To die, or not to die

My experiences from competing at a data science competition

Petri Roine | Senior Advisor

"Our commitment to innovation is why last year analysts named SAS a leader in more than 30 reports and what drives our constant curiosity into what the future holds."

Dr. Jim Goodnight

Den lille Havfrue souvenir.
Original statue by Edvard Eriksen.

Welcome to Kaggle Competitions

Challenge yourself with real-world machine learning problems

New to Data Science?
Get started with a tutorial on our most popular competition for beginners, Titanic: Machine Learning from Disaster.

Build a Model
Get the data & use whatever tools or methods you prefer to make predictions.

Make a Submission
Upload your prediction file for real-time scoring & a spot on the leaderboard.

✖ Dismiss

11 active competitions

Sort By: Prize

Active All Entered

All Categories

Data Science Bowl 2017
Can you improve lung cancer detection?
Featured · 18 days to go
$1,000,000
1,821 teams

The Nature Conservancy Fisheries Monitoring
Can you detect and classify species of fish?
Featured · 13 days to go
$150,000
2,213 teams

Intel & MobileODT Cervical Cancer Screening
Which cancer treatment will be most effective?
Featured · 3 months to go
$100,000
212 teams

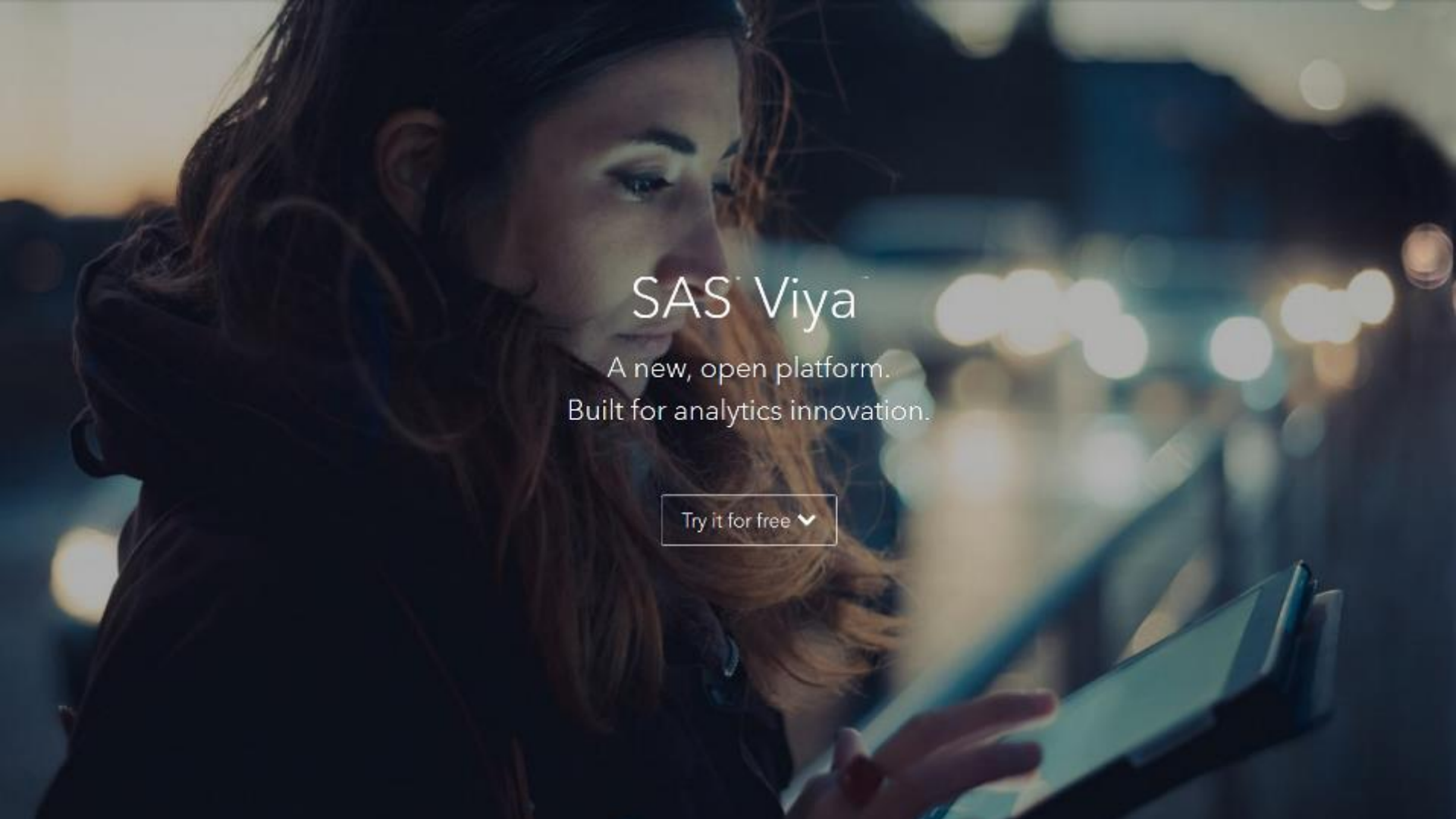
Google Cloud & YouTube-8M Video Understanding Challenge
Can you produce the best video tag predictions?
Featured · 2 months to go
$100,000
370 teams


TITANIC


891
418

{"metadata":{"kernelspec":{"name":"python"},"language_info":{"name":"python","version":"3.5.1"}},"nbformat":4,"nbformat_minor":0,"cells":[{"cell_type":"markdown","metadata":{"_cell_guid":"b8aaab93-e8bc-d65b-a92a-86dee88c7aaa","_active":false,"collapsed":false},"source":"Predictive Analysis of Survival Rate on Titanic\n==========\n\nThis is a predictive machine learning project using ```R``` based on Kaggle competition: [Titanic: Machine Learning from Disaster](https://www.kaggle.com/c/titanic). We first build basic learners using common machine learning model such as Logistic Regression, Random Forest, SVM ect., then use ensemble method to improve the predictive power. \n\n----------\n**Content**\n\n1. Introduction\n\n 1.1 Objective\n\n 1.2 Data Understanding\n\n2. Data Preparation and Exploratory Analysis\n\n 2.1 Data Cleaning\n\n 2.2 Exploratory Analysis and Data Processing\n\n3. Modeling\n\n 3.1 Feature Engineering\n\n 3.2 Model Training\n\n 3.3 Model Evaluation\n\n4. Conclusion\n\n----------
","outputs":[]},{"cell_type":"markdown","metadata":{"_cell_guid":"0b00cc1c-ee80-9adc-b834-d21a8e3696e7","_active":false},"source":"## 1. Introduction\n\n### 1.1 Objective\n\nThe sinking of the RMS Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. This sensational tragedy shocked the international community and led to better safety regulations for ships.\n\nOne of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew. Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class.\n\nIn this challenge, we are going to complete the analysis of what sorts of people were likely to survive.\n\n### 1.2 Data Understanding\n\nTo download data, click [here][1]\n\n\n [1]: https://www.kaggle.com/c/titanic/data\n\nThe data has been split into two groups:\n\n* training set (train.csv)\n* test set (test.csv)\n\nThe training set is used to build machine learning models. For the training set, we provide the outcome (also known as the “ground truth”) for each passenger. Your model will be based on “features” like passengers’ gender and class. You can also use feature engineering to create new features.\n\nThe test set should be used to see how well your model performs on unseen data. For the test set, we do not provide the ground truth for each passenger. It is your job to predict these outcomes. For each passenger in the test set, use the model you trained to predict whether or not they survived the sinking of the Titanic.\n\n**Data Dictionary**\n\n| Variable | Definition | Key |\n|-----------|------------|---------|\n| survival | Survival | 0 = No, 1 = Yes |\n| pclass\t | Ticket class | 1 = 1st, 2 = 2nd, 3 = 3rd |\n| sex | Sex | |\n| Age | Age in years | |\n| sibsp | # of siblings / spouses aboard the Titanic | |\n| parch | # of parents / children aboard the Titanic | |\n| ticket | Ticket number | |\n| fare | Passenger fare | |\n| cabin | Cabin number | |\n| embarked | Port of Embarkation |C = Cherbourg, Q = Queenstown, S = Southampton | \n\n**Variable Notes**\n\npclass: A proxy for socio-economic status (SES)\n1st = Upper\n2nd = Middle\n3rd = Lower\n\nage: Age is fractional if less than 1. If the age is estimated, is it in the form of xx.5\n\nsibsp: The dataset defines family relations in this way...\nSibling = brother, sister, stepbrother, stepsister\nSpouse = husband, wife (mistresses and fiancés were ignored)\n\nparch: The dataset defines family relations in this way...\nParent = mother, father\nChild = daughter, son, stepdaughter, stepson\nSome children travelled only with a nanny, therefore parch=0 for them.\n\nFirst, let's load the data and take a look at it.","outputs":[]},{"cell_type":"code","execution_count":null,"metadata":{"_cell_guid":"69b351c0-653e-457d-37ec-cad7583b25bd","_active":false},"outputs":[],"source":"library('dplyr') # data manipulation\nlibrary('ggplot2') # Data Visualization\nlibrary('ggthemes') # Data Visualization\n\noptions(warn = -1)\n# load train.csv\ntrain <- read.csv('../input/train.csv', stringsAsFactors = F)\n# load test.csv\ntest <- read.csv('../input/test.csv', stringsAsFactors = F)\n# combine them as a whole\ntest$Survived <- NA\nfull <- rbind(train,test)","execution_state":"busy"},{"cell_type":"code","execution_count":null,"metadata":{"_cell_guid":"508bc8d8-9936-a894-27e3-e791b0b73f2b","_active":false},"outputs":[],"source":"# show first several rows of the data\nhead(full)","execution_state":"busy"},{"cell_type":"code","execution_count":null,"metadata":{"_cell_guid":"95b81b55-4cdc-efba-f035-bbed5470d495","_active":false},"outputs":[],"source":"\n# check the data\nstr(full)","execution_state":"busy"},{"cell_type":"markdown","metadata":{"_cell_guid":"fff0d340-9540-1d72-c194-85beced3abbe","_active":false},"source":"We've got a sense of our variables, their class type, and the first few observations of each. We know we're working with 1309 observations of 12 variables. In which 891 observations are from train data set, and 418 observations are from test data set. When separate the variables by type, we have ordinal variable **PassengerId**, lable variable **Name** and **Ticket**, numeric variables such as **Age**, **SibSp**, **Parch**, **Fare**, and categorical variables like **Survived** ,**Pclass**, **Sex** ,**Cabin**, and **Embarked**. \n\n## 2. Data Preparation and Exploratory Analysis\n\n### 2.1 Data Cleaning\n\nFrom the data set, we notice that there are a lot of missing values in **Age** and **Cabin** column. We are going to replace missing values in Age with a random sample from existing ages. For Cabin, since cabin number makes little sense to the result, we are going to create a new Cabin column to indicate how many cabins the passenger has. \n ","outputs":[]},{"cell_type":"code","execution_count":null,"metadata":{"_cell_guid":"22c3765a-9e39-33ca-2202-3c73a1861b3b","_active":false},"outputs":[],"source":"# Process Age Column\n\n # create a new data set age\nage <- full$Age\nn = length(age)\n # replace missing value with a random sample from raw data\nset.seed(123)\nfor(i in 1:n){\n if(is.na(age[i])){\n age[i] = sample(na.omit(full$Age),1)\n }\n}\n # check effect\npar(mfrow=c(1,2))\nhist(full$Age, freq=F, main='Before Replacement', \n col='lightblue', ylim=c(0,0.04),xlab = \"age\")\nhist(age, freq=F, main='After Replacement', \n col='darkblue', ylim=c(0,0.04))","execution_state":"busy"},{"cell_type":"markdown","metadata":{"_cell_guid":"4a5afddf-dde3-3279-e06d-8e9046ef8426","_active":false},"source":"We can see from the histograms above that there is not much significant change of age distribution, which means the replacement is appropriate. Next we are going to process Cabin Column. We are going to create a new Cabin column to indicate how many cabins the passenger has. \n ","outputs":[
{"cell_type":"code","execution_count":null,"metadata":{"_cell_guid":"591a1c4a-cc6","_active":false},"outputs":[],"source":"# Process Cabin Column to show how many cabins the passenger has\ncabin <- full$Cabin\nn = length(cabin)\nfor (i in 1:n){\n if(nchar(cabin[i]) == 0){\n cabin[i] = \"N\"\n }else{\n cabin[i] = substr(cabin[i],1,1)\n }\n}\n# check effect\npar(mfrow=c(1,2))\nbarplot(table(cabin), main='Before Replacement', \n col='lightblue', ylim=c(0,0.04))\nbarplot(table(cabin), main='After Replacement', \n col='darkblue', ylim=c(0,0.04))","execution_state":"busy"},{"cell_type":"markdown","metadata":{"_cell_guid":"4-679d-37f3-40e","_active":false},"source":"### 2.1 Exploratory A
s our ob figure ou s wo ng to the data to e relationship ival.\n\n**A
Survival" ","execu ill,"met _guid 9","_a },"outputs":[number of different name(Age = Survived =
train$Survive ,aes(Age n(Survive hist ecution_count : null ,"meta l_guid":"ff97 4-e011-ch _active": :"From th ems that ery young age
have a respec survival lder peopl ec survival rate. To veri ate a bar chart to ationship bet rate ar
intervals.","cell_t execution_co ell_guid":"591a1c4a-cc6 75dd7502e079","_act reate b how relationshi vival r \ncut
cut(d$Age,his lot = FY e <- tapply(d$ 2 <- data.frame(age = nam e)\nbarplot(d2$rat e\")","outputs":[]},{"ce rkdown" id":" 84-593e-cb41-
887fd20df38e" n see clearly fr ove that **survival rate d age increases**. Ki **Survival**","outp rvival .5, people who's en 10 t ons al rate
around 0.4, a old has a lower sur d 0.2.\n\n\n----------\n\n\n n_count data":{"cell_gui -f3c4-d
6ccc6a9035d9" ,"source":"# create ha w effect of Sex on survival\n , aes(Sex,fill = fa xecution_state":' l_type" ,"me 3a086536-
3610-39cf-120 e", _active":false},"source":"We can se istogram above that **female's te is greater than outputs":[]},{ "code", unt":null,"meta _guid": 31a-1
adf9241a6b5b" alse},"outputs":[],"source":"# calculate te\ntapply(train$Survived,trai ,"execution_state": ell_type":"mark ata":{"_ f26e3f6e-930c-r e7daafab ':false ival rate of
female is 0.7 survival rate of male is 0.19.\n\n\n--- e Name** v.s. **Survival**\n\nW ts":[]},{ "code","exec ull,"met ell_guid 79-ae32-
8c2921aefad8" alse},"outputs":[],"source":"# extract t me\nn = length(full$Survived)\ (NA,n)\nfor (i in 1 ame = strsplit i],\" ,\" name,\". \ }\n\n# ma title v.s
survival\nd < (title = title[1:891],Survived = train$S plot(d, aes(title,fill = factc) +\n geom_histogr ount\")","exec :"busy"},{ {"_cell_gui -1cad-846b
493a71af6dc5" alse},"source":"To see clearly the survi each group, we also make a ta ts":[]},{"cell_type ecution_count ata":{"_cell_ ':false},"out ce":"# count
of title\ntab xecution_state":"busy"},{"cell_type":"c ion_count":null,"metadata":{"_ 860a7969-50ce-93ac 124bf","_acti utputs":[],"sour
rate\ntapply(d$Survived,d$title,mean)","execution_state":"busy"},{"cell_type":"markdown","metadata":{"_cell_guid":"b3ef380-901b-6512-6c10-483224e9a06","_active":false},"source":"We can see from the table, survival rates of females with Miss and Mrs title are close to the average survival rate for female group calculated in last section. Survival rates of males with Master are higher than the average male group. Titles like Col, Rev, Dr etc. also have influence on the survival. \n\nSince number of each rare title is much smaller than the majorities, we replace these rare titles to 'Rare'.","outputs":[]},{"cell_type":"code","execution_count":null,"metadata":{"_cell_guid":"5b498b31-6795-faaa-99d5-9a8644a1549b","_active":false},"outputs":[],"source":"# replace rare titles to 'Rare'\nitle[title != 'Mr' & title != 'Miss' & title != 'Mrs' & title != 'Master'] <- 'Rare'\ntable(title)","execution_state":"busy"},{"cell_type":"markdown","metadata":{"_cell_guid":"83691531-8517-173b-6e1d-450e62d540f0","_active":false},"source":"\n----------\n\n**Pclass** v.s. **Survival**","outputs":[]},{"cell_type":"code","execution_count":null,"metadata":{"_cell_guid":"58704fae-ff99-309f-768f-d78fa4038324","_active":false},"outputs":[],"source":"# make a histogram\nggplot(train, aes(Pclass,fill = factor(Survived))) +\n geom_histogram(stat = \"count\")","execution_state":"busy"},{"cell_type":"code","execution_count":null,"metadata":{"_cell_guid":"b97f386f-0e8d-4204-f0ee-a25f736a53b1","_active":false},"outputs":[],"source":"# calculate survival rate\ntapply(train$Survived,train$Pclass,mean)","execution_state":"busy"},{"cell_type":"markdown","metadata":{"_cell_guid":"261ddce1-86bd-3482-87e2-b3aec8519ce3","_active":false},"source":"From the histogram and table, we notice that **Pclass = 1 group has the highest survival rate, then is Pclass = 2 group, and Pclass = 3 group has the lowest survival rate within these three groups**.\n\n\n----------\n\n**Family Size** v.s. **Survival**\n\nWe first check SibSp and Parch column separately.","outputs":[]},{"cell_type":"code","execution_count":null,"metadata":{"_cell_guid":"080312a2-ca22-15aa-e3de-d91f883cffed","_active":false},"outputs":[],"source":"# histogram of Parch\nggplot(train, aes(Parch,fill = factor(Survived))) +\n geom_histogram(stat = \"count\")","execution_state":"busy"},{"cell_type":"code","execution_count":null,"metadata":{"_cell_guid":"5114d917-316f-97fe-888b-924cfe2aba48","_active":false},"outputs":[],"source":"# histogram of SibSp\nggplot(train, aes(SibSp,fill = factor(Survived))) +\n geom_histogram(stat = \"count\")","execution_state":"busy"},{"cell_type":"markdown","metadata":{"_cell_guid":"02cbf12a-5cb7-8391-0806-35e784247661","_active":false},"source":"We can see that they have similar trend, then we decide to combine them together to construct a column named family.","outputs":[]},{"cell_type":"code","execution_count":null,"metadata":{"_cell_guid":"3b7bae2f-6385-51f1-9f3a-adf200e74fd3","_active":false},"outputs":[],"source":"# combine SibSp and Parch \nfamily <- full$SibSp + full$Parch\nd <- data.frame(family = family[1:891],Survived = train$Survived)\nggplot(d, aes(family,fill = factor(Survived))) +\n geom_histogram(stat = \"count\")","execution_state":"busy"},{"cell_type":"code","execution_count":null,"metadata":{"_cell_guid":"963c35a2-b4fb-3af9-aa2a-380759205dd5","_active":false},"outputs":[],"source":"tapply(d$Survived,d$family,mean)","execution_state":"busy"},{"cell_type":"markdown","metadata":{"_cell_guid":"c36dc79a-16ab-b11f-0f1f-4ce1a3403c67","_active":false},"source":"We can see that **the survival rate increases as the family size increases from 0 to 3. When family size becomes greater than 3, survival rate decrease dramatically.** \n\n\n----------\n\n**Cabin** v.s. **Survival**","outputs":[]},{"cell_type":"code","execution_count":null,"metadata":{"_cell_guid":"e52425c5-db2a-7fa1-de1d-bba8071020d9","_active":false},"outputs":[],"source":"# create histogram\nd <- data.frame(Cabin = cabin[1:891],Survived = train$Survived)\nggplot(d, aes(Cabin,fill = factor(Survived))) +\n geom_histogram(stat = \"count\")","execution_state":"busy"},{"cell_type":"code","execution_count":null,"metadata":{"_cell_guid":"d70ffb4e-bc1f-546f-e584-11703328b350","_active":false},"outputs":[],"source":"# calculate survival rate\ntapply(d$Survived,d$Cabin,mean)","execution_state":"busy"},{"cell_type":"markdown","metadata":{"_cell_guid":"f16f5d70-0a8b-892a-bf64-803ad5e9c4b2","_active":false},"source":"We notice that **passenger who has no cabin has a lower survival rate, and passenger who has one or more cabins has higher survival rate.**\n\n\n----------\n\n**Fare** v.s. **Survival**","outputs":[]},{"cell_type":"code","execution_count":null,"metadata":{"_cell_guid":"8461cbe9-da59-ca36-cb15-e2ff51580ffd","_active":false},"outputs":[],"source":"# make a histogram\nggplot(train, aes(Fare,fill = factor(Survived))) +\n geom_histogram()","execution_state":"busy"},{"cell_type":"code","execution_count":null,"metadata":{"_cell_guid":"d0998202-b220-288a-23d9-4353caaf3739","_active":false},"outputs":[],"source":"# calculate\ncuts <- cut(train$Fare,hist(train$Fare,50,plot = F)$breaks)\nrate <- tapply(train$Survived,cuts,mean)\nd <- data.frame(fare = names(rate),rate)\nbarplot(d$rate, xlab = \"fare\",ylab = \"survival rate\")","execution_state":"busy"},{"cell_type":"markdown","metadata":{"_cell_guid":"98f1cb7b-ffbf-b706-e5a2-04b3a5279794","_active":false},"source":"We notice that **there is no significant relationship between Fare and survival rate.**\n\n\n----------\n\n**Embarked** v.s. **Survival**","outputs":[]},{"cell_type":"code","execution_count":null,"metadata":{"_cell_guid":"485a95c8-a765-f2ee-7d3b-d3046414fbd8","_active":false},"outputs":[],"source":"# make a histogram\nggplot(train, aes(Embarked,fill = factor(Survived))) +\n geom_histogram(stat = \"count\")","execution_state":"busy"},{"cell_type":"code","execution_count":null,"metadata":{"_cell_guid":"0b135060-3233-af3b-8d38-52a0c7e0f574","_active":false},"outputs":[],"source":"# make table\ntapply(train$Survived,train$Embarked,mean)","execution_state":"busy"},{"cell_type":"markdown","metadata":{"_cell_guid":"bcff166b-d4b7-40a6-ca54-6679e053525d","_active":false},"source":"We notice that Embarked C group has a relatively higher survival rate than other 2 groups.\n\n## 3. Modeling\n\n### 3.1 Feature Engineering","outputs":[]}]}

```
---title: 'Exploring the Titanic Dataset'author: 'Megan L. Risdal'date: '6 March 2016'output:  html_document:    number_sections: true    toc: true    fig_width: 7    fig_height: 4.5    theme: readable    highlight: tango---# IntroductionThis is my first
stab at a Kaggle script. I have chosen to work with the Titanic dataset after spending some time poking around on the site and looking at other scripts made by other Kagglers for inspiration. I will also focus on doing some illustrative data visualizations
along the way. I'll then use `randomForest` to create a model predicting survival on the Titanic. I am new to machine learning and hoping to learn a lot, so feedback is very welcome! There are three parts to my script as follows:* Feature engineering*
Missing value imputation* Prediction!## Load and check data```{r, message = FALSE}# Load packageslibrary('ggplot2') # visualizationlibrary('ggthemes') # visualizationlibrary('scales') # visualizationlibrary('dplyr') # data manipulationlibrary('mice') #
imputationlibrary('randomForest') # classification algorithm```Now that our packages are loaded, let's read in and take a peek at the data.```{r, message=FALSE, warning=FALSE}train <- read.csv('../input/train.csv', stringsAsFactors = F)test  <-
read.csv('../input/test.csv', stringsAsFactors = F)full  <- bind_rows(train, test) # bind training & test data# check datastr(full)```We've got a sense of our variables, their class type, and the first few observations of each. We know we're working with
1309 observations of 12 variables. To make things a bit more explicit since a couple of the variable names aren't 100% illuminating, here's what we've got to deal with:Variable Name | Description--------------|-------------Survived      | Survived (1) or
died (0)Pclass        | Passenger's className          | Passenger's nameSex           | Passenger's sexAge           | Passenger's ageSibSp         | Number of siblings/spouses aboardParch         | Number of parents/children aboardTicket        | Ticket
numberFare          | FareCabin         | CabinEmbarked      | Port of embarkation# Feature Engineering## What's in a name?The first variable which catches my attention is **passenger name** because we can break it down into additional meaningful variables
which can feed predictions or be used in the creation of additional new variables. For instance, **passenger title** is contained within the passenger name variable and we can use **surname** to represent families. Let's do some **feature
engineering**!```{r, message=FALSE, warning=FALSE}# Grab title from passenger namesfull$Title <- gsub('(.*, )|(\\..*)', '', full$Name)# Show title counts by sextable(full$Sex, full$Title)# Titles with very low cell counts to be combined to "rare"
levelrare_title <- c('Dona', 'Lady', 'the Countess','Capt', 'Col', 'Don',                'Dr', 'Major', 'Rev', 'Sir', 'Jonkheer')# Also reassign mlle, ms, and mme accordinglyfull$Title[full$Title == 'Mlle']        <- 'Miss' full$Title[full$Title == 'Ms']
<- 'Miss'full$Title[full$Title == 'Mme']         <- 'Mrs' full$Title[full$Title %in% rare_title]  <- 'Rare Title'# Show title counts by sex againtable(full$Sex, full$Title)# Finally, grab surname from passenger namefull$Surname <- sapply(full$Name,
function(x) strsplit(x, split = '[,.]')[[1]][1])``````{r results='asis'}cat(paste('We have <b>', nlevels(factor(full$Surname)), '</b> unique surnames. I would be interested to infer ethnicity based on surname --- another time.'))```## Do families sink or
swim together?Now that we've taken care of splitting passenger name into some new variables, w          variables. First we're going to make a **family size** variable based on number of siblings/spouse(s) (maybe
someone has more than one spouse?) and number of children/parents. ```{r}# Create a family siz          ll$SibSp + full$Parch + 1# Create a family variable full$Family <- paste(full$Surname, full$Fsize,
sep='_')```What does our family size variable look like? To help us understand how it may rela          sage=FALSE, warning=FALSE}# Use ggplot2 to visualize the relationship between family size &
survivalggplot(full[1:891,], aes(x = Fsize, fill = factor(Survived))) +  geom_bar(stat='count'          x = 'Family Size') +  theme_few()```Ah hah. We can see that there's a survival penalty to
singletons and those with family sizes above 4. We can collapse this variable into three level          milies. Let's create a **discretized family size** variable.```{r}# Discretize family
sizefull$FsizeD[full$Fsize == 1] <- 'singleton'full$FsizeD[full$Fsize < 5 & full$Fsize > 1] <-          urvival using a mosaic plotmosaicplot(table(full$FsizeD, full$Survived), main='Family Size by
Survival', shade=TRUE)```The mosaic plot shows that we preserve our rule that there's a surviva          sengers in small families. I want to do something further with our age variable, but `r
sum(is.na(full$Age))` rows have missing age values, so we will have to wait until after we add          t a few more varia          probably some potentially useful information in the **passenger cabin** variable including
about their **deck**. Let's take a look.```{r}# This variable appears to have a lot of missing          The first character          strsplit(full$Cabin[2], NULL)[[1]]# Create a Deck variable. Get passenger deck A -
F:full$Deck<-factor(sapply(full$Cabin, function(x) strsplit(x, NULL)[[1]][1]))```There's more          e here including looki          ple rooms listed (e.g., row 28: "C23 C25 C27"), but given the sparseness of the column
we'll stop here.# MissingnessNow we're ready to start exploring missing data and rectifying it          re are a number of dif          bout doing this. Given the small size of the dataset, we probably should not opt for
deleting either entire observations (rows) or variables (columns) containing missing values. W          of either replacing          sible values given the distribution of the data, e.g., the mean, median or mode. Finally,
we could go with prediction. We'll use both of the two latter methods and I'll rely on some da          e our decisions.## Se          `{r}# Passengers 62 and 830 are missing Embarkmentfull[c(62, 830), 'Embarked']``````{r
results='asis'}cat(paste('We will infer their values for **embarkment** based on present data          pe relevant: **pass          . We see that they paid<b> $', full[c(62, 830), 'Fare'][[1]][1], '</b>and<b> $', full[c(62,
830), 'Fare'][[1]][2], '</b>respectively and their classes are<b>', full[c(62, 830), 'Pclass']          o from where did they embark?'))``````{r, message=FALSE, warning=FALSE}# Get rid of our
missing passenger IDsembark_fare <- full %>%  filter(PassengerId != 62 & PassengerId != 830)#          ggplot(embark_fare, aes(x = Embarked, y = Fare, fill = factor(Pclass))) +  geom_boxplot() +
geom_hline(aes(yintercept=80),     colour='red', linetype='dashed', lwd=2) +  scale_y_continuo          e for a first class passenger departing from Charbourg ('C') coincides nicely with the $80 paid
by our embarkment-deficient passengers. I think we can safely replace the NA values with 'C'.`          embarked from 'C'full$Embarked[c(62, 830)] <- 'C'```We're close to fixing the handful of NA values
here and there. Passenger on row 1044 has an NA Fare value.```{r, message=FALSE, warning=FALSE          er who departed from Southampton ('S'). Let's visualize Fares among all others sharing their class and
embarkment (n = `r nrow(full[full$Pclass == '3' & full$Embarked == 'S', ]) - 1`).```{r, messag          full$Embarked == 'S', ],   aes(x = Fare)) +  geom_density(fill = '#99d6ff', alpha=0.4) +
geom_vline(aes(xintercept=median(Fare, na.rm=T)),    colour='red', linetype='dashed', lwd=1) +          few()```From this visualization, it seems quite reasonable to replace the NA Fare value with median for
their class and embarkment which is $`r  median(full[full$Pclass == '3' & full$Embarked == 'S'          .```{r}          with median fare for class/embarkmentfull$Fare[1044] <- median(full[full$Pclass == '3' & full$Embarked
== 'S', ]$Fare, na.rm = TRUE)```## Predictive imputationFinally, as we noted earlier, there ar          Age** valu          g to get a bit more fancy in imputing missing age values. Why? Because we can. We will create a model
predicting ages based on other variables.```{r}# Show number of missing Age valuessum(is.na(fu          nitely use          ing for regression) to predict missing ages, but I'm going to use the `mice` package for this task
just for something different. You can read more about multiple imputation using chained equati          ww.jstatsoft.          5i03.pdf) (PDF). Since we haven't done it yet, I'll first factorize the factor variables and then
perform mice imputation.```{r, message=FALSE, warning=FALSE}# Make variables factors into fact          sengerId','Pc          'Title','Surname','Family','FsizeD')full[factor_vars] <- lapply(full[factor_vars],
function(x) as.factor(x))# Set a random seedset.seed(129)# Perform mice imputation, excluding          variables:mice_          full) %in% c('PassengerId','Name','Ticket','Cabin','Family','Surname','Survived')], method='rf')
# Save the complete output mice_output <- complete(mice_mod)```Let's compare the results we ge          ibution of pass          thing has gone completely awry.```{r}# Plot age distributionspar(mfrow=c(1,2))hist(full$Age,
freq=F, main='Age: Original Data',   col='darkgreen', ylim=c(0,0.04))hist(mice_output$Age, fre          ut',   col='ligh          hings look good, so let's replace our age vector in the original data with the output from
the `mice` model.```{r}# Replace Age variable from the mice model.full$Age <- mice_output$Age#          ing Age valuessum(          ished imputing values for all variables that we care about for now! Now that we have a
complete Age variable, there are just a few finishing touches I'd like to make. We can use Age          eature engineering          ound 2Now that we know everyone's age, we can create a couple of new age-dependent
variables: **Child** and **Mother**. A child will simply be someone under 18 years of age and          who is 1) female, 2)          n 0 children (no kidding!), and 4) does not have the title 'Miss'.```{r, message=FALSE,
warning=FALSE}# First we'll look at the relationship between age & survivalggplot(full[1:891,]          (Survived))) +   geom_          Sex since we know (a priori) it's a significant predictor  facet_grid(.~Sex) +
theme_few()# Create the column child, and indicate whether child or adultfull$Child[full$Age <          d[full$Age >= 18] <- 'A          l$Child, full$Survived)```Looks like being a child doesn't hurt, but it's not going to
necessarily save you either! We will finish off our feature engineering by creating the **Moth          can hope that mothers ar          ved on the Titanic.```{r}# Adding Mother variablefull$Mother <- 'Not
Mother'full$Mother[full$Sex == 'female' & full$Parch > 0 & full$Age > 18 & full$Title != 'Miss          tstable(full$Mother, ful          rizing our two new factor variablesfull$Child  <- factor(full$Child)full$Mother <-
factor(full$Mother)```All of the variables we care about should be taken care of and there should be no missing data. I'm going to double check just to be sure:```{r}md.pattern(full)```Wow! We have finally finished treating all of the relevant missing
values in the Titanic dataset which has included some fancy imputation with `mice`. We have also successfully created several new variables which we hope will help us build a model which reliably predicts survival. # PredictionAt last we're ready to
predict who survives among passengers of the Titanic based on variables that we carefully curated and treated for missing values. For this, we will rely on the `randomForest` classification algorithm; we spent all that time on imputation, after all.##
Split into training & test setsOur first step is to split the data back into the original test and training sets.```{r}# Split the data back into a train set and a test settrain <- full[1:891,]test <- full[892:1309,]```## Building the model We then build
our model using `randomForest` on the training set.```{r}# Set a random seedset.seed(754)# Build the model (note: not all possible variables are used)rf_model <- randomForest(factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch +
Fare + Embarked + Title +                                         FsizeD + Child + Mother,                                         data = train)# Show model errorplot(rf_model, ylim=c(0,0.36))legend('topright', colnames(rf_model$err.rate), col=1:3,
fill=1:3)```The black line shows the overall error rate which falls below 20%. The red and green lines show the error rate for 'died' and 'survived' respectively. We can see that right now we're much more successful predicting death than we are survival.
What does that say about me, I wonder?## Variable importanceLet's look at relative variable importance by plotting the mean decrease in Gini calculated across all trees.```{r, message=FALSE, warning=FALSE}# Get importanceimportance    <-
importance(rf_model)varImportance <- data.frame(Variables = row.names(importance),                            Importance = round(importance[ ,'MeanDecreaseGini'],2))# Create a rank variable based on importancerankImportance <- varImportance %>%
mutate(Rank = paste0('#',dense_rank(desc(Importance))))# Use ggplot2 to visualize the relative importance of variablesggplot(rankImportance, aes(x = reorder(Variables, Importance),     y = Importance, fill = Importance)) +  geom_bar(stat='identity') +
geom_text(aes(x = Variables, y = 0.5, label = Rank),    hjust=0, vjust=0.55, size = 4, colour = 'red') +  labs(x = 'Variables') +  coord_flip() +   theme_few()```Whoa, glad we made our title variable! It has the highest relative importance out of all of
our predictor variables. I think I'm most surprised to see that passenger class fell to `r rankImportance[rankImportance$Variable == 'Pclass', ]$Rank`, but maybe that's just bias coming from watching the movie Titanic too many times as a kid.##
Prediction!We're ready for the final step --- making our prediction! When we finish here, we could iterate through the preceding steps making tweaks as we go or fit the data using different models or use different combinations of variables to achieve
better predictions. But this is a good starting (and stopping) point for me now.```{r}# Predict using the test setprediction <- predict(rf_model, test)# Save the solution to a dataframe with two columns: PassengerID and Survived (prediction)solution <-
data.frame(PassengerID = test$PassengerId, Survived = prediction)# Write the solution to filewrite.csv(solution, file = 'rf_mod_Solution.csv', row.names = F)```# ConclusionThank you for taking the time to read through my first exploration of a Kaggle
dataset. I look forward to doing more. Again, this newbie welcomes comments and suggestions!---
```


SAS Viya
A new, open platform.
Built for analytics innovation.
Try it for free

SAS® Visual Data Mining and Machine Learning

Boost analytical productivity and solve complex problems faster with a single, integrated in-memory environment that's both open and scalable.

SAS® Visual Investigator

Address all your intelligence analysis and investigation management needs with a single, cloud-based solution that uses advanced analytics and machine learning technology.

SAS® Visual Analytics

Visually explore relationships and patterns in data smartly, quickly and easily – using interactive data discovery. And illuminate critical insights with self-service analytics.

SAS® Visual Statistics

Delve deeper into different types of data with a powerful, in-memory solution that lets you create, compare and refine descriptive and predictive models on the fly.

SAS® Visual Forecasting

Manage organizational planning challenges and automate large-scale hierarchical forecasting with a solution that supports open source and SAS coding in a single environment.

SAS® Optimization

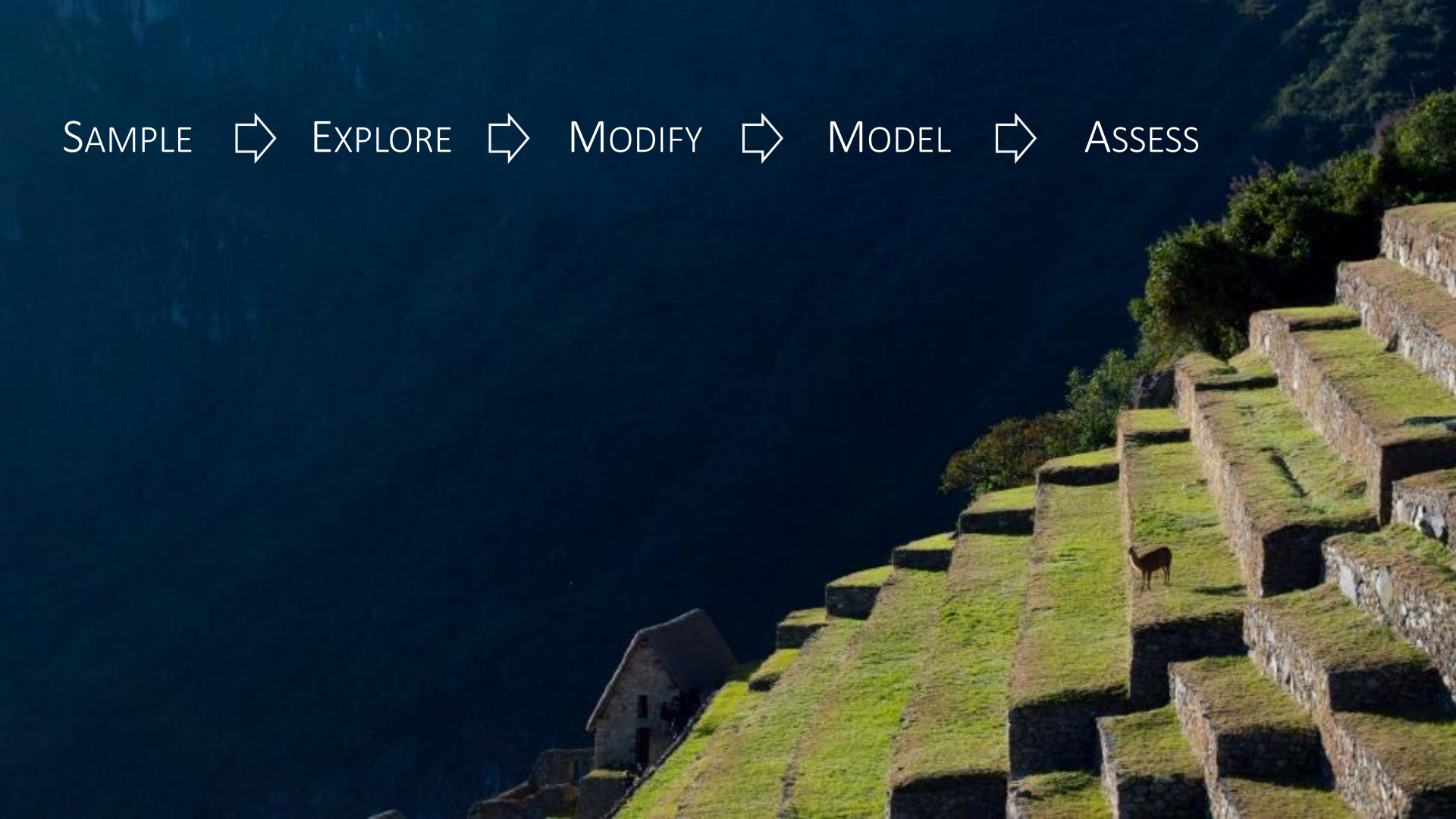
Evaluate alternative actions and scenarios with a powerful array of optimization modeling capabilities and solution techniques.

SAS® Econometrics

Analyze, describe and predict issues related to economic and financial systems by applying advanced mathematical and statistical methods.

SAS® Event Stream Processing

Analyze high-velocity big data in event streams. Train models, score data and take immediate action on what's relevant.


SAMPLE ⇨ EXPLORE ⇨ MODIFY ⇨ MODEL ⇨ ASSESS

SAMPLE

Sampling

DATA DICTIONARY		
Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	Male, female
age	Age in years	0 - 80
sibsp	# of siblings / spouses aboard	0 - 8
parch	# of parents / children aboard	0 - 6
ticket	Ticket number	Unique string
fare	Passenger fare	0 - 512
cabin	Cabin number	C85, E46,...
embarked	Port of Embarkation	C = Cherbourg Q = Queenstown S = Southampton

TRAIN

Server: cas-shared-default

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05
6	0	3	Moran, Mr. James	male	.	0	0	330877	8.4583
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625
8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.075
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	11.1333
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	30.0708
11	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	PP 9549	16.7
12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.55
13	0	3	Saundercock, Mr. William Henry	male	20	0	0	A/5. 2151	8.05
14	0	3	Andersson, Mr. Anders Johan	male	39	1	5	347082	31.275
15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14	0	0	350406	7.8542
16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55	0	0	248706	16
17	0	3	Rice, Master. Eugene	male	2	4	1	382652	29.125
18	1	2	Williams, Mr. Charles Eugene	male	.	0	0	244373	13
19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)	female	31	1	0	345763	18
20	1	3	Masselmani, Mrs. Fatima	female	.	0	0	2649	7.225
21	0	2	Fynney, Mr. Joseph J	male	35	0	0	239865	26
22	1	2	Beesley, Mr. Lawrence	male	34	0	0	248698	13
23	1	3	McGowan, Miss. Anna "Annie"	female	15	0	0	330923	8.0292
24	1	1	Sloper, Mr. William Thompson	male	28	0	0	113788	35.5
25	0	3	Palsson, Miss. Torborg Danira	female	8	3	1	349909	21.075
26	1	3	Asplund, Mrs. Carl Oscar (Selma Augusta Emilia Johansson)	female	38	1	5	347077	31.3875
27	0	3	Emir, Mr. Farred Chehab	male	.	0	0	2631	7.225
28	0	1	Fortune, Mr. Charles Alexander	male	19	3	2	19950	263
29	1	3	O'Dwyer, Miss. Ellen "Nellie"	female	.	0	0	330959	7.8792
30	0	3	Todoroff, Mr. Lalio	male	.	0	0	349216	7.8958


Training set
Validation set


Training set

Fare	Cabin	Embarked	Validate
7.25		S	1
71.2833	C85	C	0
7.925		S	1
53.1	C123	S	0
			1
			0
			1
			0
			1
			0
16.7	G6	S	1
26.55	C103	S	0
8.05		S	1
31.275		S	0

$$f(x) = Train/Validate$$

$$f(x) = Floor\left(PassangerId - x * Floor\left(\frac{PassagerId}{x}\right)\right)$$

$$f(2) \approx 50/50$$
$$f(1.4) \approx 70/30$$
$$f(1.25) \approx 80/20$$

Plan

Add calculated column

Input table:

TRAIN

Output table:

TRAIN

expression:

\"PassengerId\"n - (2 * Floor ((\...

Output column:

Validate

inColumns:

["PassengerId","(2 * Floor((Passe...

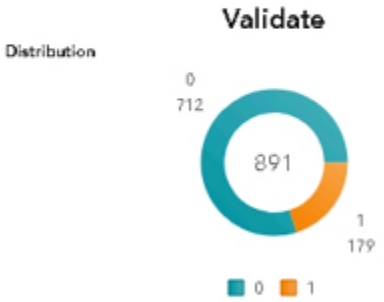
Column format:

TRAIN_NEW
Server: cas-shared-default

Table Profile | Column Profile

Rename column: Validate
Change data type: Double
Change column format: COMMA5.

Unique Groups | Column Metrics



Standard Metrics | Advanced Metrics

Unique (n):	2	Missing:	0
Minimum:	0	Maximum:	1
Mean:	0.20	Sum:	179.00

SibSp	Parch	Ticket	Fare	Cabin	Embarked	Validate
1	0	A/5 21171	7.25		S	1
1	0	PC 17599	71.2833	C85	C	0
0	0	STON/O2. 3101282	7.925		S	0
1	0	113803	53.1	C123	S	0
0	0	373450	8.05		S	0

DATA EXPLORATION

Visualization

Variable Identification

Missing Values Imputation

Outliers Treatment

SAS® Visual Analytics - *Kaggle Titanic

Search

SAS Demo User

Page 1 | Page 2 | +

Data

TRAIN_NEW

Add

Category

- Age - 89
- Cabin - 148
- Embarked - 4
- Embarked_new - 3
- Name - 891
- Parch - 7
- PassengerId - 891
- Pclass - 3
- Sex - 2
- SibSp - 7
- Survived - 2
- Ticket - 681
- Validate - 2

Measure

- Fare
- Frequency

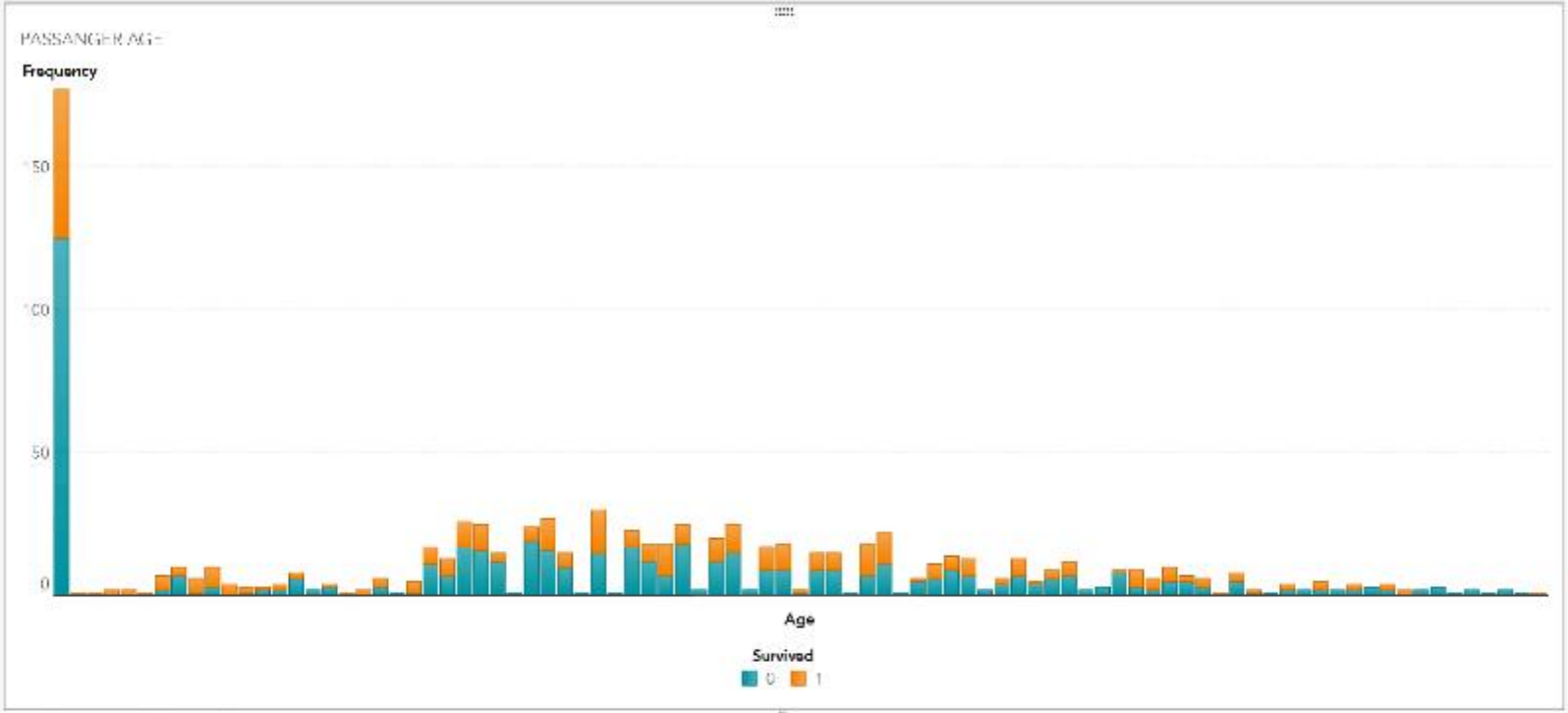
Aggregated Measure

- Frequency Percent

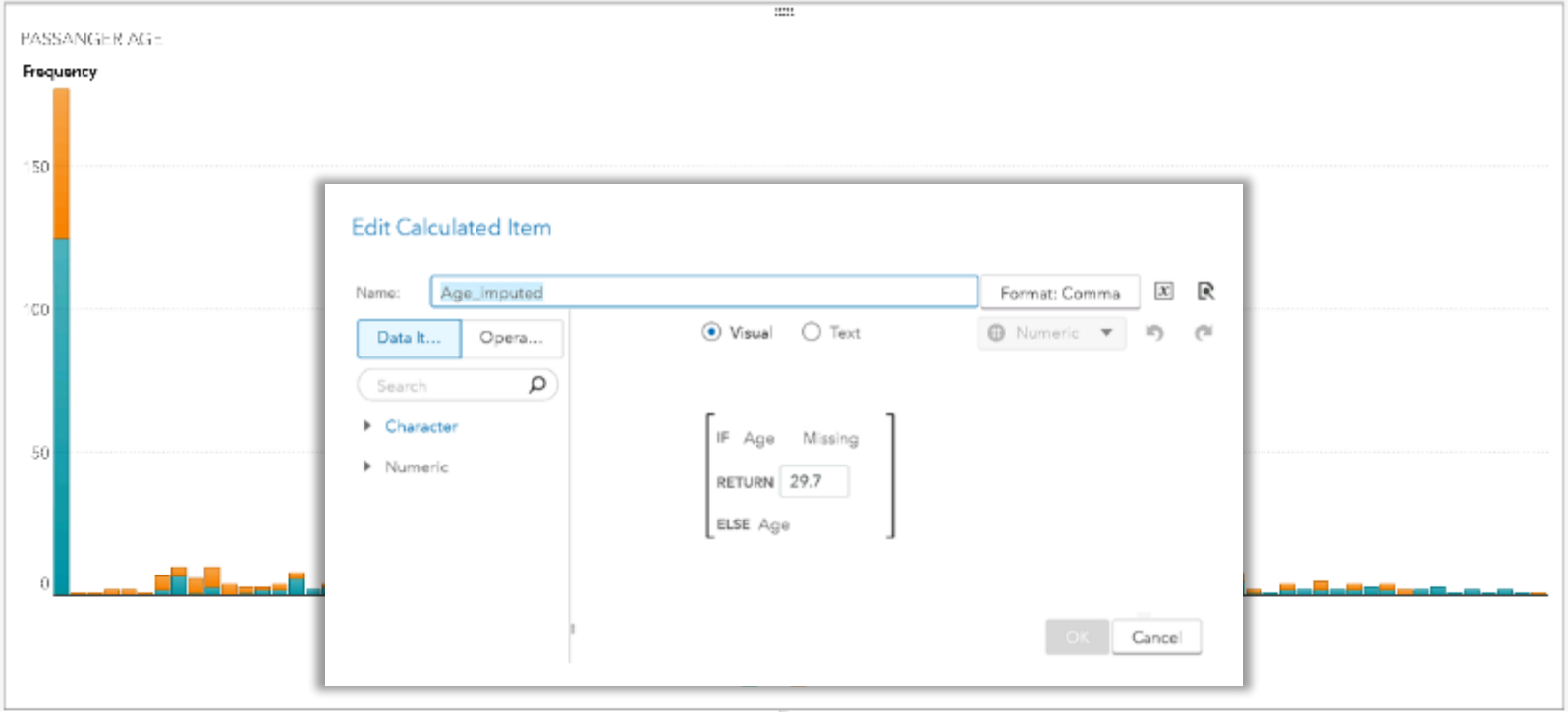
Drop a data item or content here

Roles

Select an object to see its roles

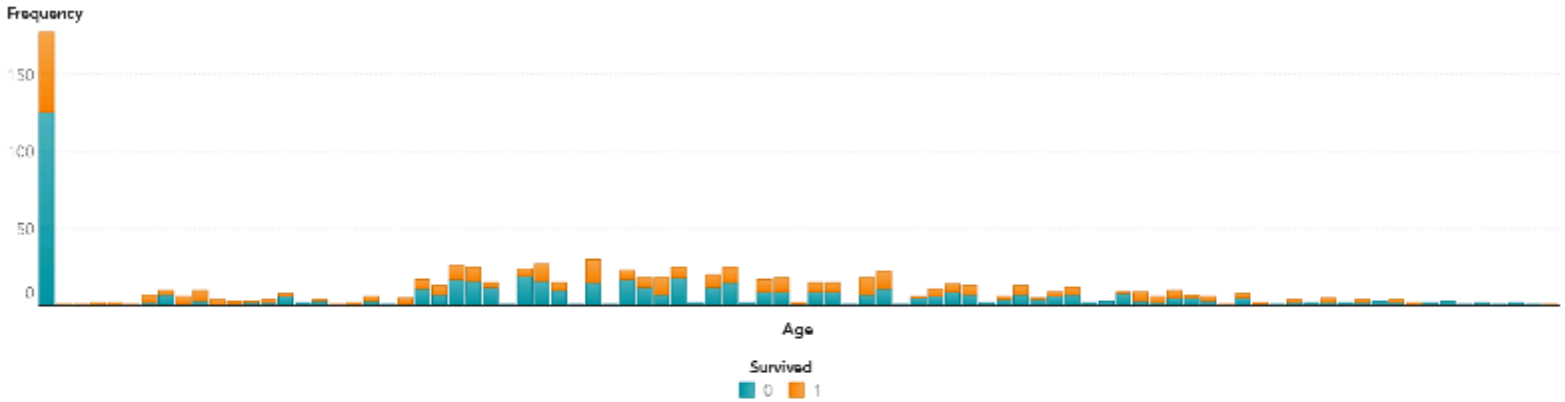


Age	Frequency	Survived
.	125	0
.	52	1
0.42	1	1
0.67	1	1
0.75	2	1
0.83	2	1
0.92	1	1
1.00	2	0

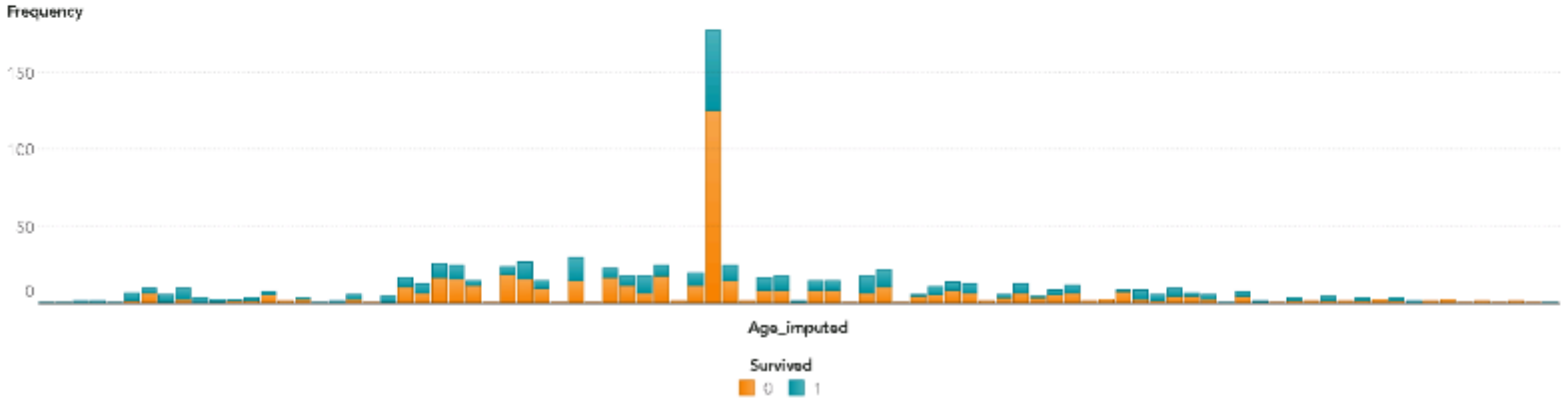


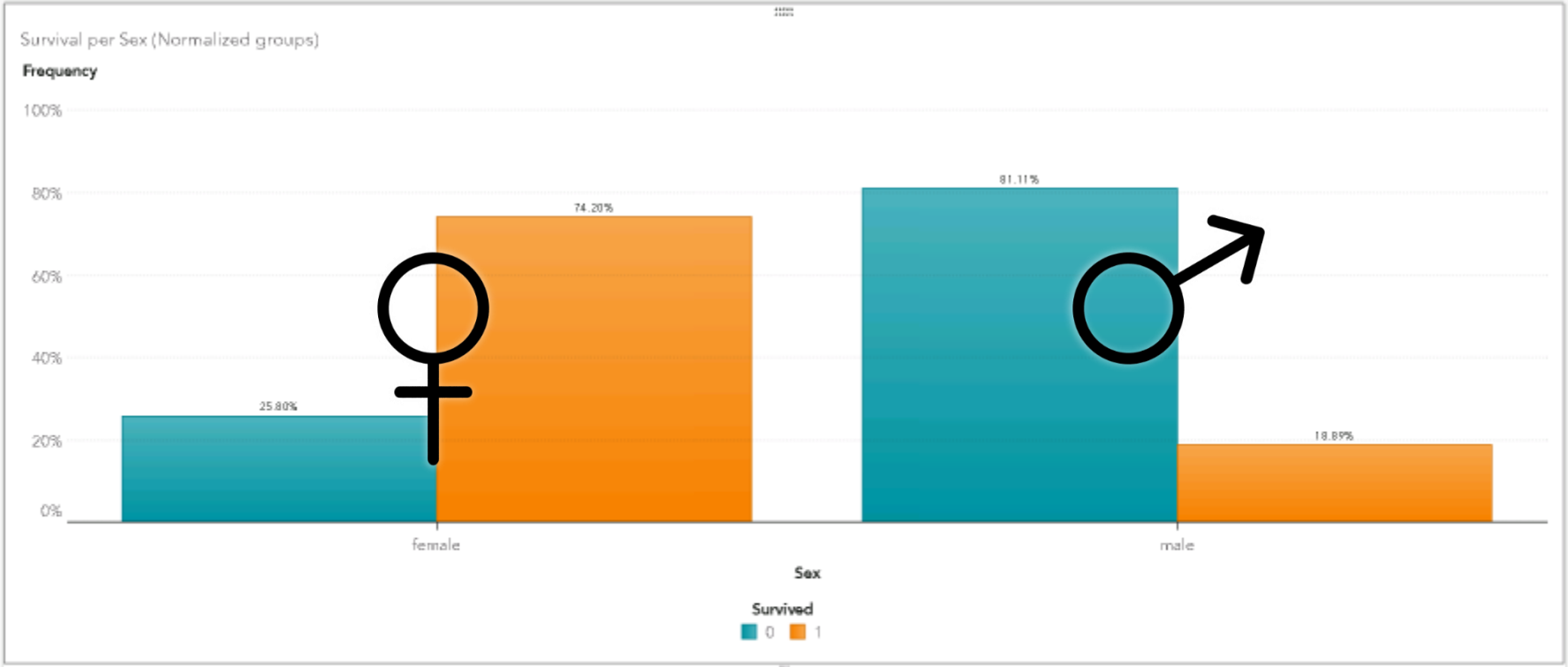
Age	Frequency	Survived
.	125	0
.	52	1
0.42	1	1
0.67	1	1
0.75	2	1
0.83	2	1
0.92	1	1
1.00	2	0

PASSANGER AGE



PASSANGER AGE (imputed)





Sex	Frequency	Survived
fem...	81	0
fem...	233	1
male	468	0
male	109	1

FEATURE ENGINEERING

Variable transformation

Variable / Feature creation

SAS® Visual Analytics - *Kaggle Titanic

Search

SAS Demo User

Page 1 | Page 2 | +

Data

TRAIN_NEW

Add

Category

- Age - 89
- Cabin - 148
- Embarked - 4
- Embarked_new - 3
- Name - 891
- Parch - 7
- PassengerId - 891
- Pclass - 3
- Sex - 2
- SibSp - 7
- Survived - 2
- Ticket - 681
- Validate - 2

Measure

- Fare
- Frequency

Aggregated Measure

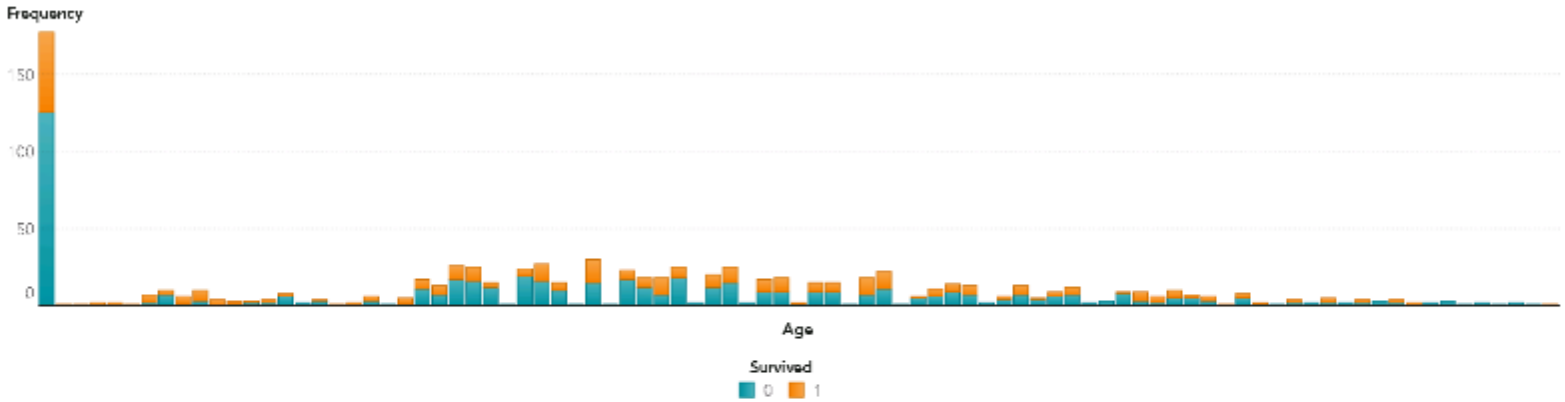
- Frequency Percent

Drop a data item or content here

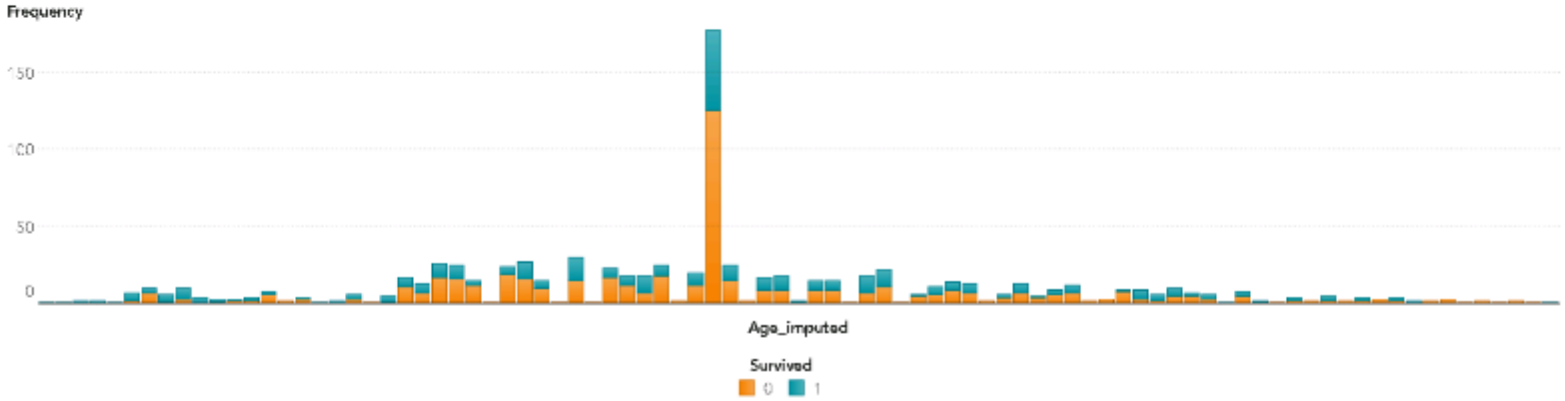
Roles

Select an object to see its roles

PASSANGER AGE

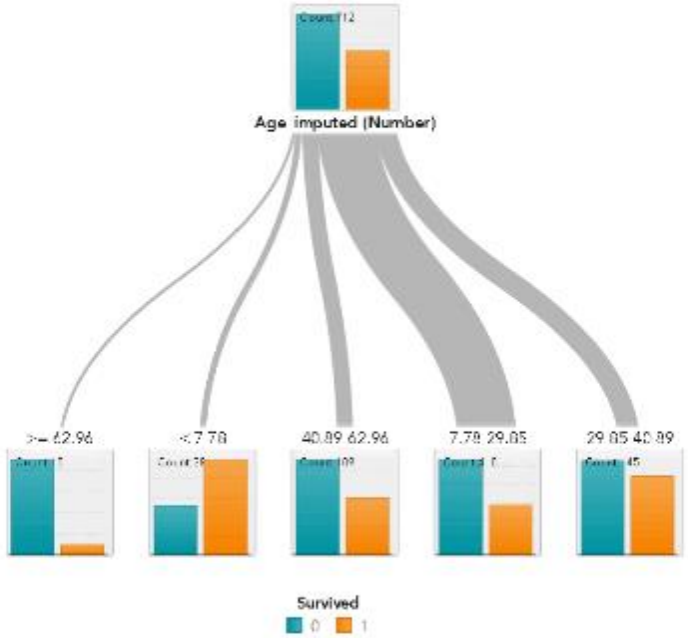


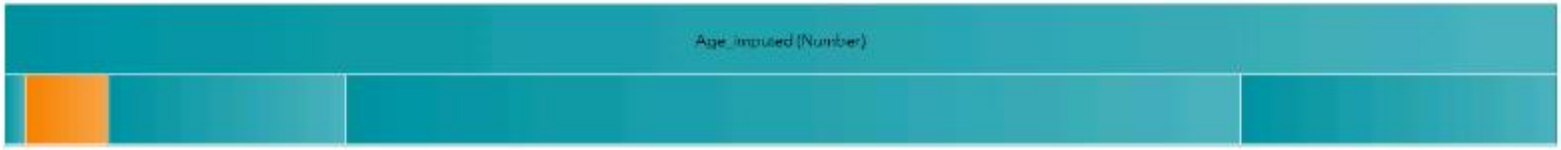
PASSANGER AGE (imputed)



Decision Tree Survived Observations used 712

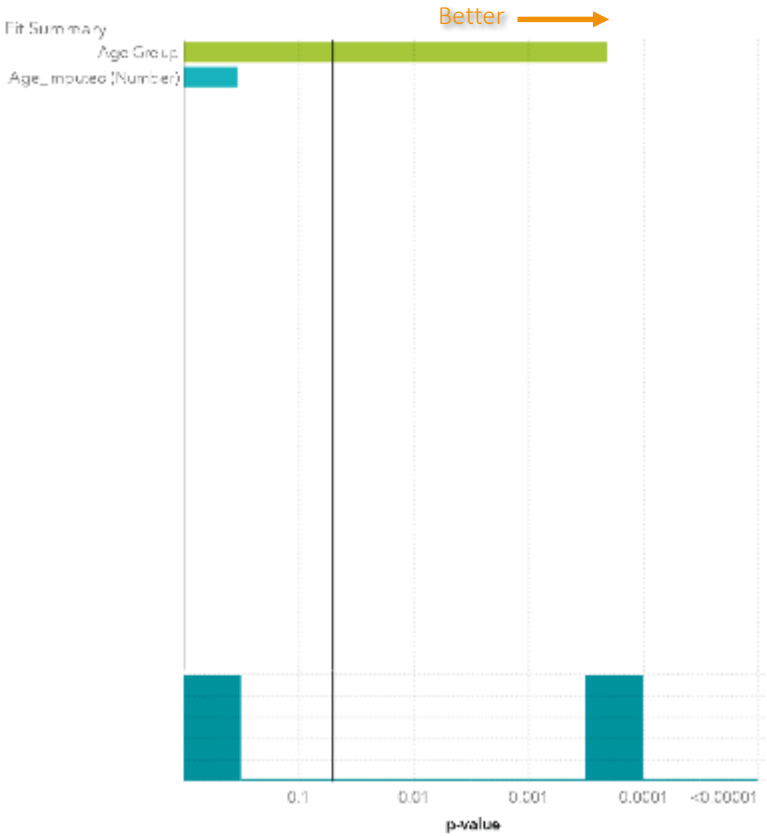
Tree



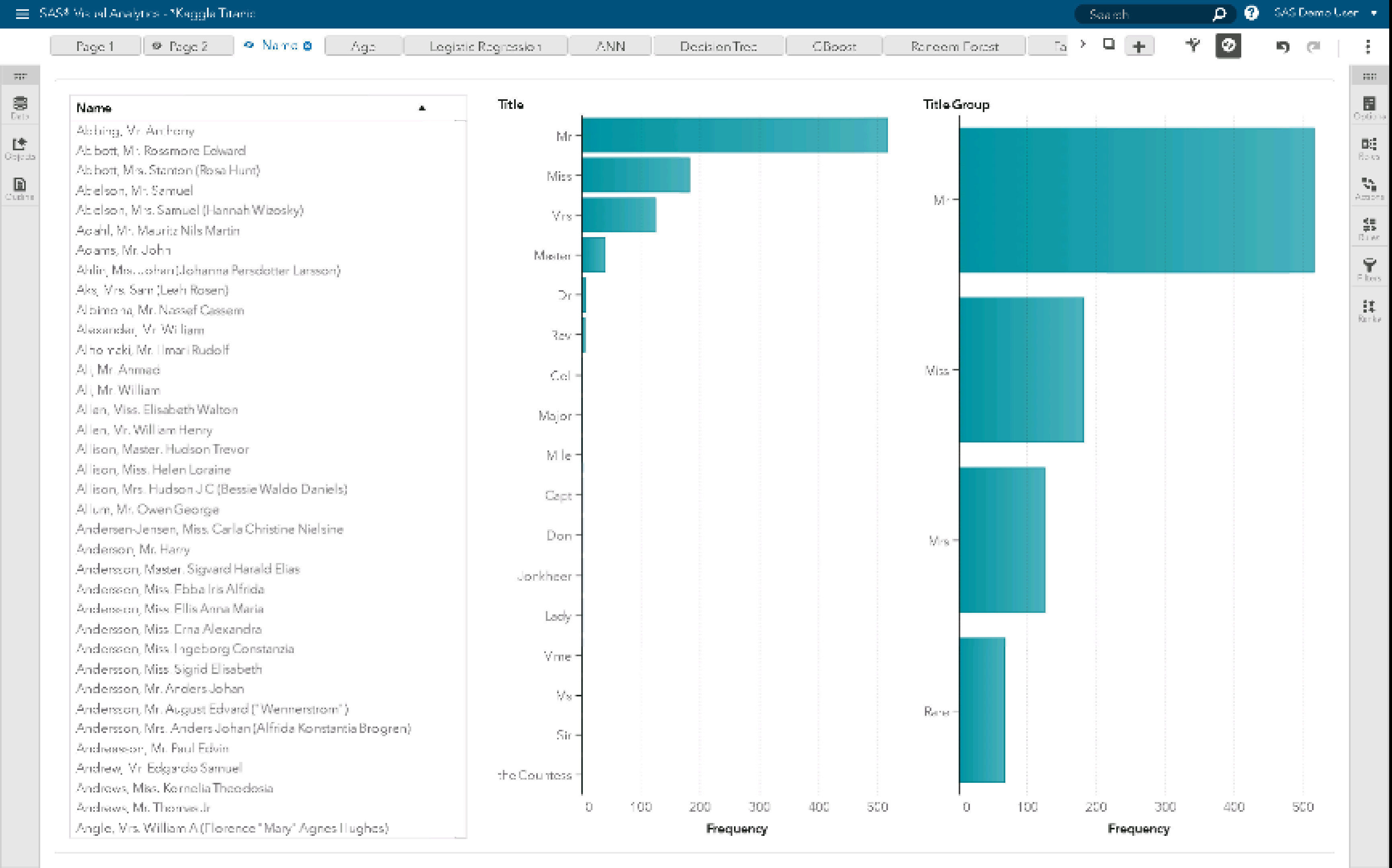


Raw vs Binning

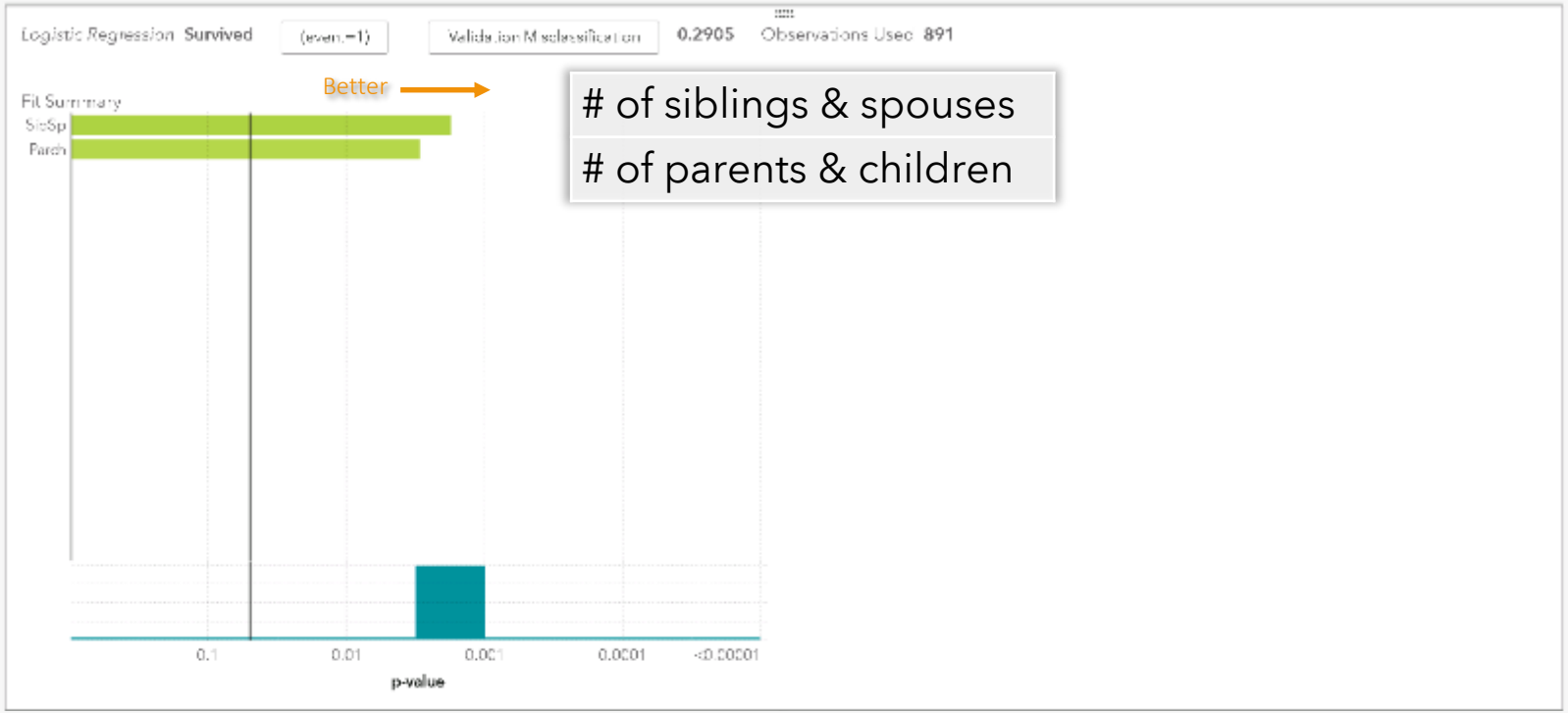
Logistic Regression **Survived** (event=1) R-Square **0.0302** Observations Used **891**



Name	Ticket	Cabin
Braund, Mr. Owen Harris	A/5 21171	
Cumings, Mrs. John Bradley (Florence Briggs Thayer)	PC 17599	C85
Heikkinen, Miss. Laina	STON/O2. 3101282	
Futrelle, Mrs. Jacques Heath (Lily May Peel)	113803	C123
Allen, Mr. William Henry	373450	
Moran, Mr. James	330877	
McCarthy, Mr. Timothy J	17463	E46
Palsson, Master. Gosta Leonard	349909	
Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	347742	
Nasser, Mrs. Nicholas (Adele Achem)	237736	
Sandstrom, Miss. Marguerite Rut	PP 9549	G6
Bonnell, Miss. Elizabeth	113783	C103
Saundercock, Mr. William Henry	A/5. 2151	
Andersson, Mr. Anders Johan	347082	
Vestrom, Miss. Hulda Amanda Adolfina	350406	
Hewlett, Mrs. (Mary D Kingcome)	248706	
Rice, Master. Eugene	382652	



Page 1
Page 2
Name
Age
Logistic Regression
ANN
Decision Tree
Random Forest
Name
Abbott, Mr. Rossmore Edward
Abbott, Mrs. Stanton (Rosa Hunt)
Abelson, Mr. Samuel
Abelson, Mrs. Samuel (Hannah Wizosky)
Adams, Mr. John
Allen, Miss. Elisabeth Walton
Allen, Mr. William Henry
Allison, Master. Hudson Trevor
Allison, Miss. Helen Loraine
Allison, Mrs. Hudson J C (Bessie Waldo Daniels)
Allum, Mr. Owen George
Andersen-Jensen, Miss. Carla Christine Nielsine
Andersson, Master. Sigvard Harald Elias
Andersson, Miss. Ebba Iris Alfrida
Andersson, Miss. Ellis Anna Maria
Andersson, Miss. Erna Alexandra
Andersson, Miss. Ingeborg Constanzia
Andersson, Miss. Sigrid Elisabeth
Andersson, Mr. Anders Johan
Andersson, Mr. August Edvard ("Wennerstrom")
Andersson, Mrs. Anders Johan (Alfrida Konstantia Brogren)
Andrews, Miss. Kornelia Theodosia
Title
Mr
Miss
Mrs
Master
Dr
Rev
Col
Major
Mlle
Capt
Don
Jonkheer
Lady
Mme
Ms
Sir
the Countess
0
100
200
300
400
500
Frequency
Title Group
Mr
Miss
Mrs
Rare
0
100
200
300
400
500
Frequency


Logistic Regression Survived
(event=1)
Validation Misclassification
0.2905
Observations Used 891
Better
Fit Summary
SibSp
Parch
0.1
0.01
0.001
0.0001
<0.00001
p-value
# of siblings & spouses
# of parents & children

Edit Calculated Item

Name: Family

Format: Comma

Data It... | Opera...

Search

- Character
- Numeric
  - _comma
  - _period
  - Age
  - Age (Num...
  - Age_imo...
  - Age_imo...
  - Fare
  - Leaf ID (A...

Visual | Text

Numeric

(Parch + SibSp)

OK | Cancel

Logistic Regression Survived (event=1) Validation Misclassification

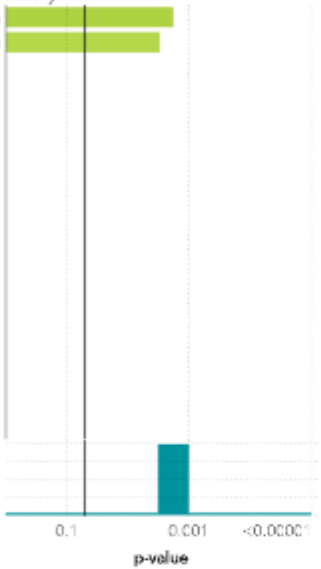
0.2905 Observations Used 891

Better

Fit Summary

SibSp

Parch



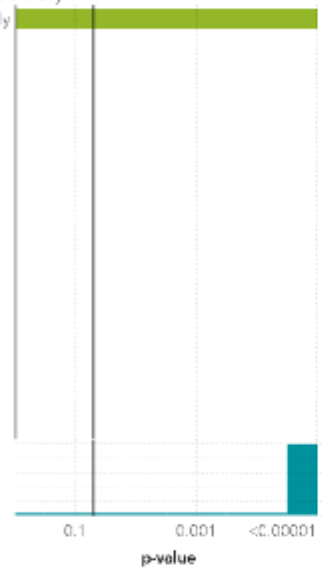
Logistic Regression Survived (event=1) Validation Misclassification

0.2961 Observations Used 891

Better

Fit Summary

Family



MODEL

Logistic Regression

Decision tree

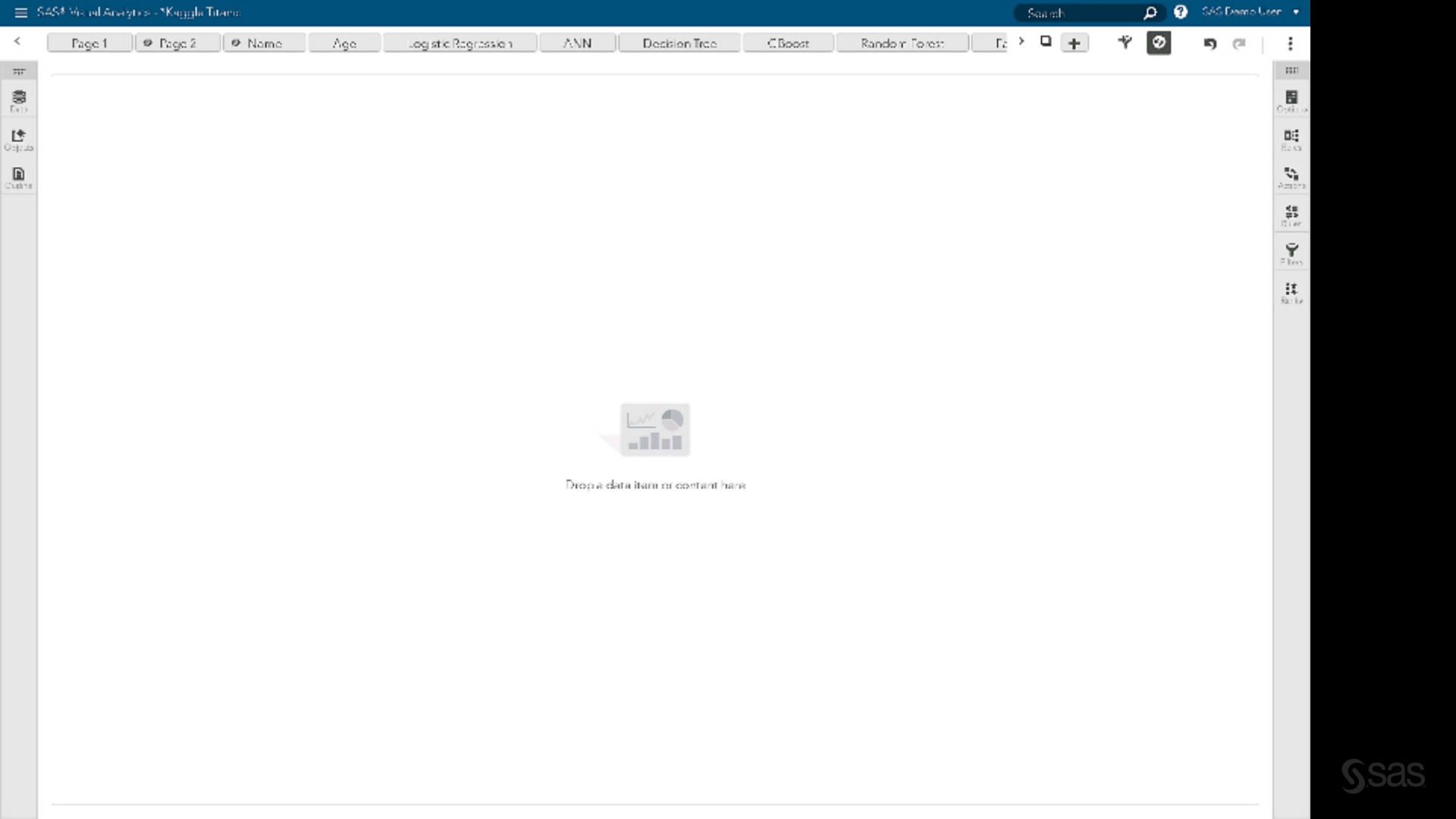
Random Forest

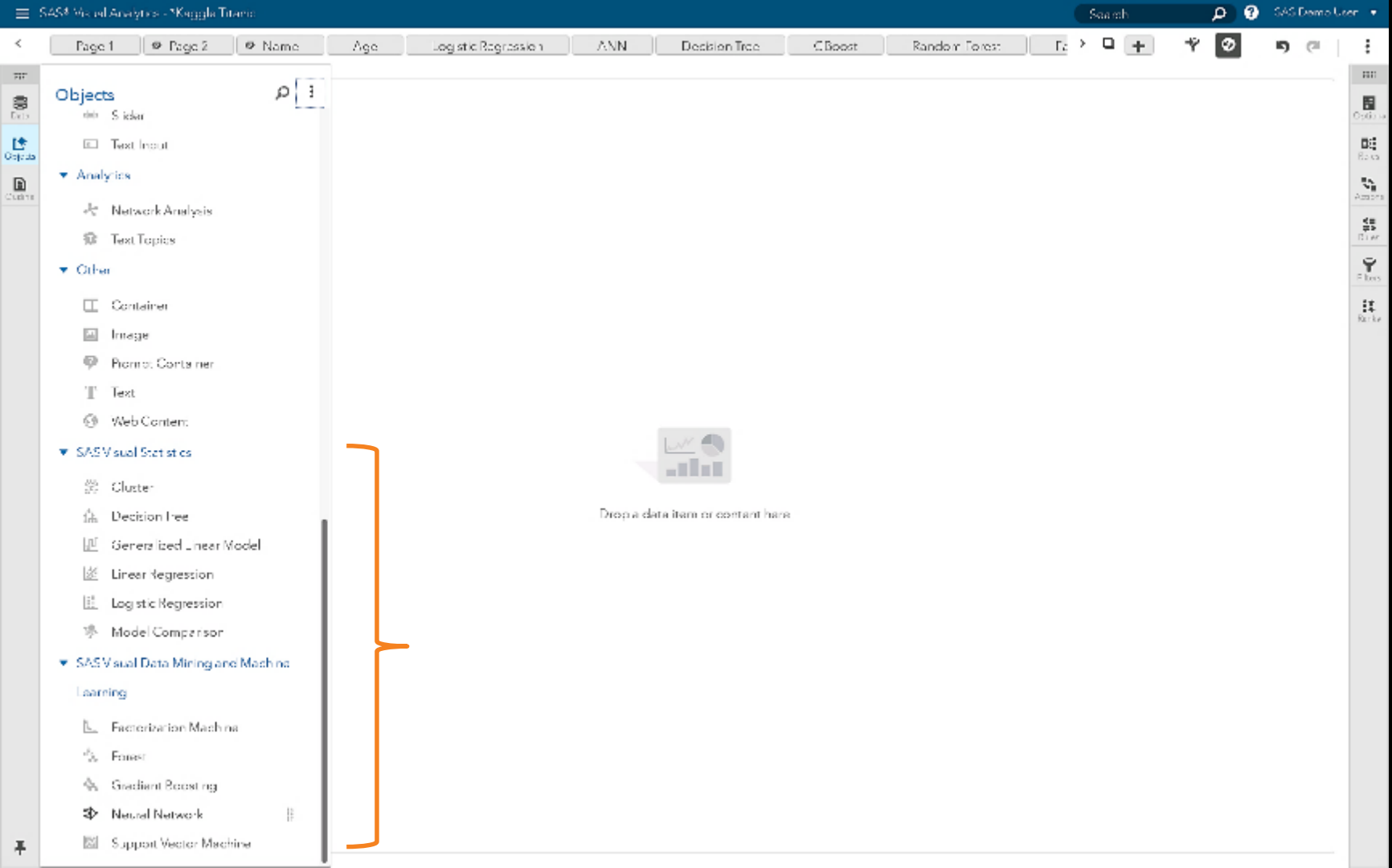
Neural Network

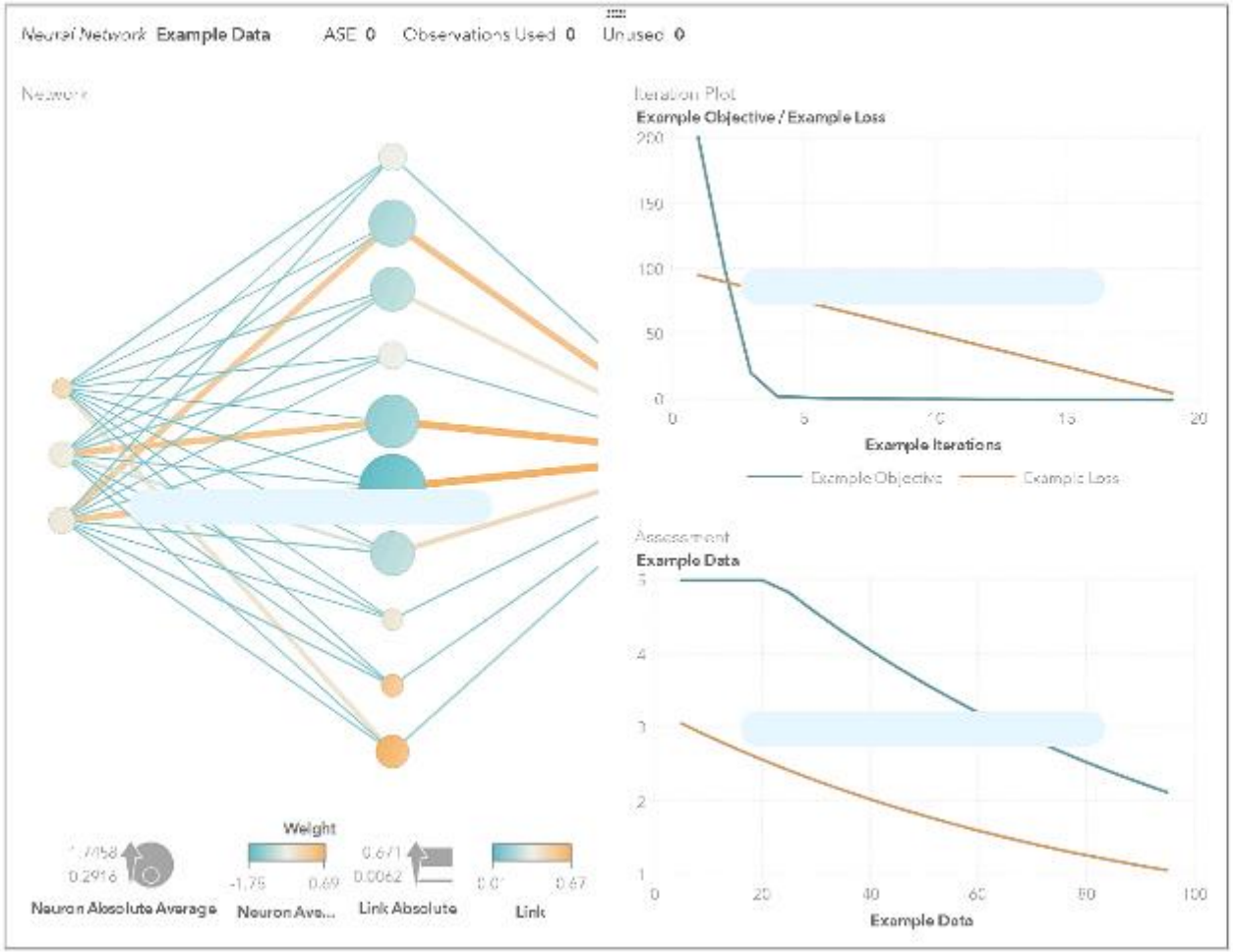
Gradient Boosting

Support Vector Machines

Factorization Machines


Page 1
Page 2
Name
Age
Logistic Regression
ANN
Decision Tree
GBoost
Random Forest
Drop a data item or content here


Page 1
Page 2
Name
Age
Logistic Regression
ANN
Decision Tree
GBoost
Random Forest
Objects
Text Input
Analytics
Network Analysis
Text Topics
Other
Container
Image
Text
Web Content
SAS Visual Statistics
Cluster
Decision Tree
Generalized Linear Model
Linear Regression
Logistic Regression
Model Comparison
SAS Visual Data Mining and Machine Learning
Factorization Machine
Forest
Gradient Boosting
Neural Network
Support Vector Machine
Drop a data item or content here

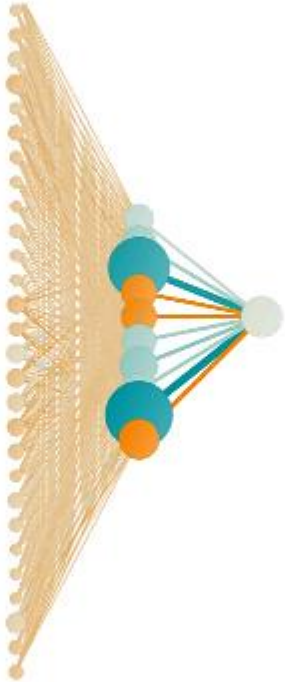


Roles

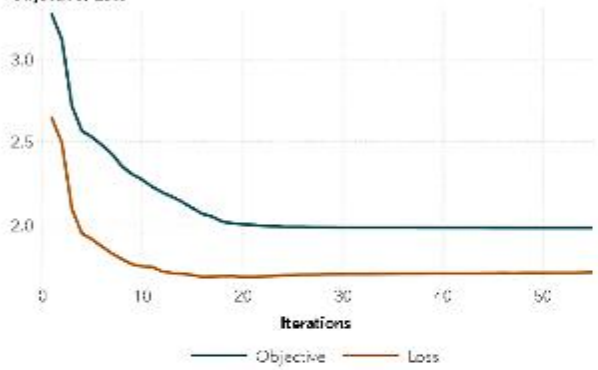
- Response*
  - Add
- Predictors*
  - Add
- Weight
  - Add

Neural Network **Survived** (event=1) Validation Misclassification 0.1788 Observations Used 891

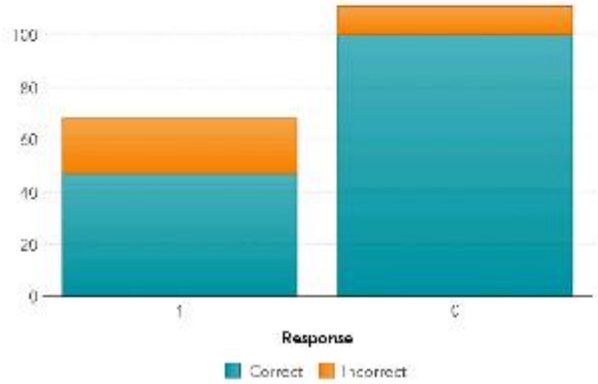
Network



Training Iteration Plot



Validation Misclassification



Roles

- Response
  - Survived
- Predictors
  - Leaf ID (Age)
  - Sex
  - Title_new
  - Sex
  - Pclass
  - Parch
  - SibSp
  - Embarked_new
  - Add
- Weight
  - Add

WHICH
Hyperparameters?

Decision Trees
Max tree depth, splitting criterion

Neural Networks
Network architecture, solver options

Support Vector Machines
Kernel, penalty

Neural Network **Example Data** KS (Youden) **0** Observations Used **0** Unused **0**

Network

The required roles have not been assigned a d...

Weight

1,7458
0,2916
Neuron Abs...

-1,75
0,69
Neuron Av...

0,671
0,0062
Link Absolute

0,01
0,67
Link

Iteration Plot

Example Objective / Example Loss

200
150
100
50
0

The required roles have not been assigned a d...

0
5
10
15
20

Example Iterations

Example Objective
Example Loss

Lift

Cumulative Lift

3,5
3,0
2,5
2,0
1,5
1,0

The required roles have not been assigned a d...

0
20
40
60
80
100

Percentile

Example Model
Example Best

Options

Neural Network 1

Neural Network

▼ General

Autotune:

Autotune...

☐ Include missing

Standardization:

Midrange

Maximum iterations: 250

Maximum time(sec): 270

Optimization method:

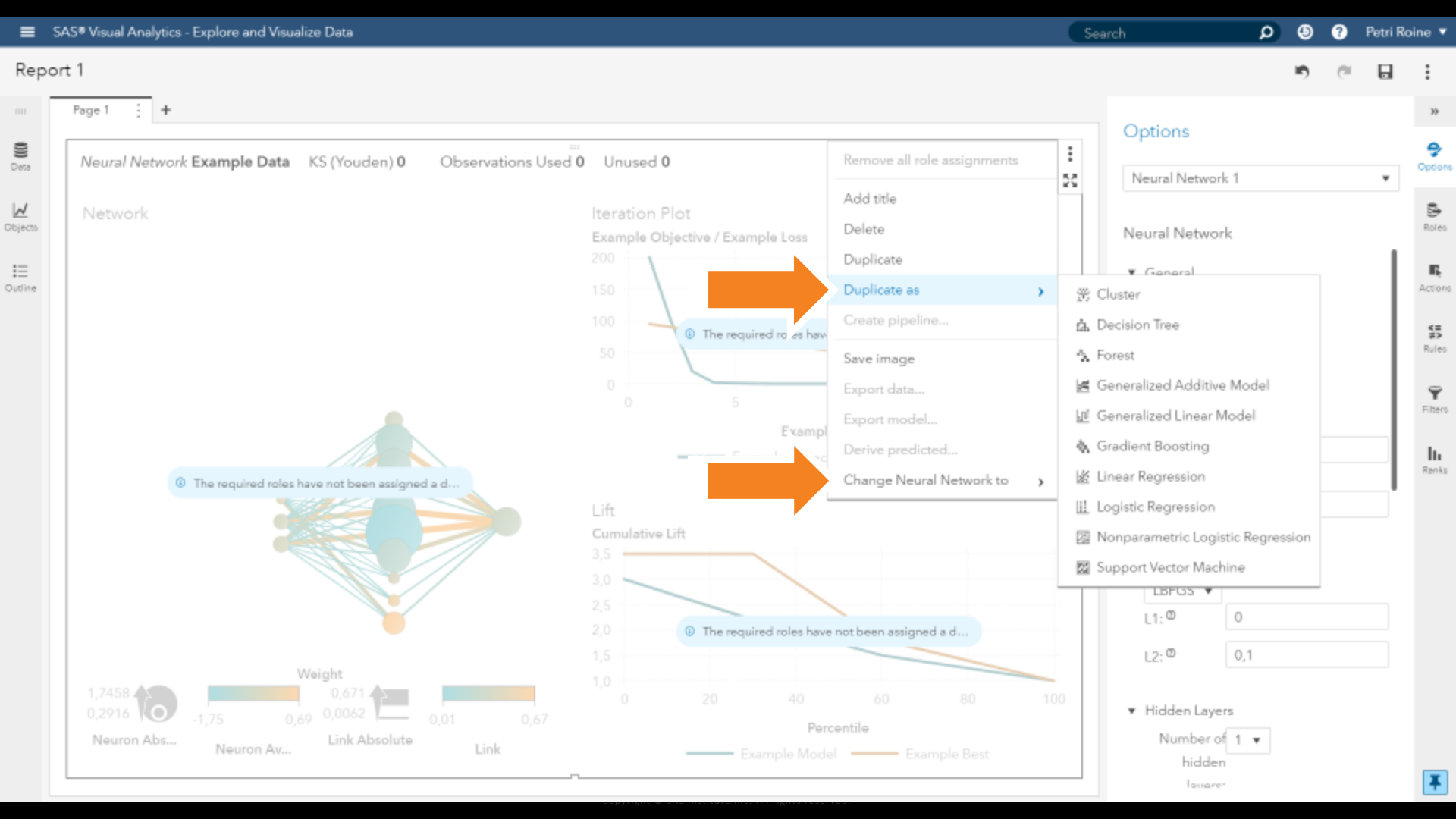
LBFGS

L1: 0

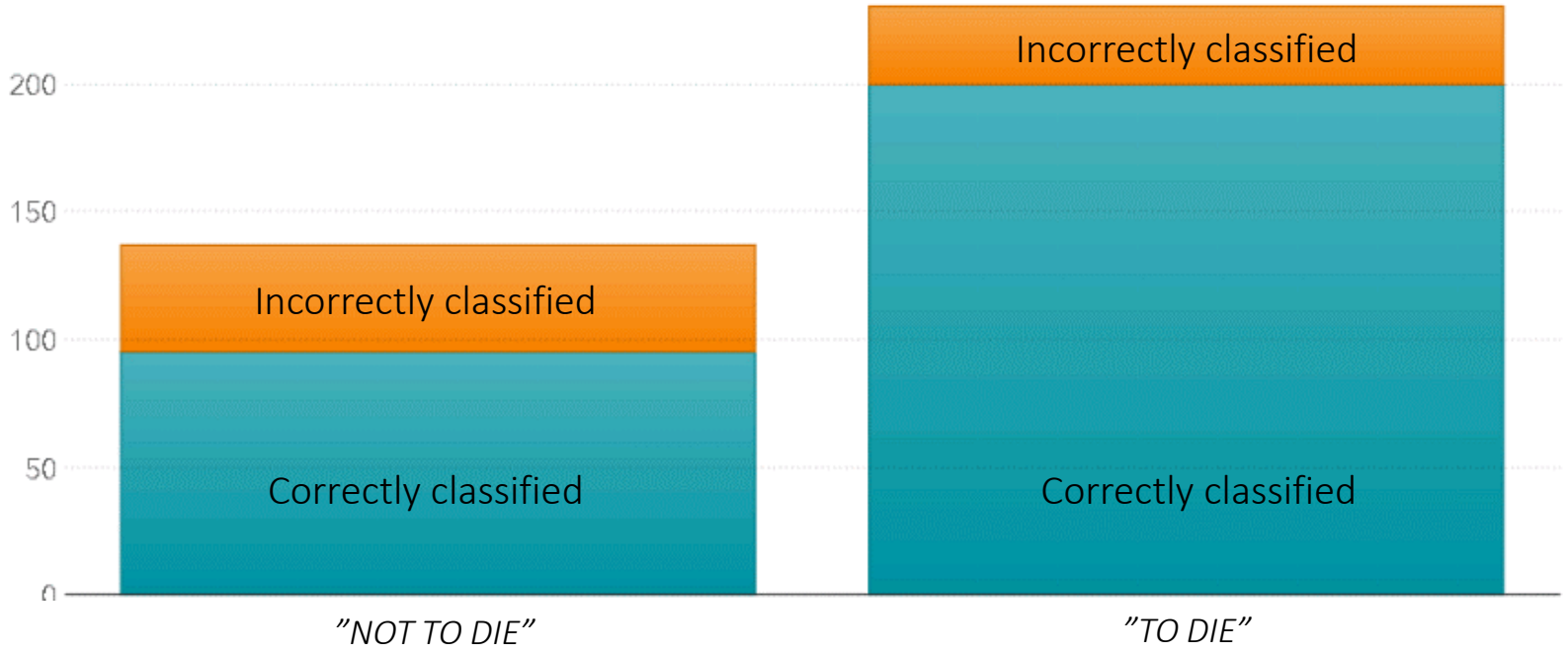
L2: 0,1

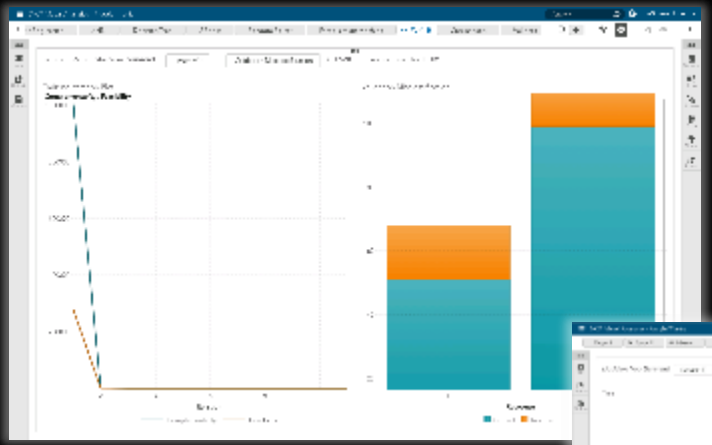
▼ Hidden Layers

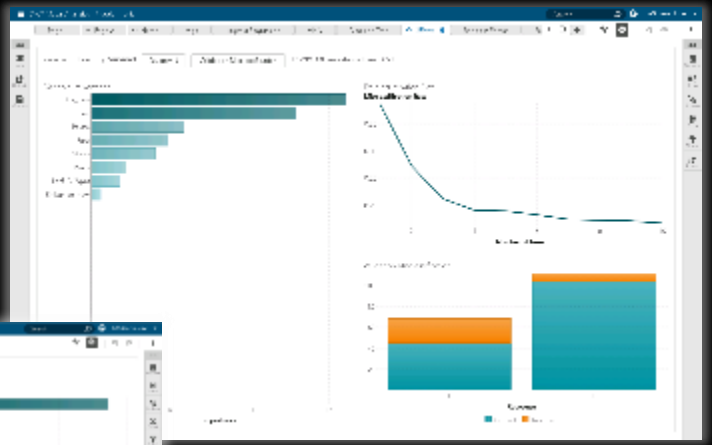
Number of hidden layers: 1

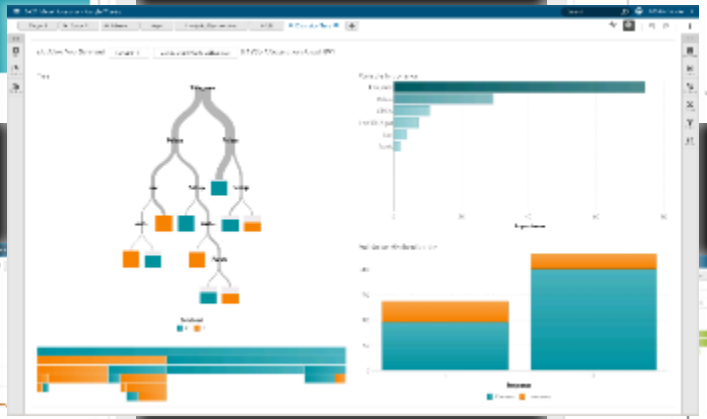

SAS® Visual Analytics - Explore and Visualize Data
Search
Petri Roine
Report 1
Page 1
Neural Network Example Data
KS (Youden) 0
Observations Used 0
Unused 0
Network
Iteration Plot
Example Objective / Example Loss
200
150
100
50
0
5
The required roles have not been assigned a d...
Weight
1,7458
0,2916
-1,75
0,69
0,671
0,0062
0,01
0,67
Neuron Abs...
Neuron Av...
Link Absolute
Link
Lift
Cumulative Lift
3,5
3,0
2,5
2,0
1,5
1,0
0
20
40
60
80
100
Percentile
Example Model
Example Best
Remove all role assignments
Add title
Delete
Duplicate
Duplicate as
Create pipeline...
Save image
Export data...
Export model...
Derive predicted...
Change Neural Network to
Cluster
Decision Tree
Forest
Generalized Additive Model
Generalized Linear Model
Gradient Boosting
Linear Regression
Logistic Regression
Nonparametric Logistic Regression
Support Vector Machine
Options
Neural Network 1
Neural Network
General
LBFGS
L1:
0
L2:
0,1
Hidden Layers
Number of hidden layers
1
Data
Objects
Outline
Options
Roles
Actions
Rules
Filters
Ranks

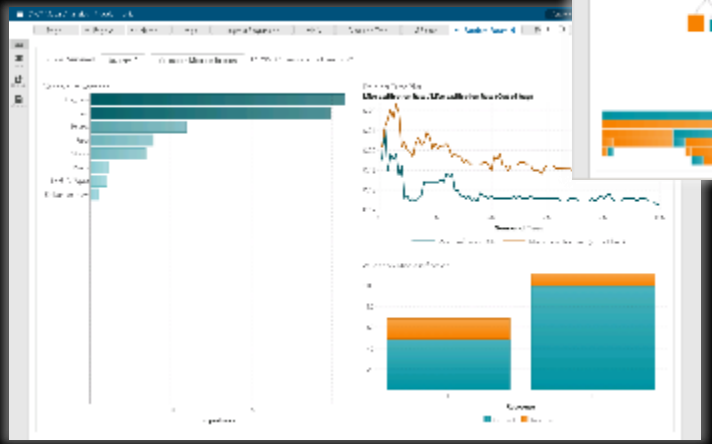
Validation Misclassification

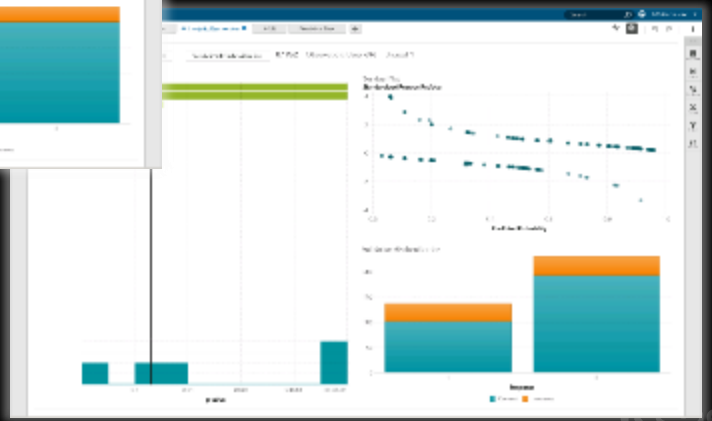






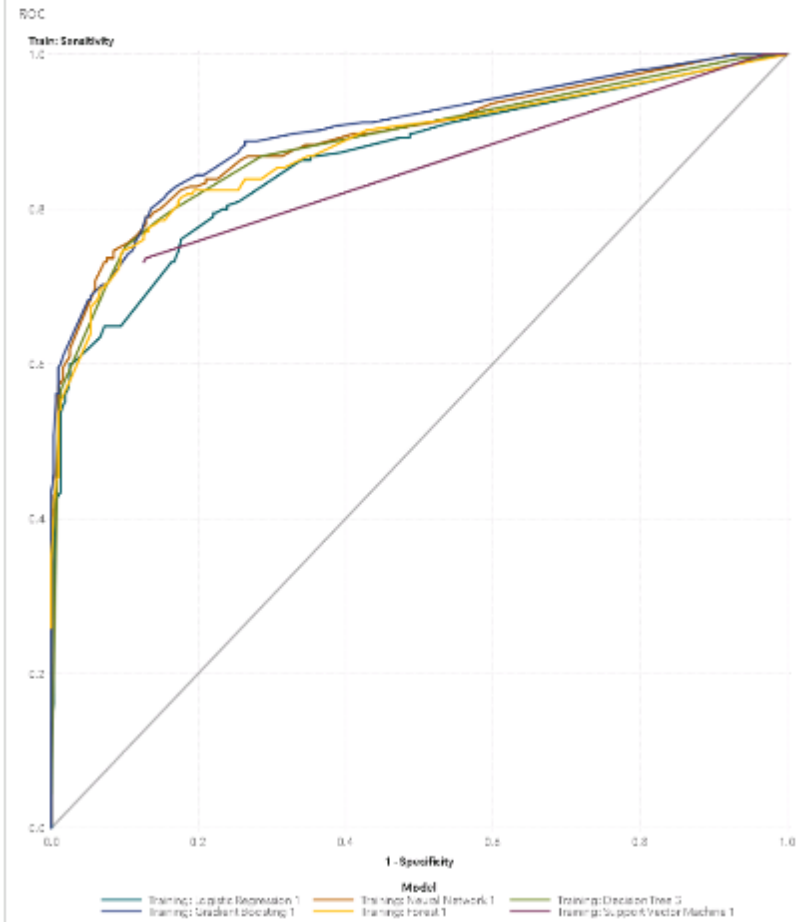


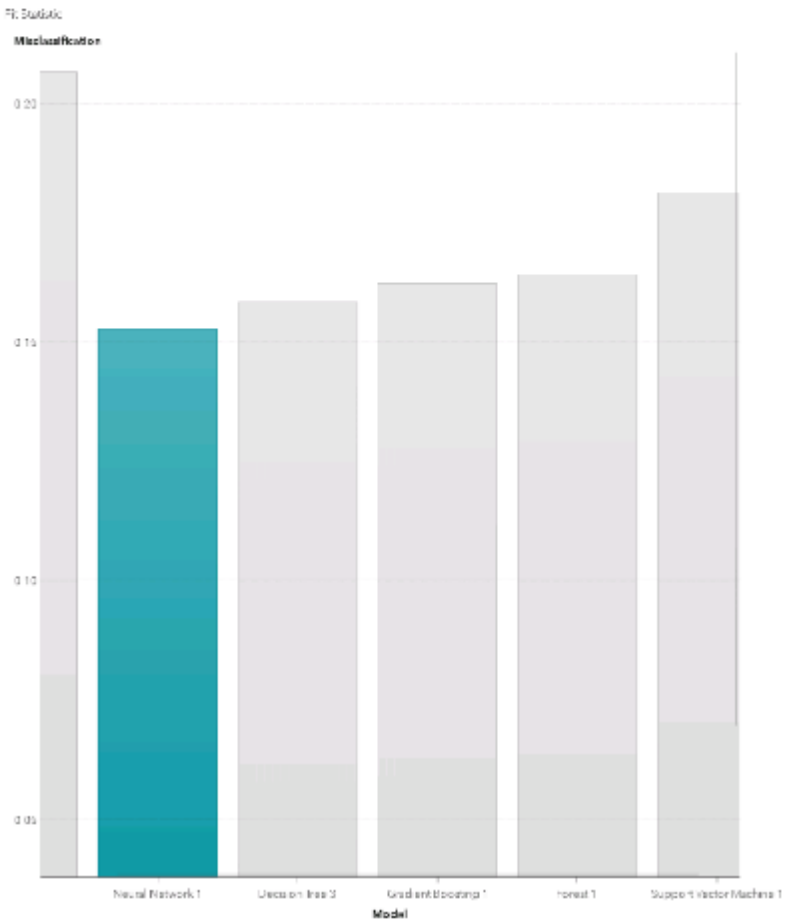




ASSESS MODELS

Model comparison

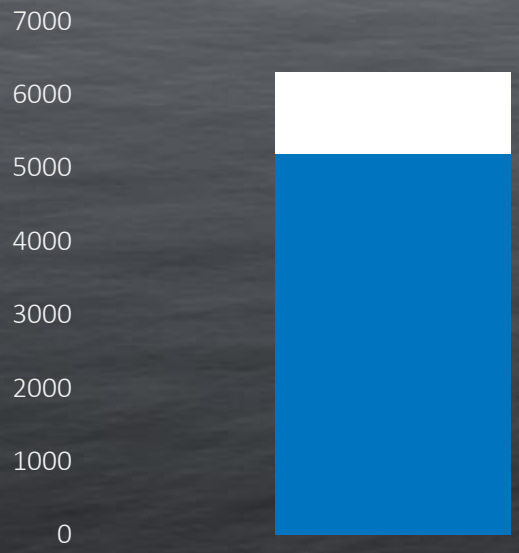

ROC
Train: Sensitivity
1 - Specificity
Model
Training: Logistic Regression 1
Training: Gradient Boosting 1
Training: Neural Network 1
Training: Forest 1
Training: Decision Tree 1
Training: Support Vector Machine 1


Fit Statistic
Misclassification
Neural Network 1
Decision Tree 1
Gradient Boosting 1
Forest 1
Support Vector Machine 1
Model

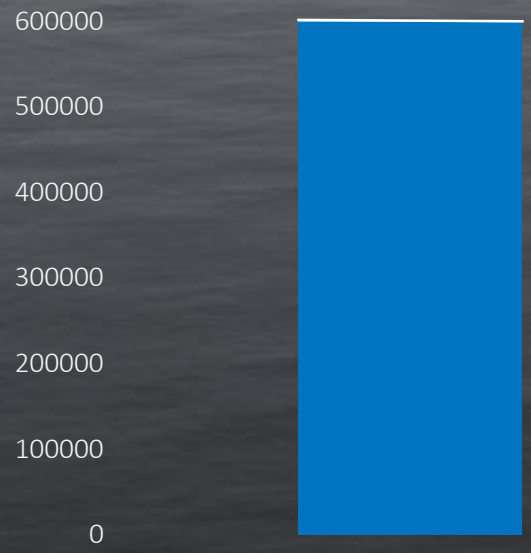
Overview Data Kernels Discussion Leaderboard More My Submissions **Submit Predictions**

SUBMIT_NNEUR.csv a month ago by PetriROI Neural Network	0.75120	☑
SUBMIT_LREGR.csv a month ago by PetriROI Logistic Regression	0.79904	☑
SUBMIT_DTREE.csv a month ago by PetriROI Decision Tree	0.74641	☑

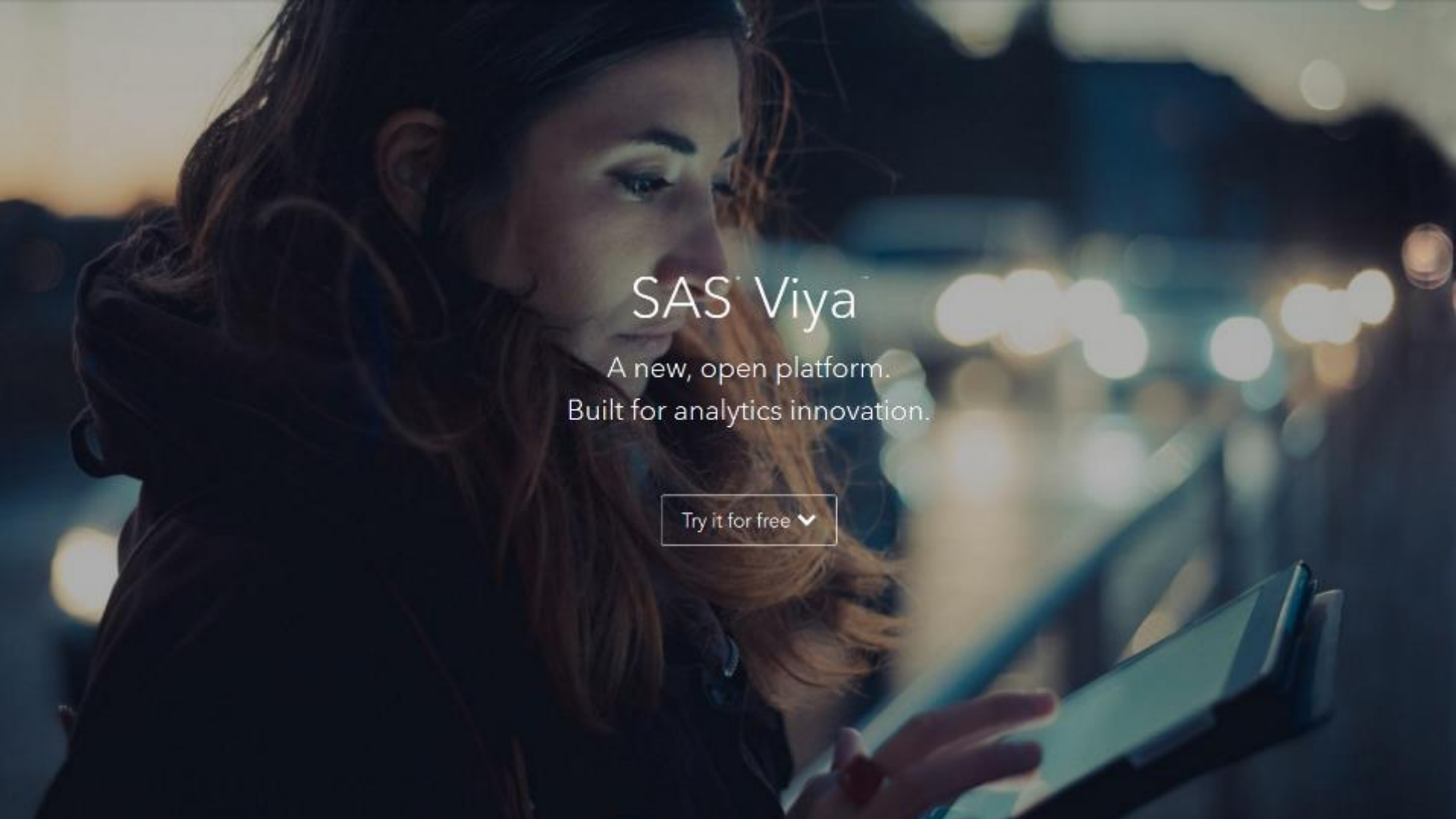
1110	▼ 111	SergeyKuntsev		0.79904	48	23d
1111	▼ 111	Elizabeth M. Cruz		0.79904	25	20d
1112	▼ 111	Nilson de Lima Jr		0.79904	3	1mo
1113	▼ 111	PetriROI		0.79904	10	1mo
1114	▼ 111	StevenJin		0.79904	10	1mo
1115	▼ 111	lunge		0.79904	21	1mo
1116	▼ 111	SaloniKakkar		0.79904	31	1mo
1117	▼ 111	Naoki Narimatsu		0.79904	16	1mo
1118	▼ 111	srishtideepani		0.79904	3	1mo
1119	▼ 111	Dmitry Kovalev		0.79904	12	10d
1120	▼ 111	Md Sadik Hussain		0.79904	4	1mo
1121	▼ 111	Koffi		0.79904	32	24d



Titanic



Kaggle


SAS Viya
A new, open platform.
Built for analytics innovation.
Try it for free