib/python3.6/dist-packages (from eli5) (0.22.2.post1)
Requirement already satisfied: graphviz in /usr/local/lib/python3.6/dist-packages (from eli5) (0.10.1)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.6/dist-packages (from jinja2->eli5) (1.1.1)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.6/dist-packages (from scikit-learn>=0.18->eli5) (0.17.0)
Installing collected packages: eli5
Successfully installed eli5-0.10.1
```

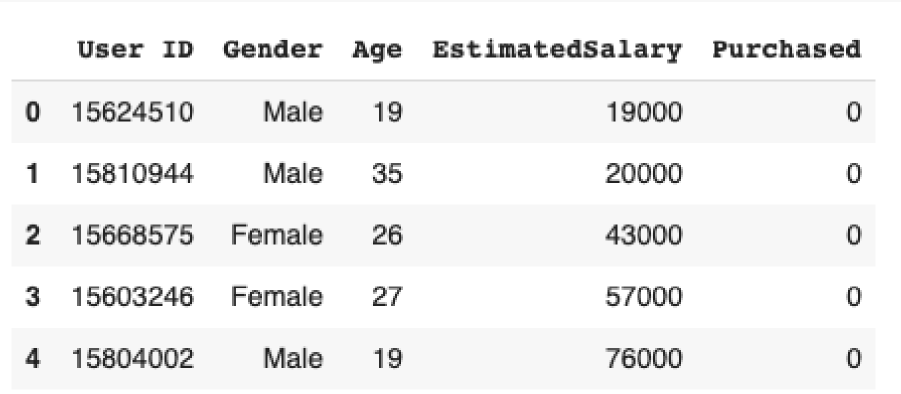
Once you have installed the package, we are all set to work with it.

Step 2: Import the important libraries

```
import pandas as pd

import numpy as np

import seaborn as sns

import eli5 as eli

from sklearn.tree import DecisionTreeClassifier

from sklearn.preprocessing import LabelEncoder

from sklearn.model_selection import train_test_split
```

Step 3: Import the dataset

```
data = pd.read_csv(r'/content/Social_Network_Ads.csv')

data.head()
```

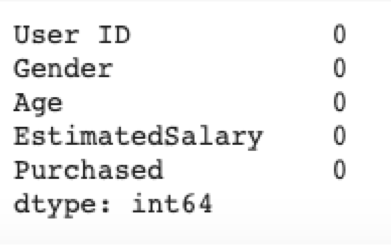


	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0

Step 4: Data preparation and preprocessing

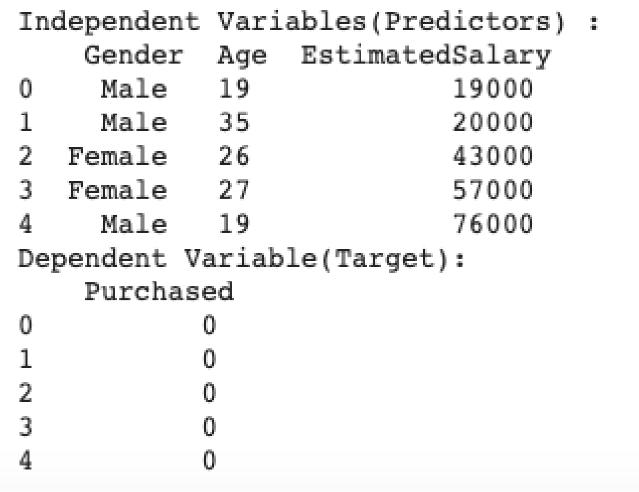
In this step, we have performed some basic preprocessing steps such as checking for NAs, splitting the dataset into inputs and outcome, training and testing sets, and label encoding categorical variables.

```
data.isna().sum()
```



```
User ID            0
Gender             0
Age                0
EstimatedSalary    0
Purchased          0
dtype: int64
```

```
#split the dataset into independent and dependent variables

X = data.iloc[:, [1, 2, 3]]

y = data.iloc[:,[4]]

print("Independent Variables(Predictors) :n", X.head(), "nDependent Variable(Target):n", y.head())
```



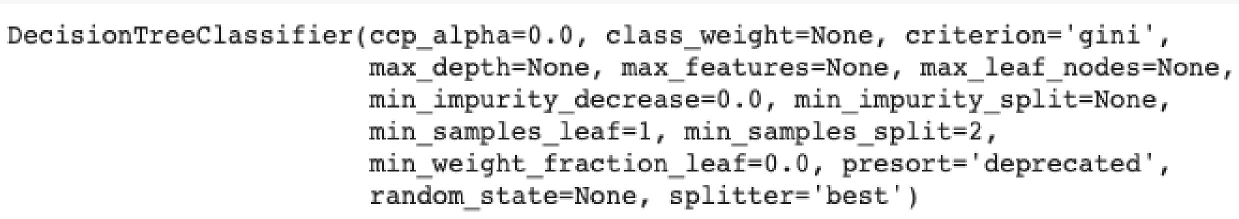
```
Independent Variables(Predictors) :
    Gender  Age  EstimatedSalary
0     Male   19            19000
1     Male   35            20000
2   Female   26            43000
3   Female   27            57000
4     Male   19            76000
Dependent Variable(Target):
    Purchased
0           0
1           0
2           0
3           0
4           0
```

```
encoder = LabelEncoder()

X.Gender = encoder.fit_transform(X.Gender)

# Splitting the dataset into the Training set and Test set (75:25)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25)
```

Once we have prepared the data, we are all set to train an ML model to make predictions. In this case, we are using a decision tree classifier.

Step 5: Model Training

```
# Decision Tree Classifier

classifier_dtc = DecisionTreeClassifier()

classifier_dtc.fit(X_train, y_train)
```



```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=None, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
```

Step 6: Model Testing

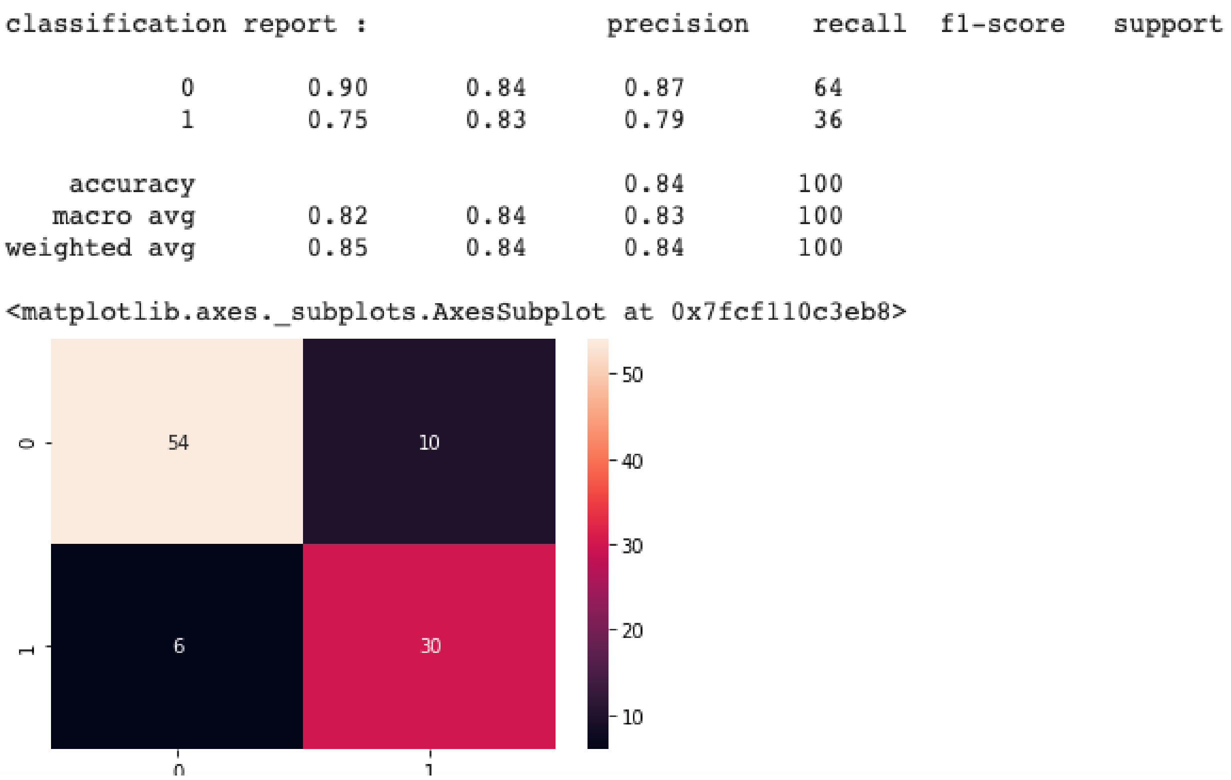
After training the model, test it on the 25% of the dataset that we kept aside for testing.

```
# Predicting the Test set results

y_pred = classifier_dtc.predict(X_test)
```

Step 7: Model Evaluation

Predictions have been made, now it’s time for model evaluation. We are using a confusion matrix here. Each row in a confusion matrix corresponds to the actual class while each column corresponds to a predicted class.

```
# Making the Confusion Matrix

from sklearn.metrics import confusion_matrix

from sklearn.metrics import classification_report

clf_rpt = classification_report(y_test,y_pred)

print("classification report :", clf_rpt)

cm = confusion_matrix(y_test, y_pred)

sns.heatmap(cm,annot=True)
```



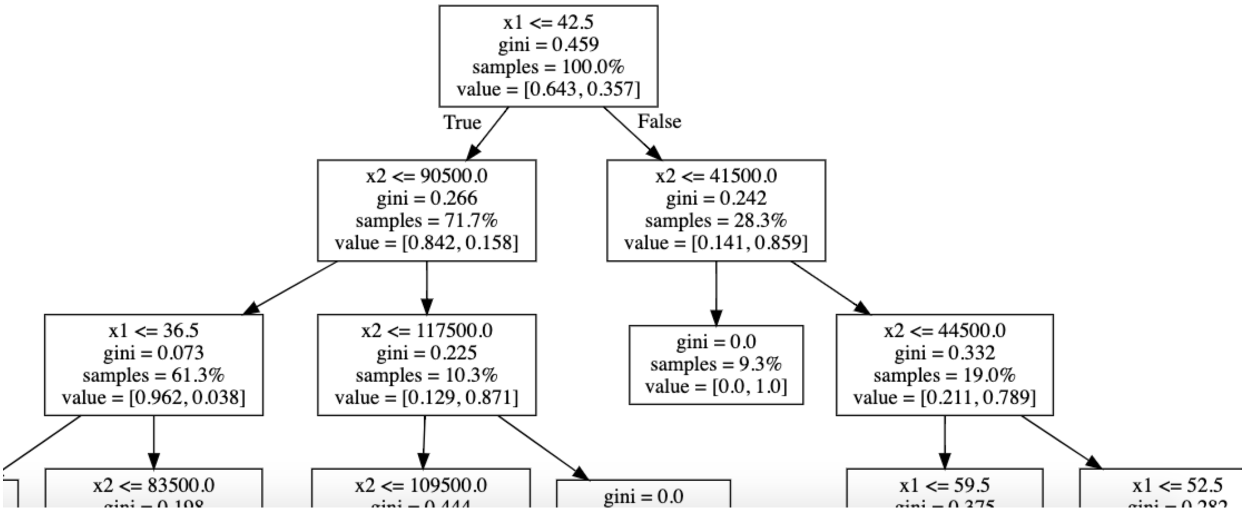
```
classification report :               precision    recall  f1-score   support

           0       0.90      0.84      0.87        64
           1       0.75      0.83      0.79        36

    accuracy                           0.84       100
   macro avg       0.82      0.84      0.83       100
weighted avg       0.85      0.84      0.84       100

<matplotlib.axes._subplots.AxesSubplot at 0x7fcf110c3eb8>
```

As can be seen from the classification report, the model is 84% accurate. But we want to know how the model is coming to this conclusion. Hence, let’s go ahead and try to use ELI5 to get some answers.

Step 8: Model Interpretation

To start with, we can use explain_weights() to find the weight given to each feature in prediction. For tree-based models, ELI5 does nothing new for calculating feature weights. It simply uses the GINI index used for preparing decision trees as weights.

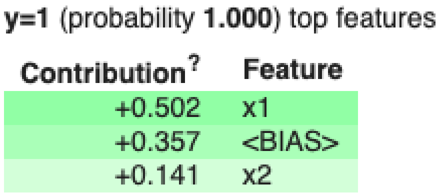
```
eli.explain_weights(classifier_dtc)
```



Here is a portion of the decision tree with feature weights.

Suppose if we wish to debug and check how the outcome has been predicted for a particular row. We can use explain_prediction().

```
eli.explain_prediction(classifier_dtc , np.array(X_test)[1])
```

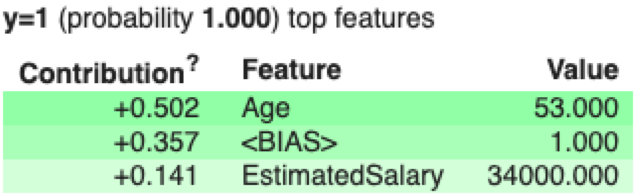


y=1 (probability **1.000**) top features

Contribution?	Feature
+0.502	x1
+0.357	<BIAS>
+0.141	x2

As can be observed from the above output, eli5 shows us the contribution of each feature in predicting the output. If you further wish to see and compare what combination of features and values lead to a particular prediction, we can use show_prediction().

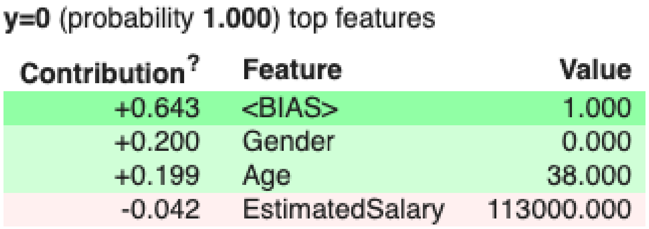
```
eli.show_prediction(classifier_dtc, X_test.iloc[1],
                    feature_names=list(X.columns),
                    show_feature_values=True)
```



y=1 (probability **1.000**) top features

Contribution?	Feature	Value
+0.502	Age	53.000
+0.357	<BIAS>	1.000
+0.141	EstimatedSalary	34000.000

```
eli.show_prediction(classifier_dtc, X_test.iloc[20],
                    feature_names=list(X.columns),
                    show_feature_values=True)
```



y=0 (probability **1.000**) top features

Contribution?	Feature	Value
+0.643	<BIAS>	1.000
+0.200	Gender	0.000
+0.199	Age	38.000
-0.042	EstimatedSalary	113000.000

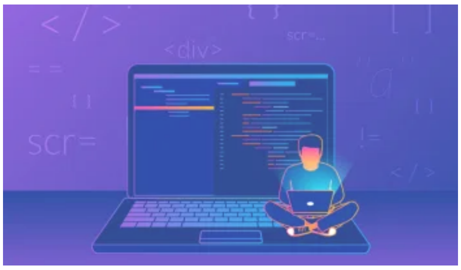
Yeah! you guess it right. In ELI5, a prediction is basically the sum of positive features inclusive of bias.

WRAP UP

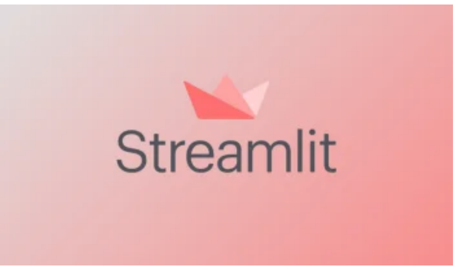
These were just a few examples of debugging using ELI5. There are many other model interpretation frameworks such as Skater (https://github.com/oracle/Skater) and SHAP (https://github.com/slundberg/shap). Be Sherlock !! go ahead and try to inspect and debug the machine learning models that make you ponder over the outcomes.

Related Articles

(https://www.analyticsvidhya.com/blog/2021/01/explain-how-your-model-works-using-explainable-ai/)
Explain How Your Model Works Using Explainable AI (https://www.analyticsvidhya.com/blog/2021/01/explain-how-your-model-works-using-explainable-ai/)



(https://www.analyticsvidhya.com/blog/2021/02/introduction-to-machine-learning-for-absolute-beginners/)
Introduction to Machine Learning for Absolute Beginners (https://www.analyticsvidhya.com/blog/2021/02/introduction-to-machine-learning-for-absolute-beginners/)



(https://www.analyticsvidhya.com/blo machine-learning-models-using-streamlit-an-introductory-guide-to-model-deployment/)
Deploying machine learning models using Streamlit - An introductory guide to Model Deployment (https://www.analyticsvidhya.com/blog/2020/12/deploying-machine-learning-models-using-streamlit-an-introductory-guide-to-model-deployment/)

TAGS : BLOGATHON (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/BLOGATHON/), ELI5 (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/ELI5/)

NEXT ARTICLE

Text Mining Simplified – IPL 2020 Tweet Analysis with R

(https://www.analyticsvidhya.com/blog/2020/11/text-mining-simplified-ipl-2020-tweet-analysis-with-r/)

•••

PREVIOUS ARTICLE

Generate Carnatic Music with Google Magenta

(https://www.analyticsvidhya.com/blog/2020/11/generate-carnatic-music-with-google-magenta/)

(https://www.analyticsvidhya.com/blog/author/priyanka870/)

Priyanka870 (Https://Www.Analyticsvidhya.Com/Blog/Author/Priyanka870/)

LEAVE A REPLY

Your email address will not be published.

Comment

Name (required)

Email (required)

Website

□ Notify me of new posts by email.

SUBMIT COMMENT

POPULAR POSTS

- Commonly used Machine Learning Algorithms (with Python and R Codes) (https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/)
- Introductory guide on Linear Programming for (aspiring) data scientists (https://www.analyticsvidhya.com/blog/2017/02/lintroductory-guide-on-linear-programming-explained-in-simple-english/)
- 40 Questions to test a data scientist on Machine Learning [Solution: SkillPower – Machine Learning, DataFest 2017] (https://www.analyticsvidhya.com/blog/2017/04/40-questions-test-data-scientist-machine-learning-solution-skillpower-machine-learning-datafest-2017/)
- 6 Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R (https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/)

40 Questions to test a Data Scientist on Clustering Techniques (Skill test Solution) (https://www.analyticsvidhya.com/blog/2017/02/test-data-scientist-clustering/)

45 Questions to test a data scientist on basics of Deep Learning (along with solution) (https://www.analyticsvidhya.com/blog/2017/01/must-know-questions-deep-learning/)

30 Questions to test a data scientist on Linear Regression [Solution: Skilltest – Linear Regression] (https://www.analyticsvidhya.com/blog/2017/07/30-questions-to-test-a-data-scientist-on-linear-regression/)

Understanding Support Vector Machine(SVM) algorithm from examples (along with code) (https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/)

CAREER RESOURCES

16 Key Questions You Should Answer Before Transitioning into Data Science (https://www.analyticsvidhya.com/16-key-questions-data-science-career-transition/?&utm_source=Blog&utm_medium=CareerResourceWidget)

NOVEMBER 23, 2020

Here's What You Need to Know to Become a Data Scientist! (https://www.analyticsvidhya.com/blog/2021/01/heres-what-you-need-to-know-to-become-a-data-scientist/?&utm_source=Blog&utm_medium=CareerResourceWidget)

JANUARY 22, 2021

These 7 Signs Show you have Data Scientist Potential! (https://www.analyticsvidhya.com/blog/2020/12/these-7-signs-show-

you-have-data-scientist-potential/?&utm_source=Blog&utm_medium=CareerResourceWidget)

DECEMBER 3, 2020

How To Have a Career in Data Science (Business Analytics)? (https://www.analyticsvidhya.com/blog/2020/11/how-to-have-a-career-in-data-science-business-analytics/?&utm_source=Blog&utm_medium=CareerResourceWidget)

NOVEMBER 26, 2020

Should I become a data scientist (or a business analyst)? (https://www.analyticsvidhya.com/blog/2020/11/become-data-scientist-business-analyst/?&utm_source=Blog&utm_medium=CareerResourceWidget)

NOVEMBER 24, 2020

RECENT POSTS

Common terminologies used in Machine Learning and Artificial Intelligence (https://www.analyticsvidhya.com/blog/2021/03/common-terminologies-used-in-machine-learning-and-artificial-intelligence/)

MARCH 20, 2021

Popular Python Data Structures: Comparison & Operations (https://www.analyticsvidhya.com/blog/2021/03/popular-python-data-structures-comparison-operations/)

MARCH 20, 2021

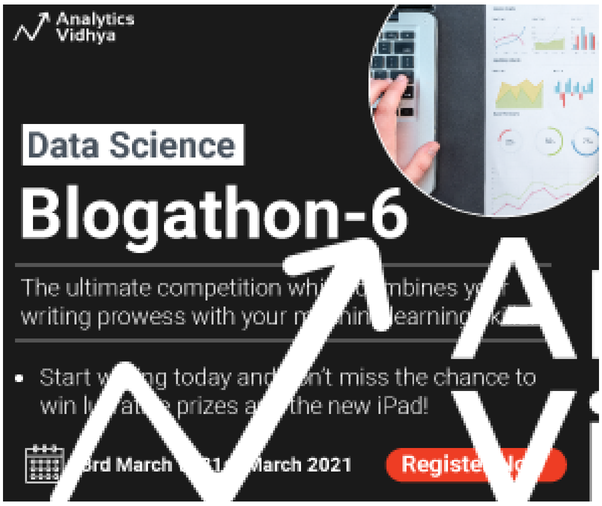
Introducing Machine Learning for Spatial Data Analysis (https://www.analyticsvidhya.com/blog/2021/03/introducing-machine-learning-for-spatial-data-analysis/)

MARCH 20, 2021

Introduction to The Architecture of Alexnet (https://www.analyticsvidhya.com/blog/2021/03/introduction-to-the-architecture-of-alexnet/)

MARCH 19, 2021

Analytics Vidhya
Your Journey to Become an
AI & ML Professional Starts Here!
- 75+ Mentorship Sessions
- 50+ Real-World Projects
- Interview Preparation
Certified AI & ML BlackBelt Plus Program
Know More


Analytics Vidhya
Data Science
Blogathon-6
The ultimate competition wh[illegible] mbines y[illegible]
writing prowess with your m[illegible]hin[illegible]learning[illegible]
Start w[illegible]g today and [illegible]n't miss the chance to
win l[illegible]at[illegible] prizes a[illegible] the new iPad!
[illegible]rd March [illegible]1[illegible]March 2021
Registe[illegible]

(https://www.analyticsvidhya.com/)

Download App (https://play.google.com/store/apps/details?id=com.analyticsvidhya.android) (https://apps.apple.com/us/app/analytics-vidhya/id1470025572)

Analytics Vidhya

About Us (https://www.analyticsvidhya.com/about-me/)

Our Team (https://www.analyticsvidhya.com/about-me/team/)

Careers (https://www.analyticsvidhya.com/about-me/career-analytics-vidhya/)

Contact us (https://www.analyticsvidhya.com/contact/)

Data Science

Blog (https://www.analyticsvidhya.com/blog/)

Hackathon (https://datahack.analyticsvidhya.com/)

Discussions (https://discuss.analyticsvidhya.com/)

Apply Jobs (https://www.analyticsvidhya.com/jobs/)

Companies

Post Jobs (https://www.analyticsvidhya.com/corporate/)

Trainings (https://courses.analyticsvidhya.com/)

Hiring Hackathons (https://datahack.analyticsvidhya.com/)

Advertising (https://www.analyticsvidhya.com/contact/)

Visit us

(https://www.facebook.com/AnalyticsVidhya/) (https://www.linkedin.com/company/analytics-vidhya/) (https://www.youtube.com/channel/UCH6gDteHtH4hg3o2343iObA) (https://twitter.com/analyticsvidhya)

Privacy Policy Terms of Use Refund Policy