Registration number: 2107601

Project: Causal Inference

Link to GitHub:
https://github.com/Utsav489/Causal-inference

MACHINE LEARNING FOR CAUSAL INFERENCE FROM OBSERVATIONAL DATA

Table of Contents

Introduction

The identified focus for the context of the particular investigation operation are execution in terms of the exploration operation for the context of the "***CI benchmark dataset***", identifying the characteristics for the distinct system and also the relevant "***performance metrics***". For the instances of the particular model, the operation in terms of the developments of an intuition has been performed, regarding how the "***CI problems***" manage to be correlated in the context of the orthodox "***machine learning jobs***".

Executive Summary

Implementation in terms of the exploration operation in terms of the "CI benchmark dataset," identification of system characteristics for a specific system, and also the relevant "performance metrics" has been identified as the primary focus areas for the context of the specific investigation operation. Regarding the "CI difficulties" that manage to be correlated within the context of traditional "machine learning tasks," the operation has been carried out in terms of the develops of an intuition for each individual instance of the particular model.

Data

The propensity score represents the likelihood that a person will be allocated to the "intervention groups" depending on their visible extracted features, which is , e(x) = P(t = 1|x). Their genetic makeup. Implementing "inverse propensity score" weighting is accomplished by dividing the number of samples even by the reverse of the "***propensity score***". Alternatively, a sample weight wi for integer may be calculated as follows:

$$w_i = \frac{t_i}{e(x_i)} + \frac{1 - t_i}{1 - e(x_i)}$$

All variables offered here are primarily concerned with determining "prediction errors". To represent them in the specified format, they are denoted as X, where X represents the amount of inaccuracy committed with regard to the forecast type X. In the context of "treatment outcomes", Y (i) t and yˆ (i) t mainly manages to denotes true as well as the "predicted outcomes" respectively for the context of "treatment t" and also the "individual i". Thus, following the definition of ITE (Eq. (2)), the difference Y (i) 1 − Y(i) 0 provides a "true effect", whereas yˆ (i) 1 − yˆ (i) 0 can be signified as the predicted one. Following this, definition can be provided for the "Precision in Estimation of Heterogeneous Effect (PEHE)", which can be identified as the "root mean squared error" between predicted and "true effects":

$$\epsilon_{PEHE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_1^{(i)} - \hat{y}_0^{(i)} - (\mathcal{Y}_1^{(i)} - \mathcal{Y}_0^{(i)}))^2}$$

Methodology

For the implementation of the particular model, the "***random forest model***" has been considered, due to this distinct model mainly manages to perform both the "***classification***" as well as the "***regression***" operation. From the execution of the "***10-fold cross-validation***" operation, the identified value is 1.4662949845656912. The "***weighting***" can be performed for a distinct function due to the implementation of the layering in a function. An "***Evaluation Metrics***" evaluates the effectiveness of a forecasting model. This often entails training a classifier on a population, using a system to generate forecasts on an outlier information that

cannot be used while development, then evaluating the forecasts to the estimated parameters in the outlier data frame.

For the implementation of the particular operation, two different set in terms of the information have been gathered, like the "***IHDP***" as well as "***JOBS***". Using the "Infant Health Development Program (IHDP)" information set, researchers were able to examine the impact of high-quality childcare and regular visits on the future cognitive test scores of low birth weight and preterm children (Raschka *et al.* 2020).

Results

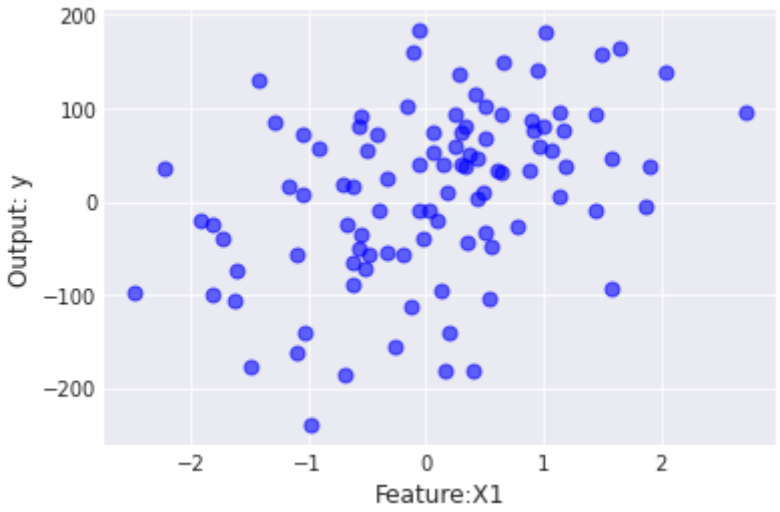


Figure 1: Output for IHDP dataset

(Source: Acquired from Python)

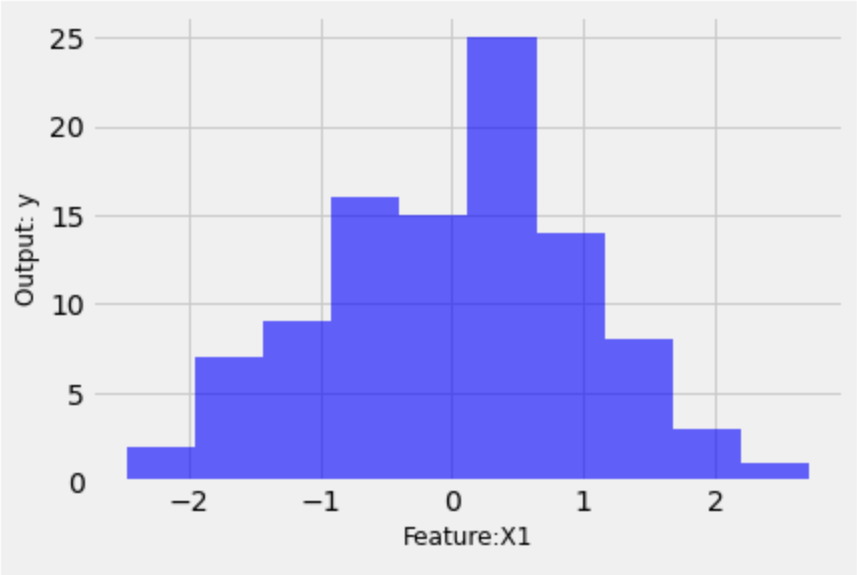


Figure 2: Output for IHDP dataset

(Source: Acquired from Python)

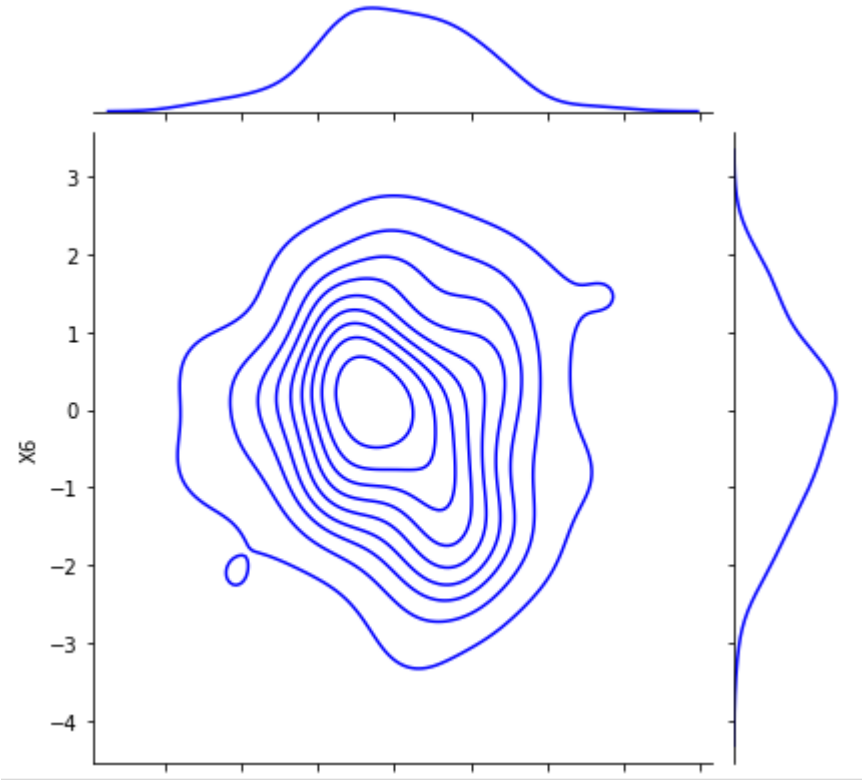


Figure 3: Output for IHDP dataset

(Source: Acquired from Python)

Discussion

Among the dataset's 25 features are measurements of the child (such as kid mass, circumference measurements, days straight birthed preterm, birth order, first born, health records benchmark, sex...), documentation more about mother at the time of delivery (such as age, relationship status, educational achievement), and details first about mother's actions during the pregnancy (such as smoking, drinking, and sleeping)

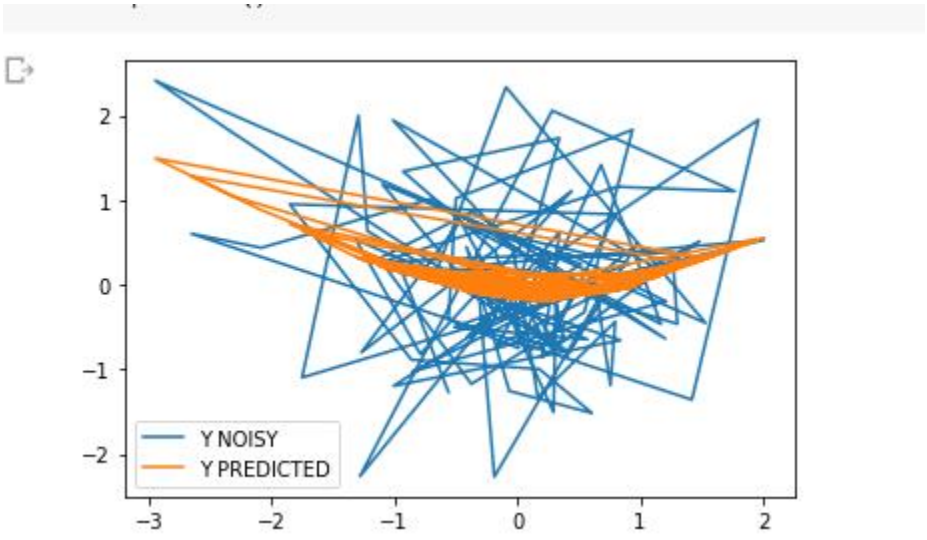


Figure 4: Output for IHDP dataset

(Source: Acquired from Python)

These are background variables, abbreviated as X. If a family has been in the comparison group (i.e., t = 0, no help was offered) or in the treatment group (i.e., t = 1, assistance was offered) is indicated by the treatment variable (t). The score on the cognitive test for the youngster is recorded in the result column. A clinical study was used to develop the dataset that was used to introduce it.

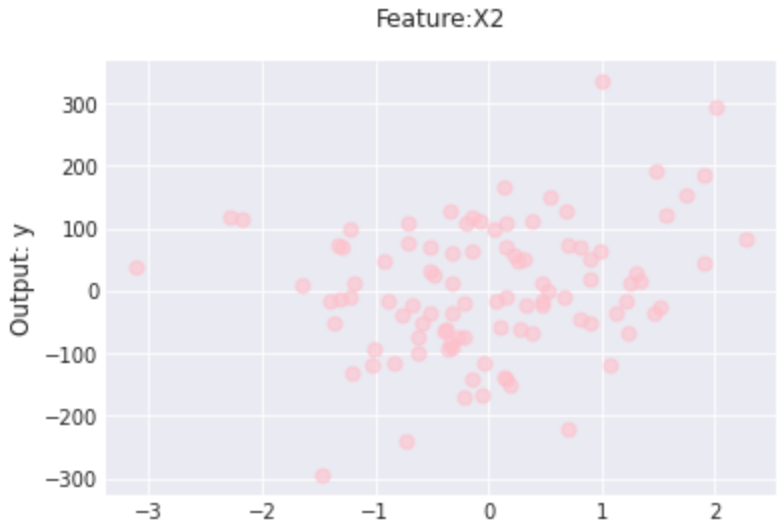


Figure 5: Output for IHDP dataset

(Source: Acquired from Python)

Depending on genuine pre-treatment variables, we employ a semi-synthetic copy of the data in which the outcomes (both true and false) are generated (with some random noise introduced) based on actual which was before covariates (Zhang *et al.* 2021). In order to account for this, the information also provides real (noiseless) personalized effects for every data packet, which are more suited for performance assessment than outcomes owing to the absence of noise than the outcomes.

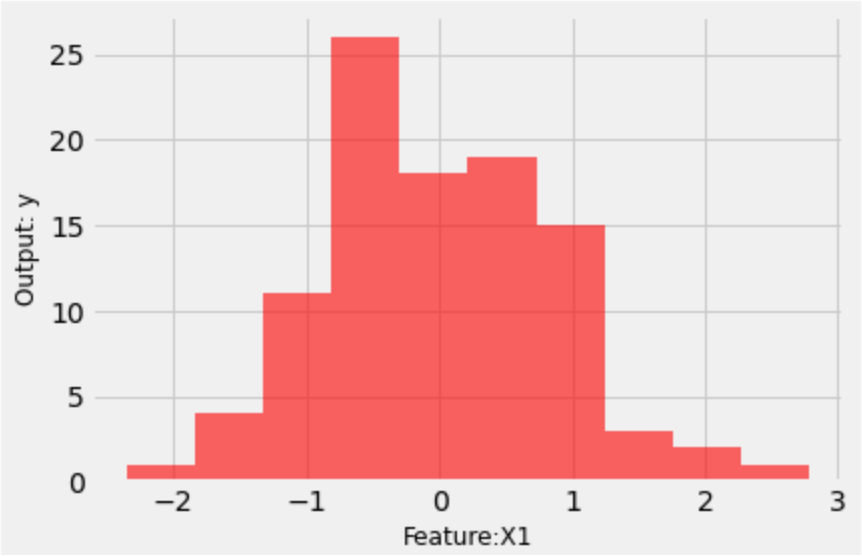


Figure 6: Output for IHDP dataset

(Source: Acquired from Python)

In the training stage, the usage of any counterfactuals/true consequences is strictly prohibited. For the instances of the information set "***JOBS***", As presented by, this collection incorporates information from an investigation carried out as part of a "Comprehensive Supported Work Program (NSWP)" with empirical evidence as from "Panel Study of Income Dynamics (PSID)".

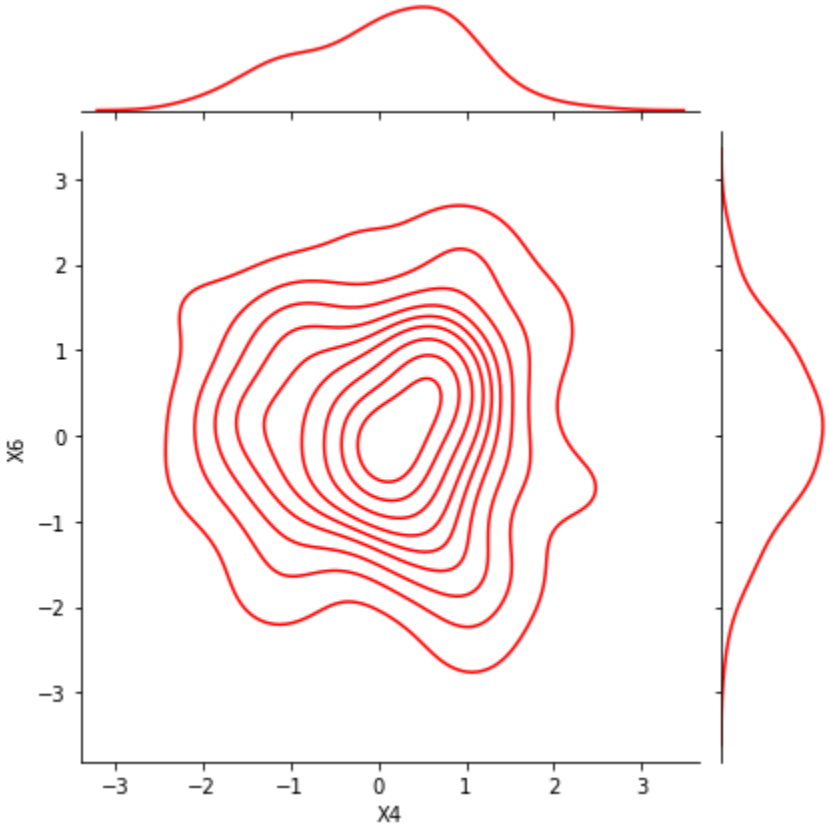


Figure 7: Output for IHDP dataset

(Source: Acquired from Python)

Overall, the dataset contains people's fundamental features via 17 variables that include their history, whether or not they got career counselling from NSWP (treatment), including their work status (outcome). It is indicated in column, whether a sample was obtained by exploratory or conventional collection of data; nonetheless, all samples should have been considered inside the model development.

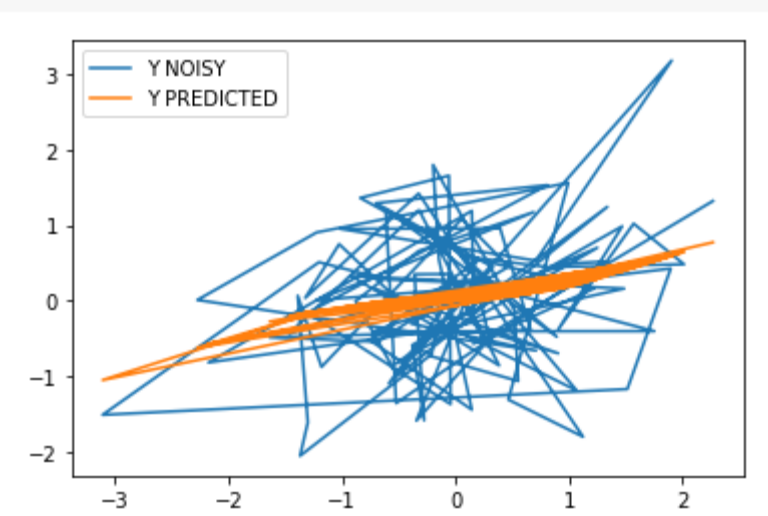


Figure 8: Output for IHDP dataset

(Source: Acquired from Python)

"Weighted regression" is a technique that you may apply whenever the least squares condition of constant variance in the residuals is broken (heteroscedasticity). With the "suitable weight", this process reduces the total of "weighted squared residuals" to create residuals with an "***equal variances (homoscedasticity)***". Whenever the witness defects do not even have an equal variances and the homogeneity of variance condition of "***linear regression***" is violated, balanced linear regression can be used instead of "***linear regression***" (Babar *et al.* 2020). The most significant disadvantage of weighted linear regression is that it is very dependent on the "***covariance matrix***" of the interpretation error as a predictor of the outcome.

Conclusion

In the execution of the particular operation, the importation of the "***dataset***" has been identifiers as the significant segment for the particular experiment. For the instances of the particular operation, the regression operation has been performed, which assists in highlighting the performances for the context of the two distinct models. For the instances of the particular experiment, at the initial segment the importation operation regarding the "***dataset***" has been performed, in which the "***pandas***" library has been utilized.

Reference List

Babar, B., Luppino, L.T., Boström, T. and Anfinsen, S.N., 2020. Random forest regression for improved mapping of solar irradiance at high latitudes. *Solar Energy*, *198*, pp.81-92.

Raschka, S., Patterson, J. and Nolet, C., 2020. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *Information*, *11*(4), p.193.

Sanni, R.R. and Guruprasad, H.S., 2021. Analysis of performance metrics of heart failured patients using Python and machine learning algorithms. *Global Transitions Proceedings*, *2*(2), pp.233-237.

Tran, M.K., Panchal, S., Chauhan, V., Brahmbhatt, N., Mevawalla, A., Fraser, R. and Fowler, M., 2022. Python-based scikit-learn machine learning models for thermal and electrical performance prediction of high-capacity lithium-ion battery. *International Journal of Energy Research*, *46*(2), pp.786-794.Zhang, J., Ma, G., Huang, Y., Aslani, F. and Nener, B., 2019. Modelling uniaxial compressive strength of lightweight self-compacting concrete using random forest regression. *Construction and Building Materials*, *210*, pp.713-719.

Zhang, W., Wu, C., Li, Y., Wang, L. and Samui, P., 2021. Assessment of pile drivability using random forest regression and multivariate adaptive regression splines. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, *15*(1), pp.27-40.

APPENDIX

```
[10] import pandas as pd
```

Dataset **1**

```
[13] df = pd.read_csv("/content/ihdp.csv")
     df.head()
```

	x4	x5	x6	x7	x8	x9	x10	...	x20	x21	x22	x23	x24	x25	t	yf	ycf	ite
879606	0.308569	-1.023402	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	1.0	1.0	4.771232	-0.298509	4.657928
161703	-0.629189	1.460832	1.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.956273	5.783770	3.428604
161703	-0.629189	0.963985	1.0	0.0	1.0	1.0	...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	4.164164	7.055789	3.658195
879606	0.371086	-0.692171	1.0	0.0	0.0	0.0	...	1.0	0.0	0.0	0.0	0.0	0.0	1.0	6.172307	1.379697	4.585505
879606	0.558638	0.301522	0.0	1.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	1.0	7.834469	2.747986	4.265591

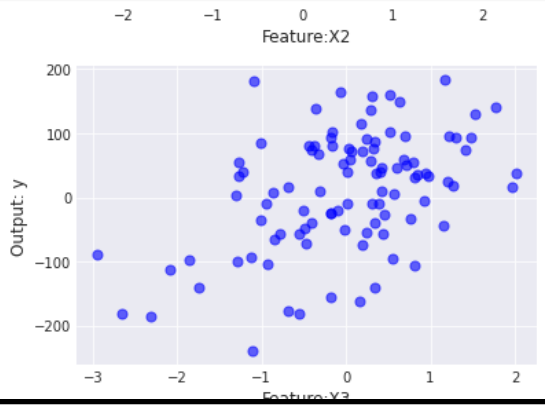
```
[14] import numpy as np

     def pehe(effect_true, effect_pred):
         """
         Precision in Estimating the Heterogeneous Treatment Effect (PEHE)
         :param effect_true: true treatment effect value
         :param effect_pred: predicted treatment effect value
         :return: PEHE
         """
         # This function should be completed as part of Unit 4.

     def abs_ate(effect_true, effect_pred):
         """
         Absolute error for the Average Treatment Effect (ATE)
         :param effect_true: true treatment effect value
         :param effect_pred: predicted treatment effect value
         :return: absolute error on ATE
         """
         # This function should be completed as part of Unit 4.

     def abs_att(effect_pred, yf, t, e):
         """
         Absolute error for the Average Treatment Effect on the Treated
```

```
[28] with plt.style.context(('seaborn-dark')):
         for i,col in enumerate(df.columns[:-1]):
             plt.figure(figsize=(6,4))
             plt.grid(True)
             plt.xlabel('Feature:'+col,fontsize=12)
             plt.ylabel('Output: y',fontsize=12)
             plt.scatter(df[col],df['y'],c='blue',s=50,alpha=0.6)
```

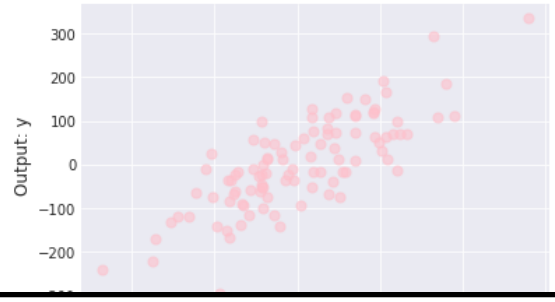



```
[72] import pandas as pd

[73] df = pd.read_csv("/content/jobs.csv")
     df.head()
```

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13
0	-0.614282	1.464727	0.0	0.0	1.0	0.0	2.393250	2.746196	-0.653311	-0.656913	1.627531	2.462337	2.937244
1	-0.802463	0.101835	0.0	0.0	1.0	0.0	0.109885	0.498271	-0.785284	-0.743407	-0.022502	-0.177193	0.082537
2	-0.896553	-0.238888	1.0	0.0	1.0	1.0	-0.085212	-0.148097	-0.847312	-0.781606	-0.361348	-0.286221	-0.303615
3	-0.896553	-0.238888	0.0	0.0	0.0	1.0	0.405581	0.325594	-0.847312	-0.781606	-0.361348	0.023020	-0.039630
4	0.138440	-1.601779	1.0	0.0	1.0	1.0	-0.722531	-0.212734	-0.019840	-0.156019	-1.422084	-0.514563	-0.331552

```
[82] with plt.style.context(('seaborn-dark')):
         for i,col in enumerate(df.columns[:-1]):
             plt.figure(figsize=(6,4))
             plt.grid(True)
             plt.xlabel('Feature:'+col,fontsize=12)
             plt.ylabel('Output: y',fontsize=12)
             plt.scatter(df[col],df['y'],c='pink',s=50,alpha=0.6)
```




```
[84] with plt.style.context(('fivethirtyeight')):
         for i,col in enumerate(df.columns[:-1]):
             plt.figure(figsize=(6,4))
             plt.grid(True)
             plt.xlabel('Feature:'+col,fontsize=12)
             plt.ylabel('Output: y',fontsize=12)
             plt.hist(df[col],alpha=0.6,facecolor='r')
```