|  |  |
| --- | --- |
| Exam preparation mcq: | <https://www.programsbuzz.com/clustering-mcq> |
|  |  |
| Standard error | Standard error measures the variability in the estimate for these co-efficient. |
|  | Linear regression is a form of parametric regression.  Parametric regression:   * data follows fixed parameters   Non-parametric regression:   * Data does not follow fixed parameters. * Dynamic in nature * Decision tree is of this example |
| Prediction vs projection | Prediction:   * Focus on driver fields * No assumption, linear regression * Simple models are better than complex models (for ex: business peoples make decision) – Accuracy is not that important as long we can understand parameter behaviour   Projection:   * Focus on outcome * Forecast assumes that conditions remain the same as they were when the model was built * If parameters change then outcome change. (for ex: company GDP growth depends on political party, if party changes outcome changes) * Accuracy is important (we don’t care model is simple or complex) |
| Regularization | * In regularization, we combat overfitting by controlling the model's complexity, i.e. by introducing an additional term in our cost function in-order to penalize large weights. This biases our model to be simpler, where simpler is weights of smaller magnitude (or even zero). We want to make the weights smaller because complex models and overfitting are characterized by large weights. |
| Interaction variable | Creating variable by combining two or more variable for ex: high income, low saving.   * Random forest and neural network capture interaction variable so they are better than logistic * Logistic regression does not capture interaction variable and focuses more on independent variable. |
| Lasso & Ridge regression | * Lasso regression is of one degree polynomial * Ridge is of two degree polynomial (L2) * Lasso makes co-efficient zero for features which are not releveant.   They both are used to minimise the cost function |
| RSE | * Square root of (RSS/df). Where df = n-2 * RSE has same disadvantages as RSS, so R^2 is used for analysis |
|  | * The slope of the line does not determine the correlation coefficient. For ex: two lines with slope 1 \* 0.5 has all the data points falling on line may have the coefficient as 1. |
| Scaling. | * Model scaling only changes co-efficient and does not change p-values & accuracy * Standardisation & min-max scaling   **Standardisation** basically brings all of the data into a standard normal distribution with mean zero and standard deviation one.   * Standardisation:  x=x−mean(x) / sd(x)   MinMax scaling, on the other hand, brings all of the data in the range of 0 and 1. The formulae in the background used for each of these methods are as given below:  **Minx-max scaling/normalisation:**   * (normalisation) MinMax Scaling: x= x−min(x) / max(x)−min(x) * Min-max scaling takes care of outlier. * Low AIC is better but high AIC is not * Correlation (pearson) is also called R or pearson |
| Variance from logistic regression | VIFi=1/ (1−Ri2) |
|  | Graphical user interface, application, Teams  Description automatically generated |
|  |  |
| Linear regression/logistic regression. | **Suggested Answer**  The two main important differences between logistic and linear regression are:  1. Dependent/response variable in linear regression is continuous whereas, in logistic regression, it is the discrete type.  2. Cost function in linear regression minimise the error term Sum(Actual(Y)-Predicted(Y))^2 but logistic regression uses maximum likelihood method for maximising probabilities. |
| Multiple linear regression model:  Model assessment | Hence, there are two new parameters that come into picture:   * Adjusted R2=1− [(1−R2)(N−1)/N−p−1] * AIC=n×log(RSS/n)+2p   Here, n is the sample size meaning the number of rows you'd have in the dataset and p is the number of predictor variables. |
| Ridge regression | If the penalty is very large it means model is less complex, therefore the bias would be high.  What will happen when you apply very large penalty?   * In lasso some of the coefficient value become zero, * but in case of Ridge, the coefficients become close to zero but not zero. |
| stochastic and non-stochastic | * Stochastic means for same input multiple output * In linear regression predictor x is non-stochastic and error free. |

|  |  |
| --- | --- |
| **df.size** | **df.size:** Return an int representing the number of elements in this object. Return the number of rows if Series, otherwise returns the number of rows times the number of columns if Data Frame. |
| **Linear regression RSS, TSS, ESS** | RSS = actual-predicted  TSS = actual-mean  ESS explained variation = predicted – mean |
| **R^2 (R square)**  **Residual square error (RSE)** | 1 – (RSS/TSS)  RSE = Square root(RSS/DF) where df = n-2 (where n = number of data points) |
| Linear regression | from sklearn.linear\_model import LinearRegression  lm = LinearRegression()  lm.fit(X\_train\_lm, y\_train\_lm)  print(lm.intercept\_) : this is C in y= mx + c  print(lm.coef\_): this is m in y=mx+C  stats model:  lr.params()/  lr.summary() |
| ROC curve (TPR/FPR) | TPR : Y-axis  FPR : X-axis. [1 – True Negative Rate(Specificity)]  When TPR increases FPR also increases |
| k-fold cross validation | y-axis: r-squared  X-axis: number of features  If the gap Is high it means over fitting for ex: lok at features 10-12  Chart, line chart  Description automatically generated |
|  | **Two-sample proportion test**is used when your sample observations are categorical, with two categories. It could be True/False, 1/0, Yes/No, Male/Female, Success/Failure etc. |

|  |
| --- |
| String.isalpha()  String.index  Isinstance(“value”, str)  Int(“string\_value”), float(“string\_value”) 🡪 convert string to numeric  Str() convert to string  print("{:.2f}".format(a\_float)) |
| Let us summarise all we have done till now:   * vect.fit(train) learns the vocabulary of the training data * vect.transform(train) uses the fitted vocabulary to build a document-term matrix from the training data * vect.transform(test) uses the fitted vocabulary to build a document-term matrix from the testing data (and ignores tokens it hasn't seen before) |
| * import scipy.stats as st * m=float(input())#mean * sd=float(input())#standard deviation * dist = (90 - m)/sd * print(round(1- st.norm.cdf(dist), 2)) * df\_1 = pd.merge(churn\_data, customer\_data, how='inner', on='customerID') * # Reading the input * import ast,sys * input\_str = sys.stdin.read() * input\_list = ast.literal\_eval(input\_str) |
| The **'fit\_transform'** command first fits the data to have a mean of 0 and a standard deviation of 1, i.e. it scales all the variables using:  Xscaled=X−μ/σ  Now, once this is done, all the variables are transformed using this formula. Now, when you go ahead to the test set, you want the variables to not learn anything new. You want to use the old centralisation that you had when you used fit on the train dataset. And this is why you don't apply 'fit' on the test data, just the 'transform'.  You can also refer to [this](https://datascience.stackexchange.com/questions/12321/difference-between-fit-and-fit-transform-in-scikit-learn-models)StackOverflow answer. |
| Some methods that can be used to deal with multicollinearity are:   1. **Dropping variables**    * Drop the variable which is highly correlated with others    * Pick the business interpretable variable 2. **Create new variable** using the interactions of the older variables    * Add interaction features, i.e. features derived using some of the original features 3. **Variable transformations**    * Principal Component Analysis (covered in a later module) |
| * import numpy as np * seed=int(input()) * n=int(input()) * p=float(input()) * np.random.seed(seed) * s = np.random.binomial(n,p,10) * print(s) |