Cheatsheets / **Machine Learning and Algorithms for Data Science Interviews**

Supervised Learning Interview Questions

Logistic Regression Classifier

Logistic Regression is supervised binary classification algorithm used to predict binary response variables that may indicate the presence or absence of some state. It is possible to extend *Logistic Regression* to multi-class classification problems by creating several one-vs-all binary classifiers. In a one-vs-all scheme, n - 1 classes are grouped as one and a classifier learns to discriminate the remaining class from the ensembled group.

Assessing Fit Of A Linear Regression Model: RSE

Residual Standard Error (RSE) provides an absolute measure of lack of fit of a linear regression model to the data. Because it is measured in the units of the outcome variable, it is not always clear what RSE value constitutes a strongly fitted model.

For example, if we create a model that was trying to predict the amount of money earned by sales based on TV advertisements, RSE would be measured in dollars (the units of the outcome variable).

In R, the RSE of a linear regression model can be found by calling the `summary()` function using the model as a parameter. It can also be found by calling the `sigma()` function using the model as a parameter.

```
# RSE can be found in the summary of
a model.
summary(model)

# This will also return the RSE of
a model.
sigma(model)
```

Linear Regression Assumptions

The assumptions of simple linear regression are:

- linear functional form: the relationship between the outcome and predictor variable must be linear (not curved)
- normality: the residuals should be approximately normally distributed
- homoscedasticity: the variance of the residuals should be equal for all values of the predictor

Adjusted R-Squared

If we want to compare nested models, R-squared can be problematic because it will ALWAYS favor the larger (and therefore more complex) model. Adjusted R-squared is an alternative metric that penalizes R-squared for each additional predictor. Therefore, larger nested models will always have larger R-squared but may have smaller adjusted R-squared.

Inspecting Variable Types

One of the most important first steps when working with a dataset is to inspect the variable types, and identify relevant variables. An efficient method to use when inspecting variables is the `.head()` method which will return the first rows of a dataset.

```
print(df.head())
```

Multiple Linear Regression

Multiple linear regression is an extension of simple linear regression used to model the relationship between a quantitative response variable and two or more predictors, which may be quantitative, categorical, or a mix of both. This allows us to control for confounding variables, which may distort the perceived relationship between two variables if not accounted for.

R-Squared

One method for comparing linear regression models is R-squared. R-squared is interpreted as the proportion of variation in an outcome variable which is explained by a particular model. Therefore, we generally prefer models with higher R-squared.

Decision Tree Representation

In a decision tree, leaves represent class labels, internal nodes represent a single feature, and the edges of the tree represent possible values of those features.
Unlike other classifiers, this visual structure gives us great insight about the algorithm performance.

Cross-Entropy Loss

Cross-entropy is a score that summarizes the average difference between the actual and predicted probability distributions for all classes. In a classification model, the goal is to minimize the score, with a perfect cross-entropy value is 0.
We can calculate cross-entropy loss by using the `log_loss()` function in scikit-learn.

```
# example implementation of cross-entropy loss

true_labels = [1, 0, 0]
predicted_labels = [0.7, 0.2, 0.1]
print(log_loss(true_labels, predicted_labels))
```

Logistic Regression interpretability

Logistic Regression models have high interpretability compared to most classification algorithms due to optimized feature coefficients. Feature coefficients can be thought as a measure of sensitivity in feature values.

One-Hot Encoding with Python

When working with nominal categorical variables in Python, it can be useful to use One-Hot Encoding, which is a technique that will effectively create binary variables for each of the nominal categories. This encodes the variable without creating an order among the categories. To one-hot encode a variable in a pandas dataframe, we can use the `.get_dummies()` .

```
df = pd.get_dummies(data = df, columns=['column1', 'column2'])
```