Assignment 8

1. **What are some beneﬁts of feature selection? How do you use F-test to select the features?**

ANS :

**Benefits:**

Three key benefits of using feature selections are :

1. Reduces Overfitting: Less redundant data means less opportunity to make decisions based on noise.
2. Improves Accuracy : Less misleading data means modelling accuracy improves .
3. Reduces Training Time : Less data means that algorithm trains faster.

**How do we use F-test to select the features : For this answer please go through jupyter notebook file which is given in this folder.**

1. **Can we use PCA for feature selection? If yes, then why?**

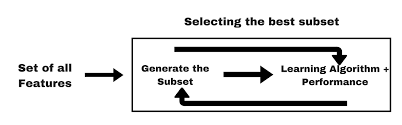
ANS:

Yes, we can use PCA for feature selection. The only way PCA is a valid method of feature selection is if the most important variables are the ones that happen to have the most variation in them. However, this is usually not true. The difference is that PCA will try to reduce dimensionality by exploring how one feature of the data is expressed in terms of the other features (linear dependency). Feature selection instead, takes the target into consideration.

PCA **helps us to identify patterns in data based on the correlation between features**. In a nutshell, PCA aims to find the directions of maximum variance in high-dimensional data and projects it onto a new subspace with equal or fewer dimensions than the original one.

1. **What’s the difference between forwarding Feature Selection and Backward Feature Selection?**

Forward feature selection:



Forward Selection: Forward selection is **an iterative method in which we start with having no feature in the model**. In each iteration, we keep adding the feature which best improves our model till an addition of a new variable does not improve the performance of the model.

Backward elimination:

Backward elimination is **a feature selection technique while building a machine learning model**. It is used to remove those features that do not have a significant effect on the dependent variable or prediction of output.

**Difference:** Forward selection starts with a (usually empty) set of variables and adds variables to it, until some stop- ping criterion is met. Similarly, backward selection starts with a (usually complete) set of variables and then excludes variables from that set, again, until some stopping criterion is met.

1. **How do you transform skewed distribution into a Normal Distribution, name some techniques?**

Ans:

There are some techniques to transform skewed distribution into Normal Distribution.

1. **Log Transform**

Log transformation is **a data transformation method in which it replaces each variable x with a log(x)**. The choice of the logarithm base is usually left up to the analyst and it would depend on the purposes of statistical modeling.

Log transformation is most likely the first thing you should do to remove skewness from the predictor.

It can be easily done via Numpy, just by calling the log() function on the desired column.

**2. Square Root Transform**

**a procedure for converting a set of data in which each value, xi, is replaced by its square root**, another number that when multiplied by itself yields xi. Square-root transformations often result in homogeneity of variance for the different levels of the independent variable (x) under consideration.

The square root sometimes works great and sometimes isn’t the best suitable option. In this case, I still expect the transformed distribution to look somewhat exponential, but just due to taking a square root the range of the variable will be smaller.

You can apply a square root transformation via *Numpy*, by calling the sqrt() function.

1. **Box-Cox Transform**

A Box Cox transformation is **a transformation of non-normal dependent variables into a normal shape**. Normality is an important assumption for many statistical techniques; if your data isn't normal, applying a Box-Cox means that you are able to run a broader number of tests.

You should only know that it is just another way of handling skewed data. To use it, **your data must be positive**— so that can be a bummer sometimes.

You can import it from the *Scipy*library, but the check for the skew you’ll need to convert the resulting Numpy array to a Pandas Series:

1. **Reciprocal Transformation:**

**a transformation of raw data that involves (a) replacing the original data units with their reciprocals and (b) analyzing the modified data**. It can be used with nonzero data and is commonly used when distributions have skewness or clear outliers.

**5 . Exponential Transformation :**

Our basic exponential function is f(x) = b^x, where b is our base, which is a positive constant. All other exponential functions are modifications to this basic form. Transformations are changes to the graph. Transformations include **vertical shifts, horizontal shifts, and graph reversals**.

1. **How to perform Feature Engineering on Unknown features?**

Ans:

There are few important methods to do feature engineering when you don’t have much domain knowledge or you have numerous features and manual feature engineering is the last resort.

1. **Comparison of Manual Feature Engineering with Automated Feature Engineering**

|  |  |
| --- | --- |
| **Manual** | **Automated** |
| Needs domain knowledge. | Minimal domain knowledge is needed. |
| Tedious and Time consuming . | Fast Process |
| Needs skill set to perform . | Tool takes care of feature engineering. |
| Error-Prone | Efficient and Prevent Data leakage. |
| Features are method specific. | Creates interpretable features. |
| Suitable for smaller feature set with domain knowledge. | Used when large no of features , less domain knowledge and limited time frame for modelling. |
|  |  |

**What is Automated Feature Engineering?**

Automated Feature Engineering automatically extracts *useful and meaningful* features from a set of related data tables with a framework that can be applied *to any problem.* It not only cuts down on the time spent feature engineering, but creates meaningful features and prevents data leakage by filtering time-dependent data.

Automated feature engineering is more efficient and repeatable than manual feature engineering allowing you to build better predictive models faster.

**Various methods for Automated Feature Engineering**

Advanced algorithmic methods that analyze existing data to find opportunities for creating new features are :

* Principal component analysis (PCA) and independent component analysis (ICA) map existing data to another feature space
* Deep feature synthesis (DFS) allows for transfer of intermediate learnings from middle layers in the neural networks.