CAPSTONE PROJECT – ADVANCED STATISTICAL MODELLING

**Predicting The Onset of Diabetes – An investigation into the optimal statistical model for predicting the onset of gestational diabetes**

**William Belcher**

**Introduction**

**Data**

The Diabetes data set, originally from the National Institute of Diabetes and Digestive and Kidney Diseases, contains several medical predictors and one target variable [https://www.kaggle.com/uciml/pima-indians-diabetes-database]. The medical predictors include: the number of pregnancies the patient has had, the result of an oral glucose test, diastolic blood pressure, tricep skin fold thickness, their two-hour serum insulin, BMI, the result of a pedigree function and their age. The dataset set is a subset of a larger database, with each observation being taken from a female patient over the age of 21 and of Pima Indian heritage. The following table details all 9 variables

|  |  |
| --- | --- |
| Variable | Type |
| Pregnancies | quantitative, discrete |
| Glucose | quantitative, discrete |
| BloodPressure | quantitative, discrete |
| SkinThickness | quantitative, discrete |
| Insulin | quantitative, discrete |
| BMI | quantitative, continuous |
| DiabetesPedigreeFunction | quantitative, continuous |
| Age | quantitative, discrete |
| Outcome | quantitative, binary |

Table 1: List of variables and their associated types

In total, there were 768 observations. It is worth noting not much information about the DiabetesPedigreeFunction field was supplied. However, for this report it is assumed that this function returns some information based on the family’s history of gestational diabetes.

**Methods**

A range of data science methods were implemented to pre-process the data set before it could be explored further. The software R-Studio was utilised along with its associated language, R [4]. A select few third-party libraries were also installed to enabling the data wrangling process to be made more seamless, namely, **dplyr**, **ggplot2**, **tidyr.**

* Data Representation

The original data set is a ‘comma-separated values’ (.csv) file. By making use of the **read.csv** function and its associated arguments: **header** = TRUE, **sep** = “,”, **dec** = “.”, the data set can be imported into the R workspace.

* Data Cleaning

The data set was first checked for missing entries by a column-wise summation of the number of NA entries. This returned zero for every column indicating that there was no apparent missing data. A summary of the dataset was done using R’s inbuilt summary function. This showed that minimum value for: Glucose, BloodPressure, SkinThickness, Insulin and BMI, was zero. It is unreasonable to have zero for these values so that assumption was made that a zero was used as a placeholder value. As these predictors are unique and cannot be calculated from the remaining variables in the data set, the rows containing these placeholder values were removed leaving 392 observations.

* Exploratory Visualisation using ggplot2:

The basic commands **ggplot**, **geom\_boxplot, geom\_bar, geom\_point, geom\_line, geom\_histogram, geom\_density, geom\_tile** and **geom\_step,** along with the applicable arguments were used to create visualisations of the chosen subsets.

**Results and Discussion**

The original data was a comma separated file with 7,050 observations of 12 variables. After the pre-processing of the data was completed, the remaining data set contained 6,997 rows of 17 variables. Once the initial data had been processed, the initial visual exploration of the data could be completed. An initial probing into the number of each post type was completed, revealing that the majority of sellers are relying on photo and video posts to interact with the followers with these post types combining for almost 88% of posts. Figure 1 illustrates this disparity.

Chart, bar chart

Description automatically generated

Figure 1: A tally count of each status type

A brief look into the mean engagements received on each status type revealed that Videos were able to receive on average over double the amount of engagements as statuses and over quintuple times that of a Photo. Figure 2 displays this fact.

Chart, bar chart

Description automatically generated

Figure 2: Mean engagements per status type

An investigation into the mean number of engagements received on posts per season was conducted next. Figure 3 shows the relationship between the mean engagements and the season, noting the significant increase in reactions during spring as opposed to those during the remaining season.

Chart, bar chart

Description automatically generated

Figure 3: Average engagements received on a post per season

Further breaking down the mean engagements on each post by looking at the weekday each status was posted reveals a decrease in engagements over the mid to late working week followed by a resurgence over the weekend. The data was then grouped by status type to add a further dimension to the plot. Figure 4 illustrates this trend.

Chart, line chart

Description automatically generated

Figure 4: Mean engagements received each day, separate by status type

Investigating the hour at which each status was posted reveals an interesting trend. Figure 5 shows the maximum engagements received on a post per hour it was posted with the mean values overlayed in a line graph. Note the differing scales of the y axis.

Chart, histogram

Description automatically generated

Figure 5: Maximum engagements per hour the status was posted, line indicates mean

A statistical analysis into the correlation between when a Video was posted, and the engagement received was then conducted in order to find what variables impacted the number of engagements the most. As such, a correlation heatmap was constructed and can be found below. A value of one indicates a strong linear correlation with a value of negative one indicating a strong negative correlation, a value of 0 indicates there is no correlation between the two variables.

Chart

Description automatically generated

Figure 6: Correlation Heatmap for the data set

When looking at the engagements row, it can be noted that modifying the status type alters the engagements with a negative correlation of -0.24. The remaining adjustable variables provide no correlation with the number of engagements. The decimal value is due to the correlation being calculated from the factor level of the status type.

These levels are:

|  |  |
| --- | --- |
| Decimal | Level |
| 1 | Video |
| 2 | Status |
| 3 | Link |
| 4 | Photo |

Table 2: Levels of the status type factor

A decision importance tree was then graphed to provide more insight into which variables provided the most impact on the engagements. From the plot, it can be seen that the status type provides the most change in engagements. Followed by season, weekday and then hour.

Chart, bar chart

Description automatically generated

Figure 7: Decision importance tree

A final engagement heatmap can then be constructed to provide a quick and visual way of finding when the best time to post a video is. The resulting heatmap can be seen below.

A screenshot of a cell phone

Description automatically generated

Figure 8: Engagement heatmap

**Conclusion**

When looking at the decision importance tree, as well as figure 2, figure 3, figure 4 and figure 5, it can be seen that posting a video will drastically improve the engagement efficiency of the post. Posting in spring will, on average, bring a 20% increase to the engagements received. Altering the weekday and hour a status is posted provide minimal benefits when compared to the status type and season. If one were so inclined, posting on either a Monday or Tuesday between the hours of 6 to 10am will provide an approximate 45% increase in engagements over a post during any other time period.

In conclusion, the best combination of post type and time posted that will maximise engagements is a Video posted between 6 to 10AM on either a Monday or Tuesday.

**References**

* [4] RStudio Team (2020). RStudio: Integrated Development for R. RStudio, Inc., Boston, MA <http://www.rstudio.com/>

Mode: Desktop. Version 1.25042