Machine Learning Engineer Nanodegree Program
Capstone Project

Bruno Arcelino da Silva
Janeiro, 2018

Prevendo Salários de desenvolvedores a partir da pesquisa da StackOverflow em 2017

Definição

Visão Geral do projeto

A área de desenvolvimento de software tem uma grande variedade de salário onde as premissas não são nenhum pouco claras, então temos uma grande discrepância de salários.

A intenção desse projeto é pegar dados da pesquisa da Stackoverflow de 2017, onde teve 64 mil desenvolvedores respondendo a essa pesquisa. Seguindo os dados desta pesquisa, podemos criar um modelo de aprendizagem supervisionada que irá predizer o salário do desenvolvedor.

Em paralelo, temos a intenção de descobrir quais são os tópicos mais valorizados para um desenvolvedor.

Os dados utilizados para a montagem do modelo vão ser de acordo com o país escolhido, nos testes vamos utilizar o Estados Unidos pela quantidade de usuários serem a maior, mas o modelo pode ser generalizado para qualquer pais.

Os dados da pesquisa podem ser encontrados no kaggle:
https://www.kaggle.com/stackoverflow/so-survey-2017/

Descrição do problema

O problema vai ser resolvido com uma abordagem de um aprendizado supervisionado. Vamos utilizar regressões para essa abordagem, sendo o salário como variavel alvo e as perguntas como atributos.

Métricas

Nos problemas de regressão é usado normalmente a métrica R2 Score, e vamos acatar como uma das nossas métricas para esse estudo. O R2 Score pode ser chamado também de “Coeficiente de determinação” que nós da medida da variância da nossa variável alvo.

O coeficiente pode ir de zero (muito ruim) até um (modelo perfeito).

Outra métrica será a média do erro ao quadrado, onde faz a media da diferença entre o valor estimado e o valor real. Quanto mais próximo do zero, melhor o modelo é.

Análise

Exploração dos dados

Na base de dados fornecido pelo Kaggle, temos 51.392 participantes da pesquisa de 201 países. Vamos considerar a análise principal com o país Estados Unidos, então teremos 11.455 participantes.

Esse dataset tem um número grande de features, 154 ao total. Existem diversas informações, desde qual IDE o desenvolvedor usa até a metodologia do trabalho utilizados.

Como descrever 154 features seria um trabalho muito exaustivo e iria poluir o documento, vou descrever as características principais.

- Professional: Se o usuário trabalha na área, ele pode ser professional, estudante ou trabalha na área mas não é desenvolvedor
- ProgramHobby: Diz se o usuário programa no tempo livre
- Country: País do usuário
- University: Diz se o usuário está frequentando a faculdade
- EmploymentStatus: O estado de trabalho do usuário
- FormalEducation: Qual é o nível de escolaridade do usuário
- MajorUndergrade: Qual é a graduação feita pelo usuário
- CompanySize: Tamanho da empresa que ele atua
- YearsProgram: Anos que ele programa
- YearsCodedJob: Anos que ele trabalha com programação
- DeveloperType: Tipo de desenvolvimento, como web developer e mobile developer
- HaveWorkedLanguage: Quais linguagens o usuário trabalha
- HaveWorkedDatabase: Quais bancos o usuário trabalha
- HaveWorkedPlatfom: Quais plataformas o usuário trabalha
- Methodology: Quais metodologias de programação o usuário utiliza
- Salary: O salário que o usuário recebe no atual trabalho

Primeiramente os dados que vamos usar vai ser de pessoas do Estados Unidos da América, já que se consideramos todos os países os valores dos salários vão variar demais, tanto pela moeda do pais e como é o trabalho em si em um pais. Então vamos usar todos que trabalham

no Estados Unidos, estão empregados full-time e recebem em dólares, com isso temos 5.067 (sem retirar os nulos ainda) dos 11.455 americanos das 51.392 que tínhamos em todo dataset.

Todas as perguntas desta pesquisa não foram obrigatórias, então temos uma porcentagem alta de questões respondidas com “NA”, nesse caso podemos fazer duas coisas:

- Considerar NA como uma resposta, assim conseguindo manter o volume de dados do dataset
- Excluir as linhas que contém qualquer NA, porém vamos diminuir em muito o dataset já pequeno

No caso da nossa label, não podemos permitir um NA, então somos obrigados a excluir qualquer linha sem “Salary”.

Temos apenas três entradas que são númericas: “CareerSatisfaction”, “JobSatisfaction", “HoursPerWeek” e a própria label “Salary”.

Logo que a grande maioria dos inputs são categóricos, cada resposta vai virar uma variável categórica.

Existem perguntas que podem ser respondidas com varias respostas, como “HaveWorkedLanguage” (imagem abaixo), nesses casos também vamos criar uma variável categórica para cada uma das respostas possível na questão.

Agora vamos explorar um pouco mais a nossa variável alvo

	Salary
count	3719.000000
mean	97083.681904
std	34246.971246
min	5.000000
25%	71000.000000
50%	95000.000000
75%	120000.000000
max	197000.000000

	Salary
count	3343.000000
mean	96491.234520
std	27570.244612
min	47800.000000
25%	74000.000000
50%	95000.000000
75%	118000.000000
max	158100.000000

Algumas linhas estão vindo com “Salary” somente com 5, já que o mínimo é 5, então devemos pelo menos fazer uma validação para serem maiores que 5. Na segunda imagem mostra a

descrição do “Salary” retirando as linhas que os quartis são maiores do que 95% e menores do que 5%, com essas alterações perdemos 376 linhas. Mesmo com algumas perdas de linha, os dados parecem um pouco mais consistente (principalmente o valor mínimo).

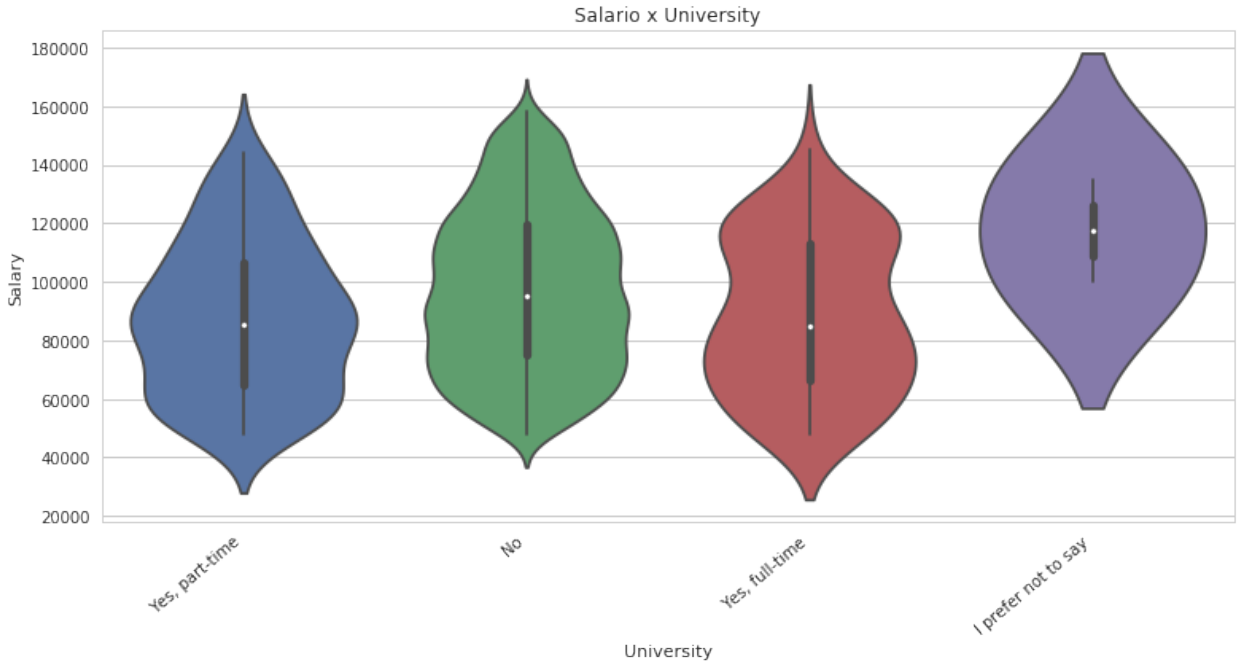
Visualização exploratória

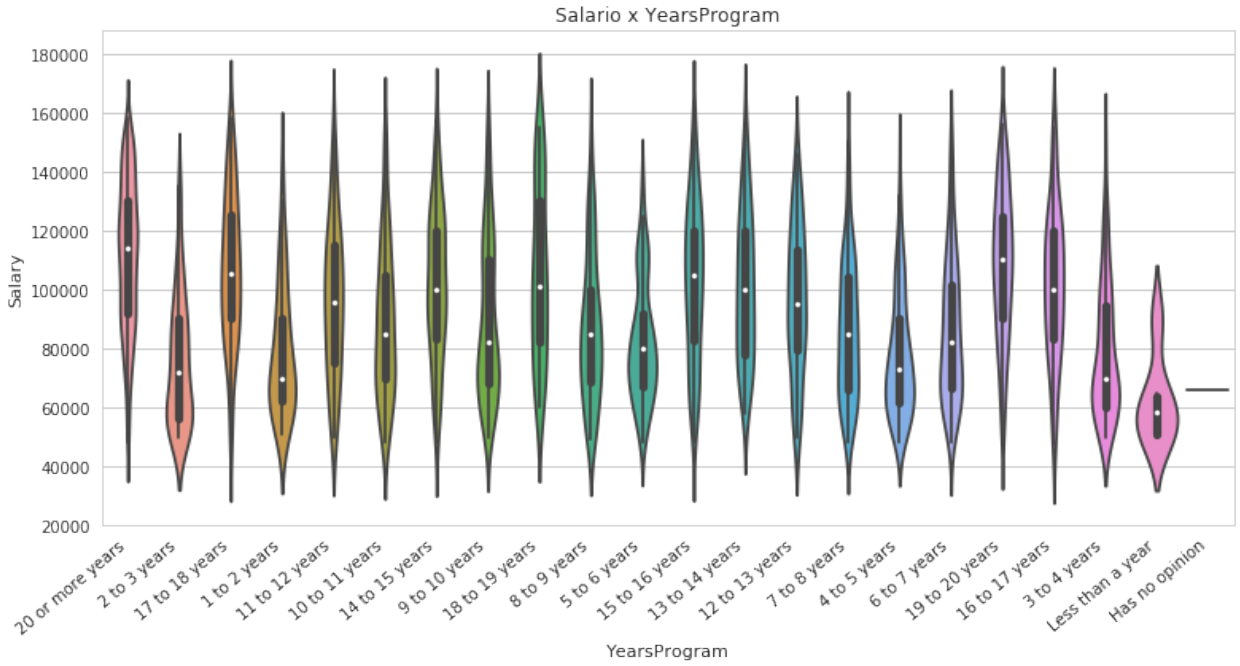
Vamos fazer uma pesquisa de correlação entre as variáveis que tem apenas uma resposta.

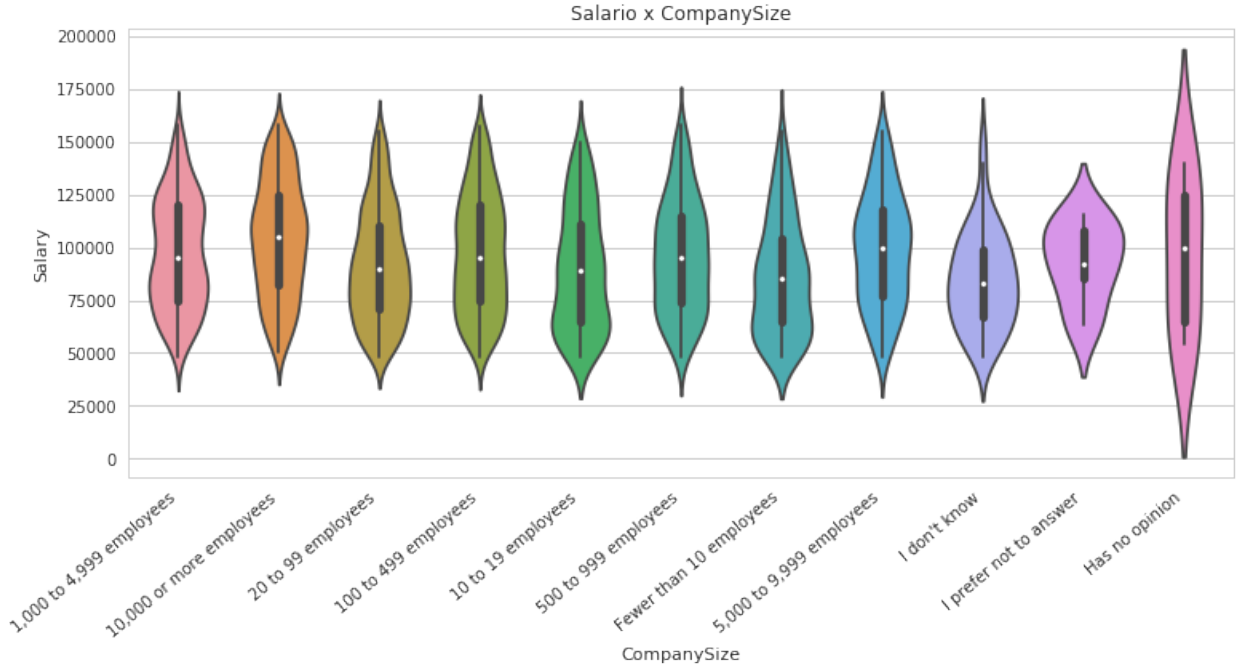
Estaremos utilizando um gráfico da biblioteca Seaborn, o violin plot, onde ele já pega a nossa variável categórica e já relaciona a uma variável da nossa escolhe, que será o “Salary”.

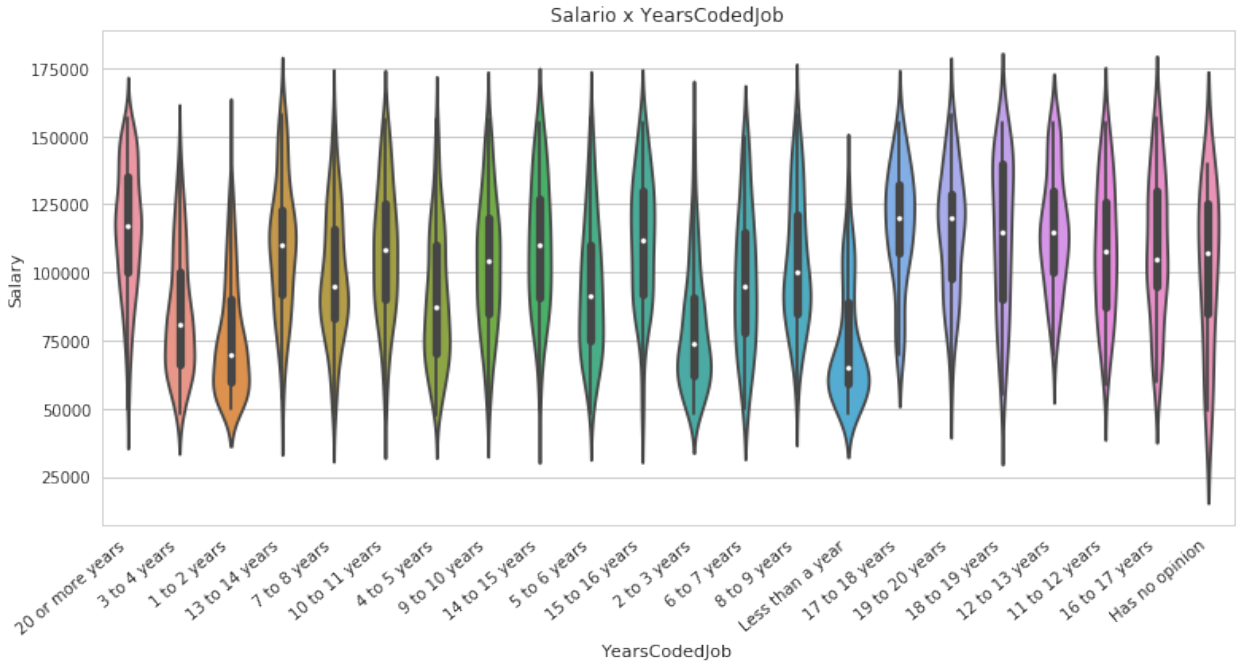
Vilolin plot é uma junção de box plot com density plot, onde mostra todas as informações das categorias correlacionadas ao eixo Y e a distribuição do dado em si.

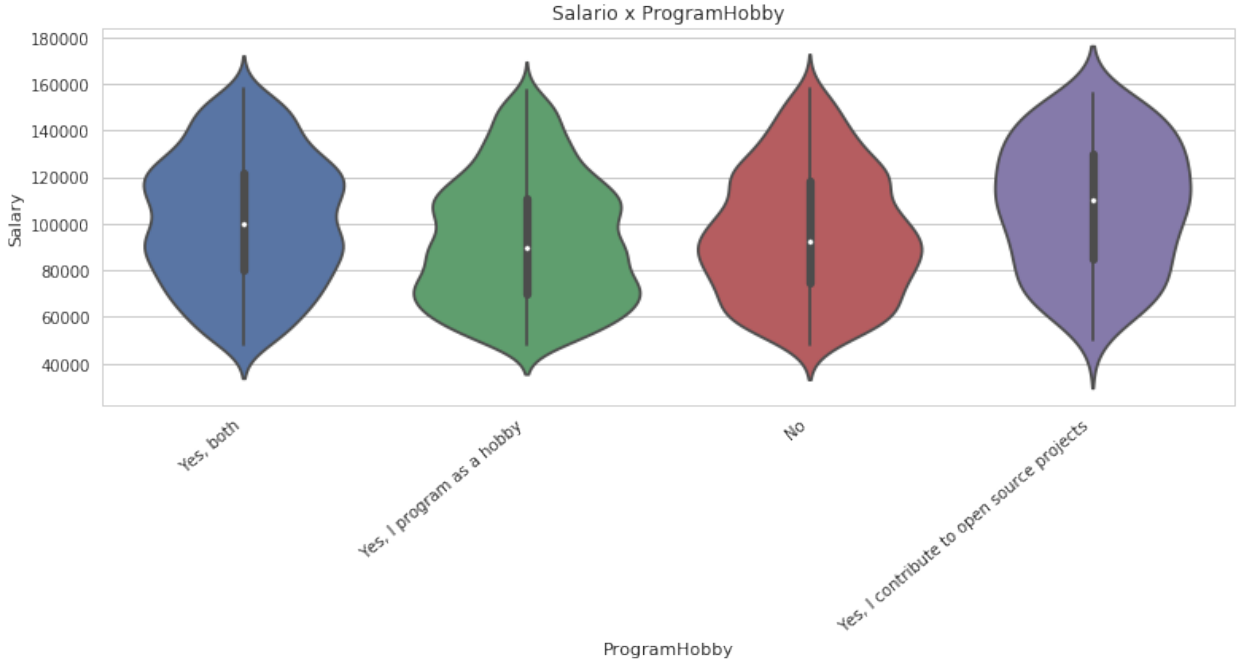
Os Vilolin plots:



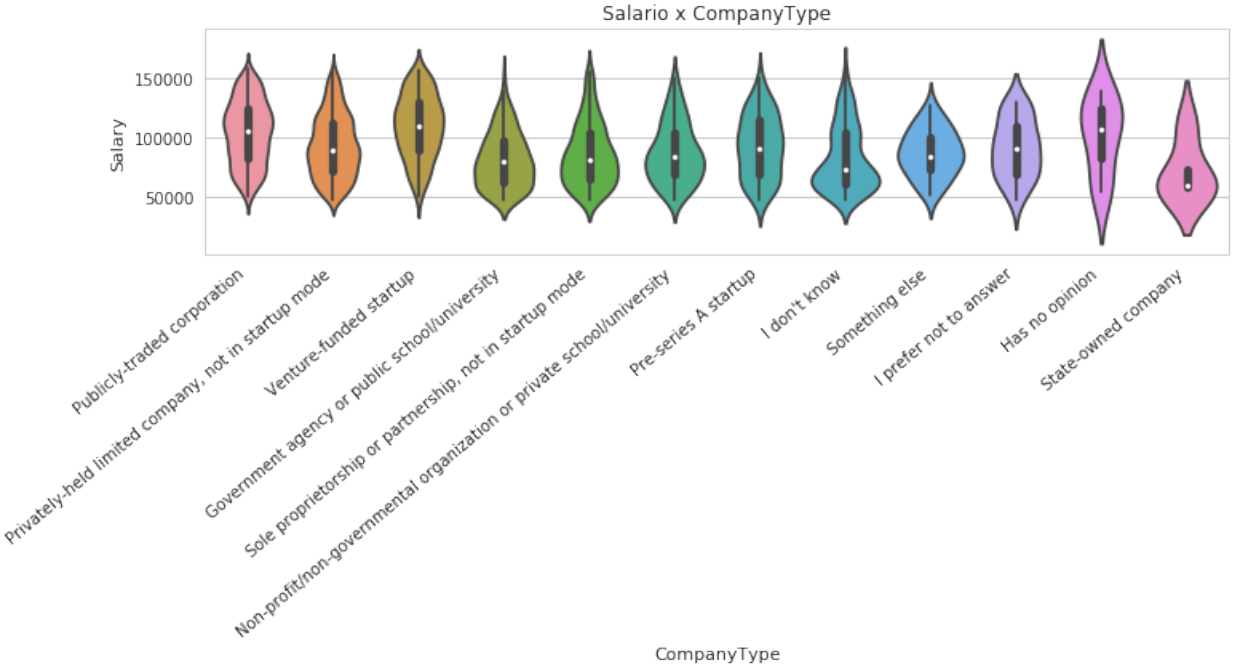

Salario x YearsProgram
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
20 or more years
2 to 3 years
17 to 18 years
1 to 2 years
11 to 12 years
10 to 11 years
14 to 15 years
9 to 10 years
18 to 19 years
8 to 9 years
5 to 6 years
15 to 16 years
13 to 14 years
12 to 13 years
7 to 8 years
4 to 5 years
6 to 7 years
19 to 20 years
16 to 17 years
3 to 4 years
Less than a year
Has no opinion
YearsProgram

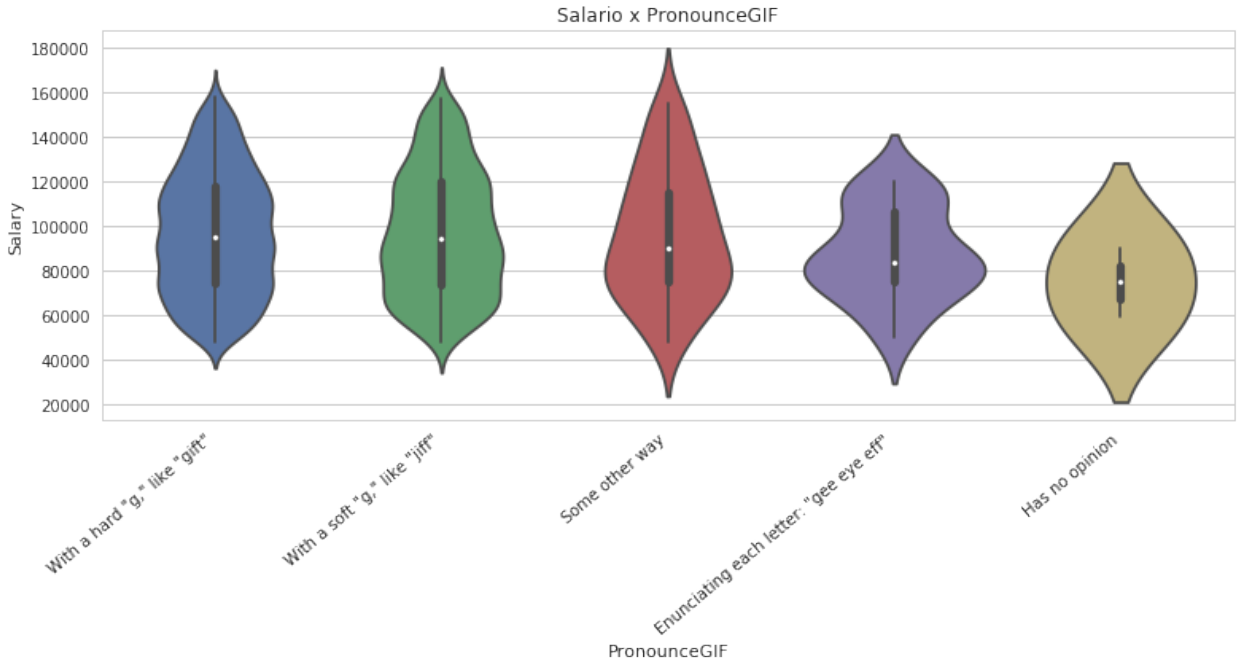

Salario x CompanySize
Salary
200000
175000
150000
125000
100000
75000
50000
25000
0
1,000 to 4,999 employees
10,000 or more employees
20 to 99 employees
100 to 499 employees
10 to 19 employees
500 to 999 employees
Fewer than 10 employees
5,000 to 9,999 employees
I don't know
I prefer not to answer
Has no opinion
CompanySize

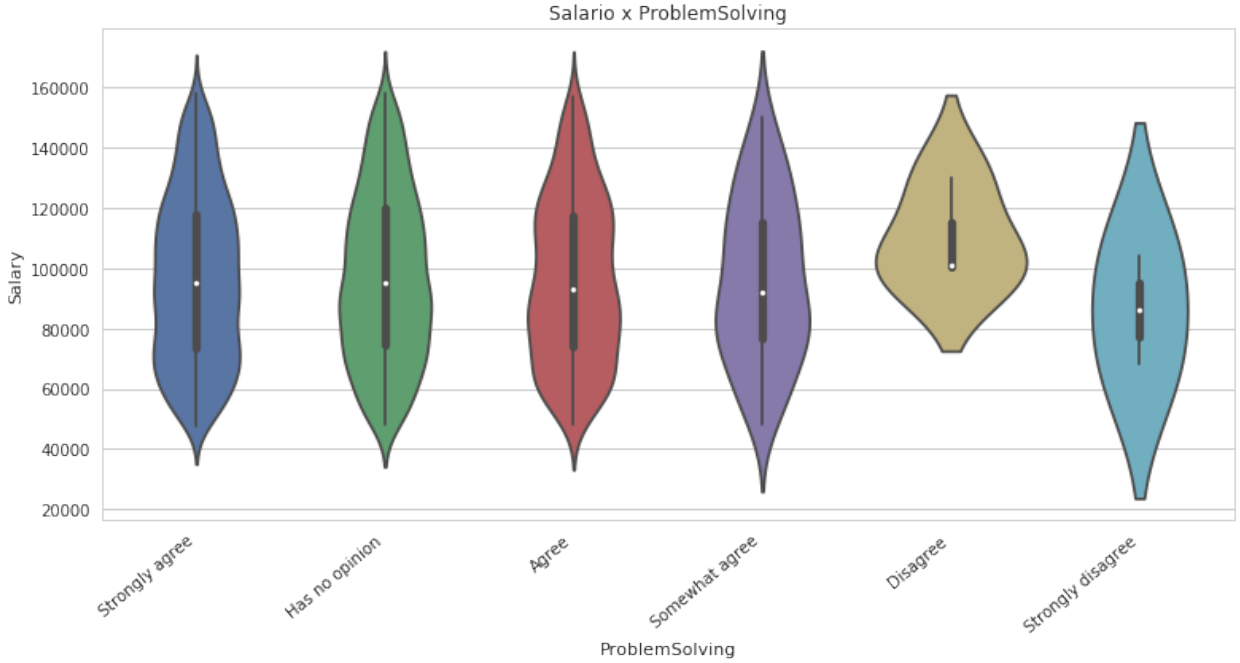

Salario x YearsCodedJob
Salary
175000
150000
125000
100000
75000
50000
25000
20 or more years
3 to 4 years
1 to 2 years
13 to 14 years
7 to 8 years
10 to 11 years
4 to 5 years
9 to 10 years
14 to 15 years
5 to 6 years
15 to 16 years
2 to 3 years
6 to 7 years
8 to 9 years
Less than a year
17 to 18 years
19 to 20 years
18 to 19 years
12 to 13 years
11 to 12 years
16 to 17 years
Has no opinion
YearsCodedJob

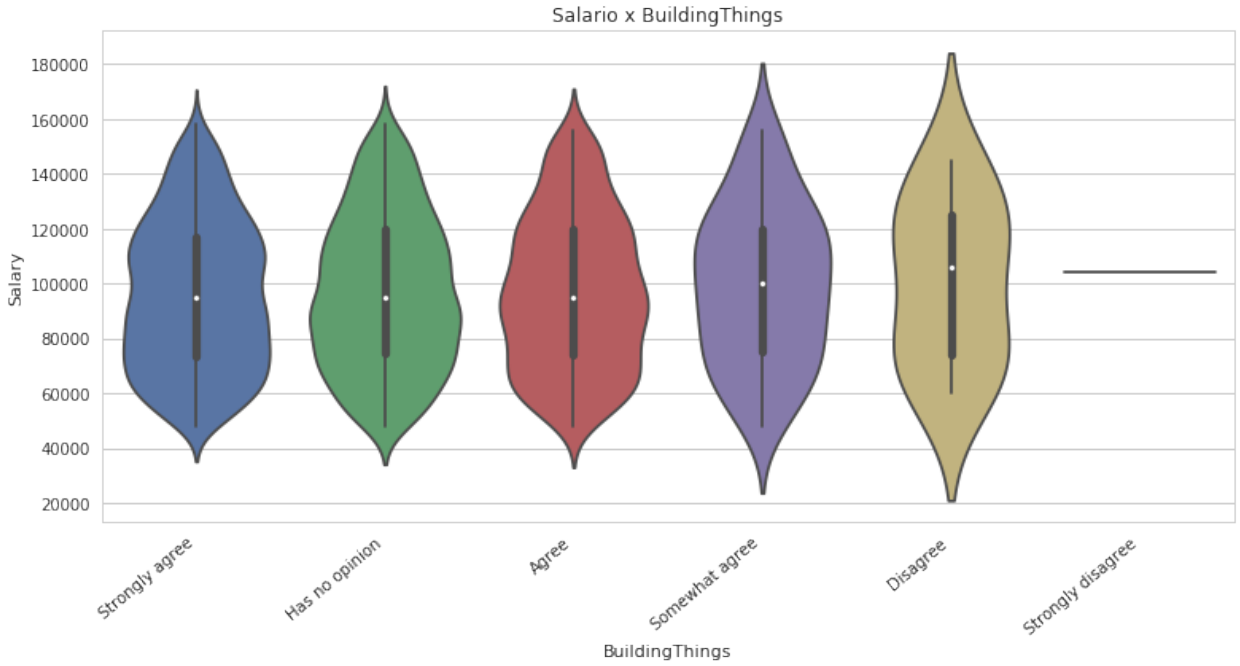

Salario x ProgramHobby
Salary
180000
160000
140000
120000
100000
80000
60000
40000
Yes, both
Yes, I program as a hobby
No
Yes, I contribute to open source projects
ProgramHobby

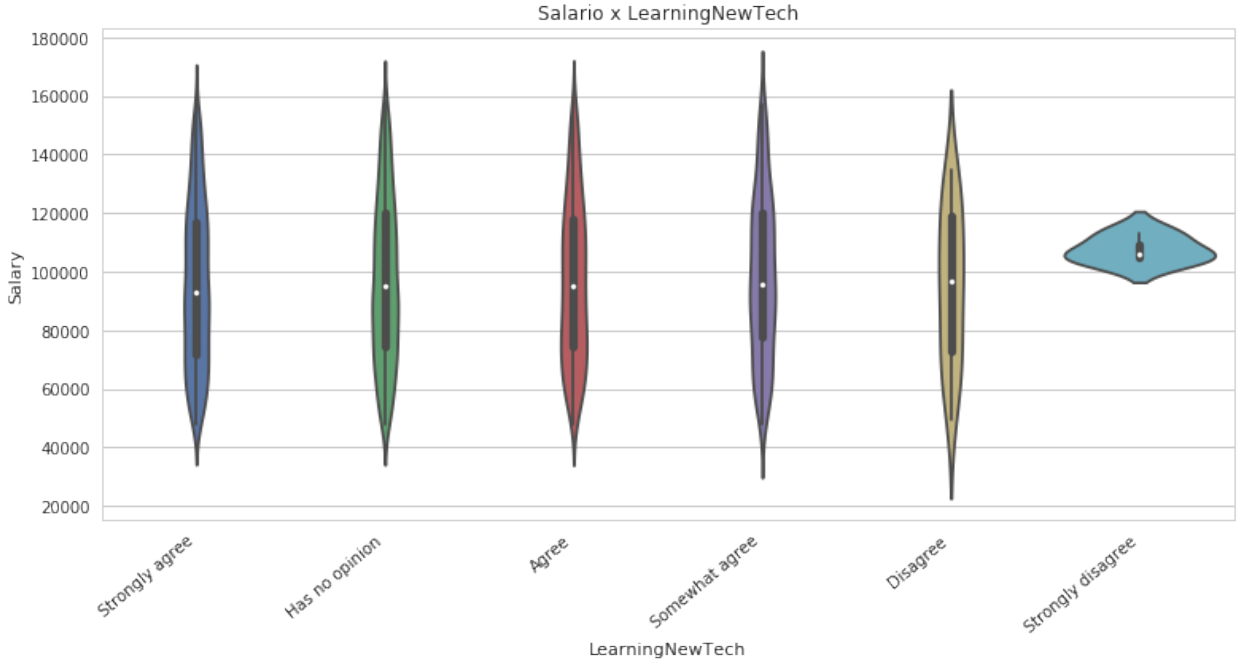

Salario x HomeRemote
Salary
180000
160000
140000
120000
100000
80000
60000
40000
All or almost all the time (I'm full-time remote)
A few days each month
More than half, but not all, the time
Never
Less than half the time, but at least one day each week
It's complicated
About half the time
Has no opinion
HomeRemote

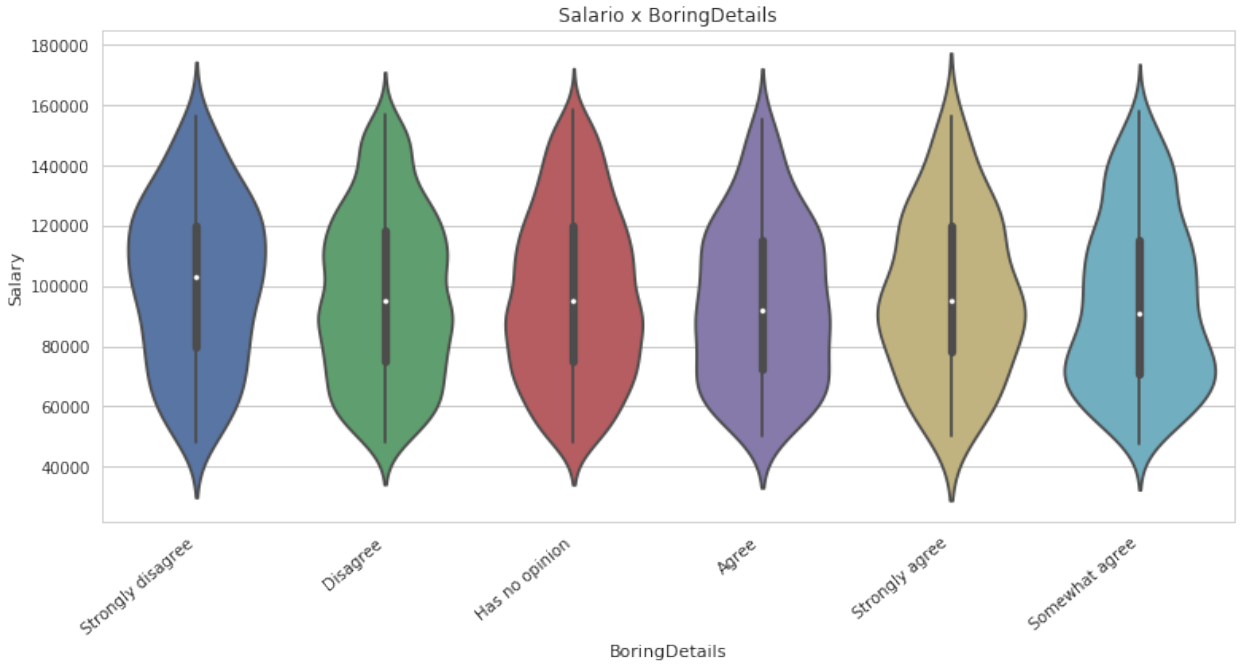

Salario x CompanyType
Salary
150000
100000
50000
Publicly-traded corporation
Privately-held limited company, not in startup mode
Venture-funded startup
Government agency or public school/university
Sole proprietorship or partnership, not in startup mode
Non-profit/non-governmental organization or private school/university
Pre-series A startup
I don't know
Something else
I prefer not to answer
Has no opinion
State-owned company
CompanyType

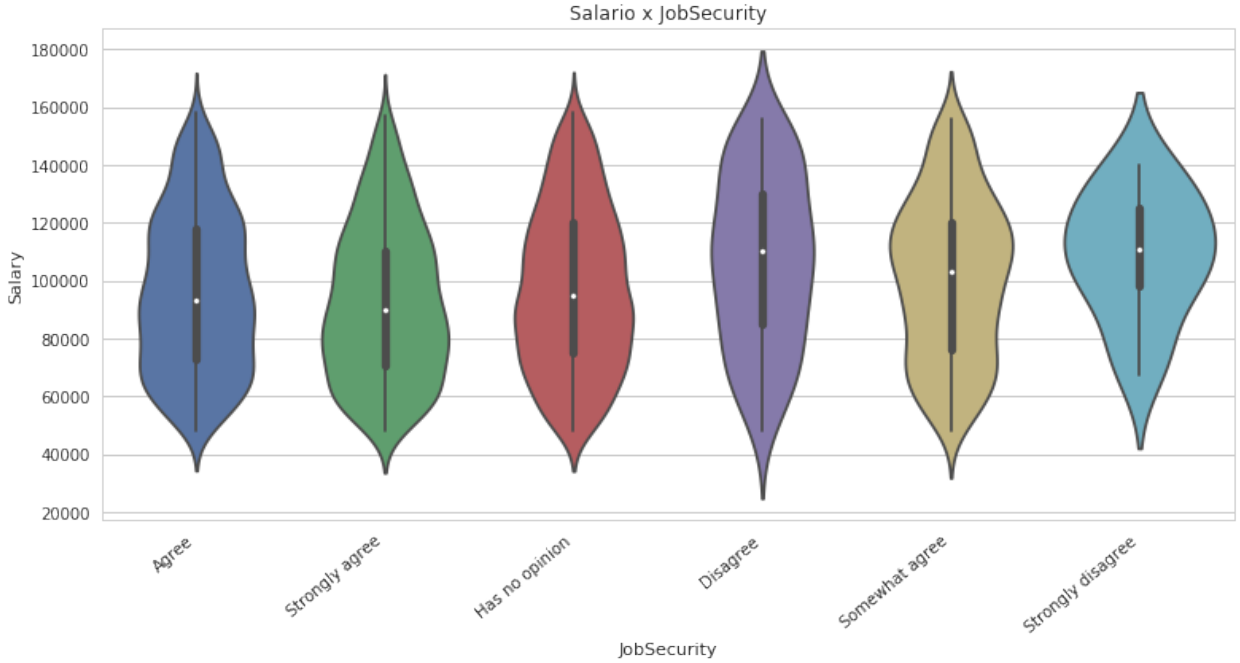

Salario x PronounceGIF
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
With a hard "g," like "gift"
With a soft "g," like "jiff"
Some other way
Enunciating each letter: "gee eye eff"
Has no opinion
PronounceGIF

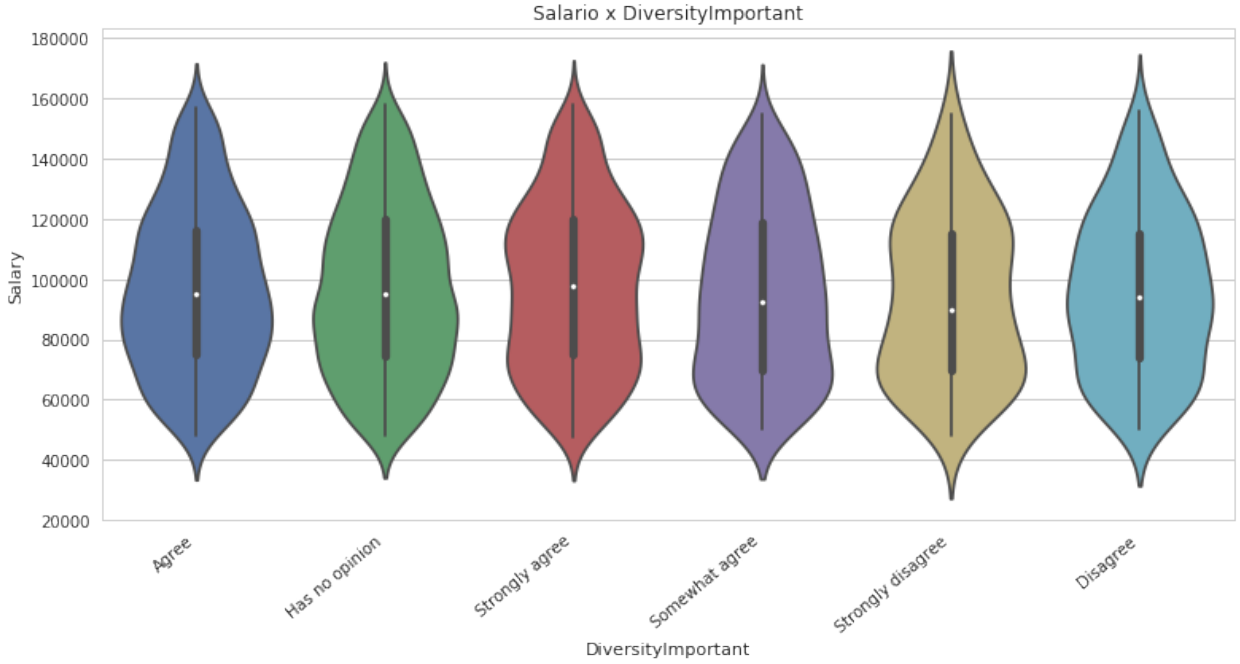

Salario x ProblemSolving
Salary
160000
140000
120000
100000
80000
60000
40000
20000
Strongly agree
Has no opinion
Agree
Somewhat agree
Disagree
Strongly disagree
ProblemSolving

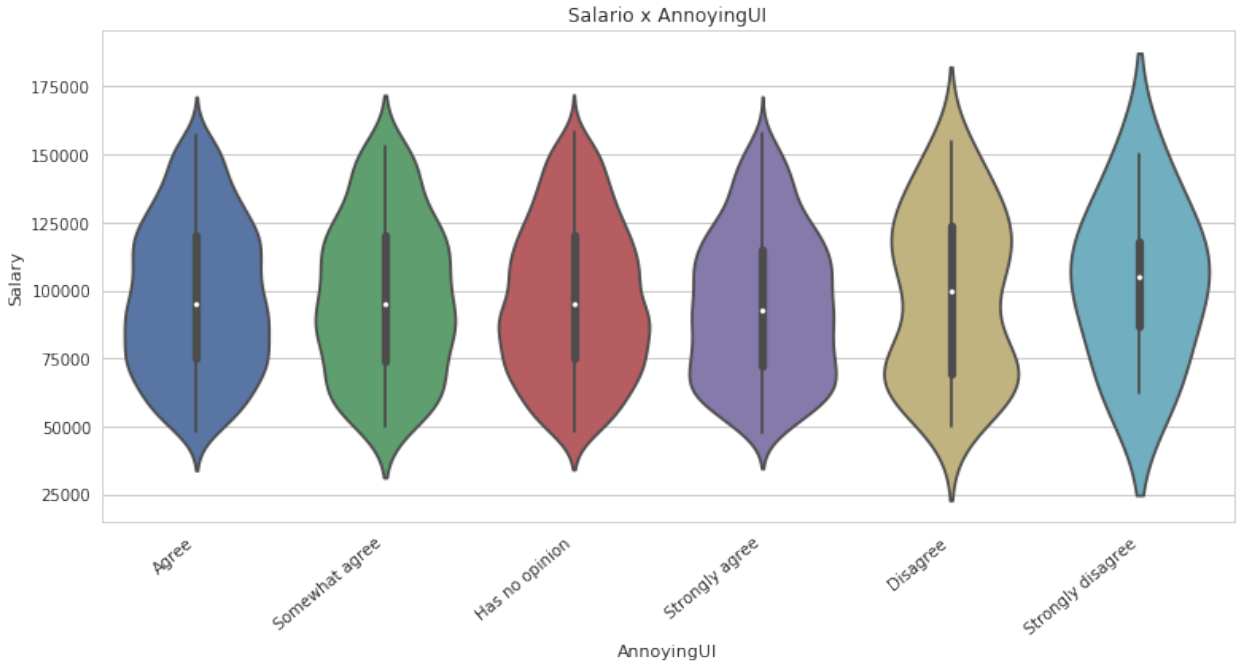

Salario x BuildingThings
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Strongly agree
Has no opinion
Agree
Somewhat agree
Disagree
Strongly disagree
BuildingThings

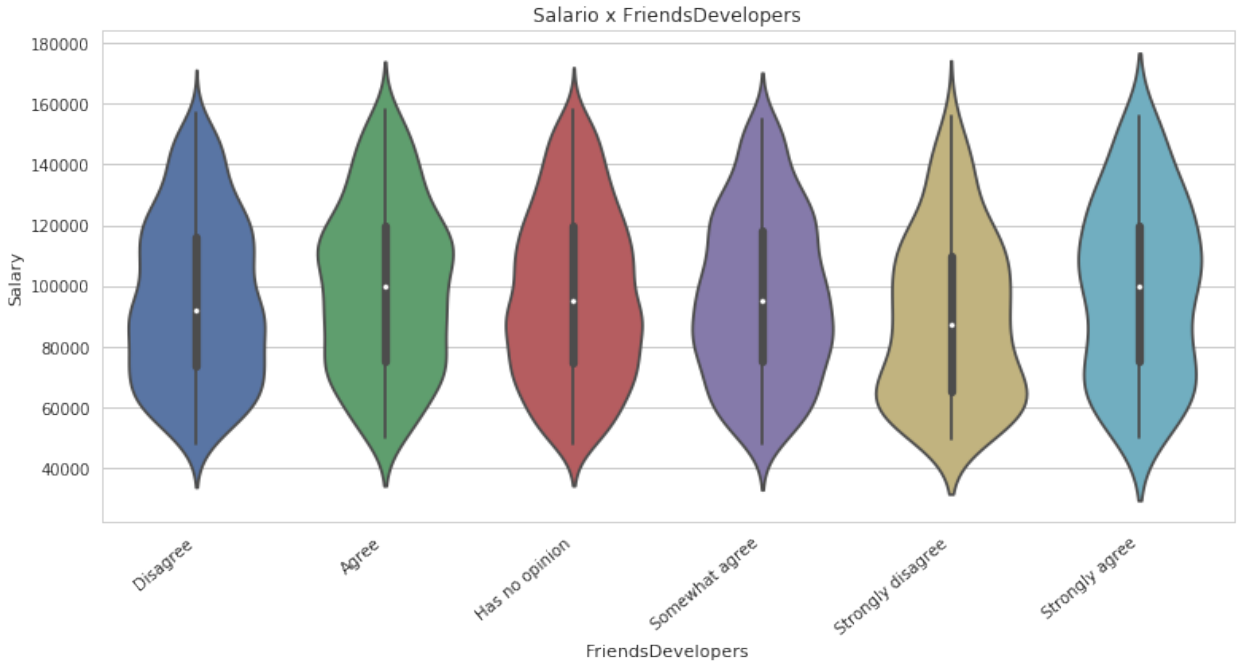

Salario x LearningNewTech
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Strongly agree
Has no opinion
Agree
Somewhat agree
Disagree
Strongly disagree
LearningNewTech

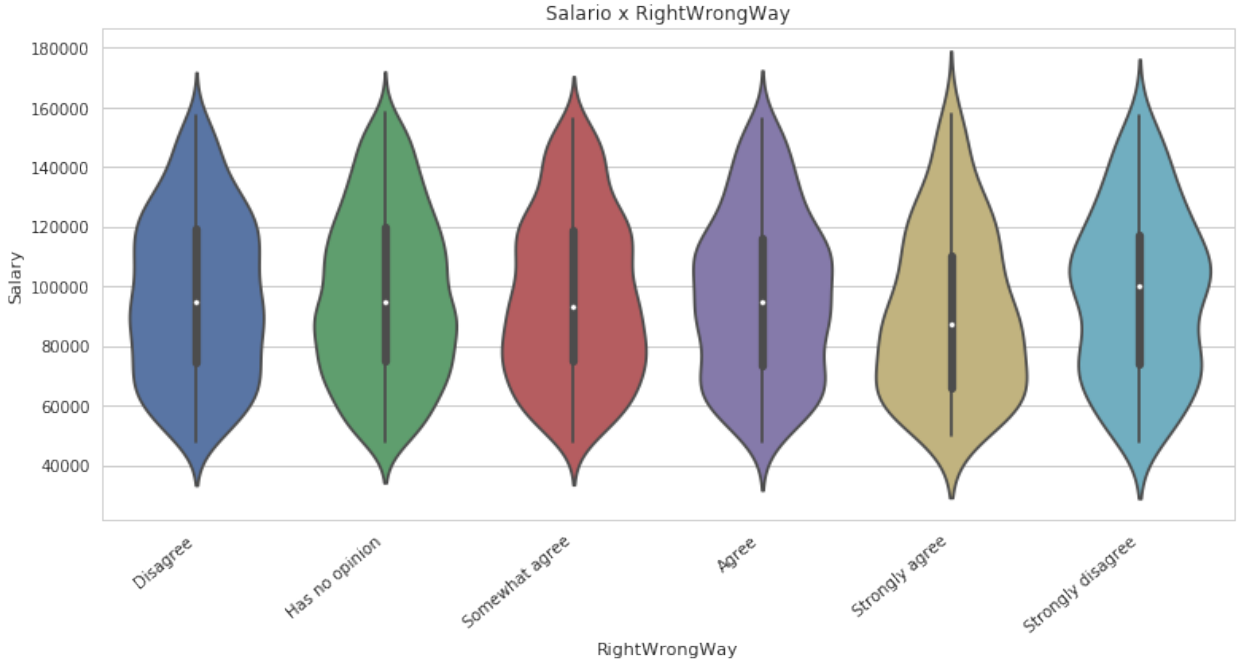

Salario x BoringDetails
Salary
180000
160000
140000
120000
100000
80000
60000
40000
Strongly disagree
Disagree
Has no opinion
Agree
Strongly agree
Somewhat agree
BoringDetails

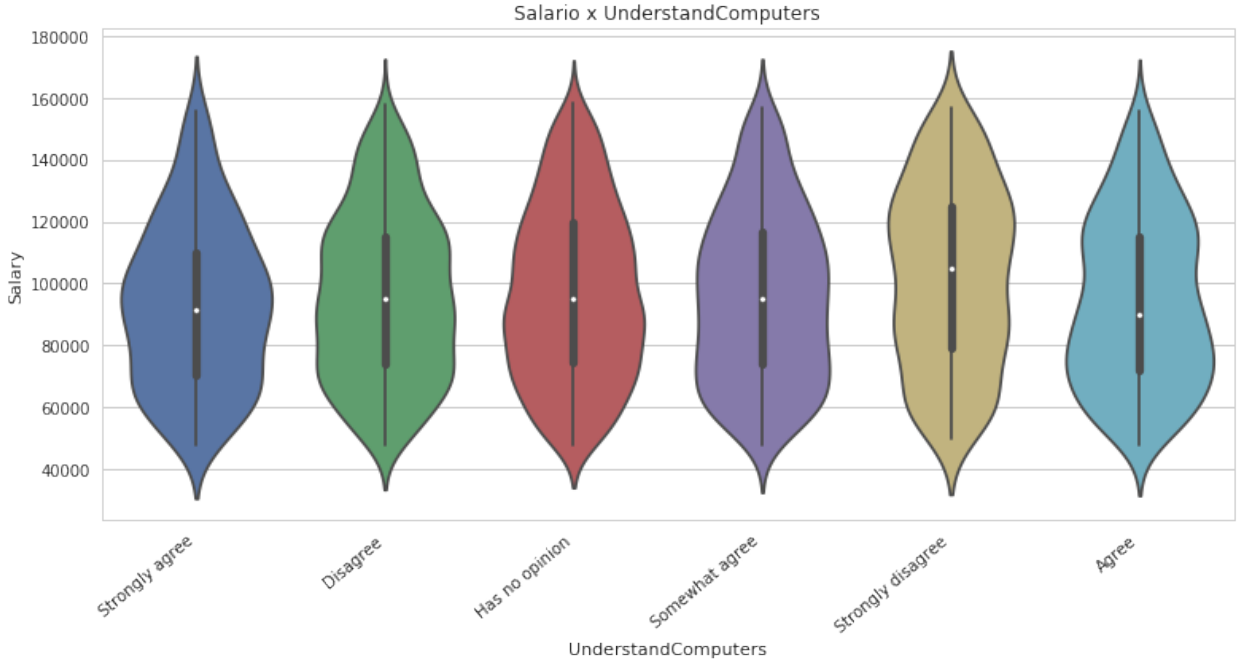

Salario x JobSecurity
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Agree
Strongly agree
Has no opinion
Disagree
Somewhat agree
Strongly disagree
JobSecurity

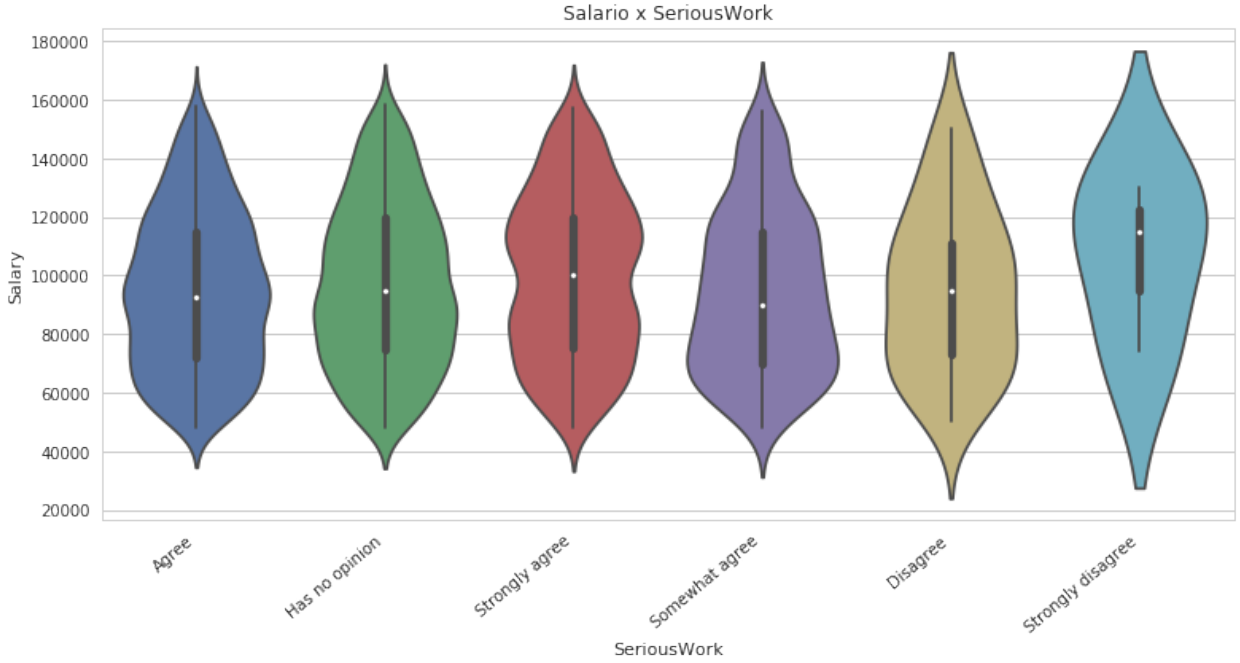

Salario x DiversityImportant
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Agree
Has no opinion
Strongly agree
Somewhat agree
Strongly disagree
Disagree
DiversityImportant

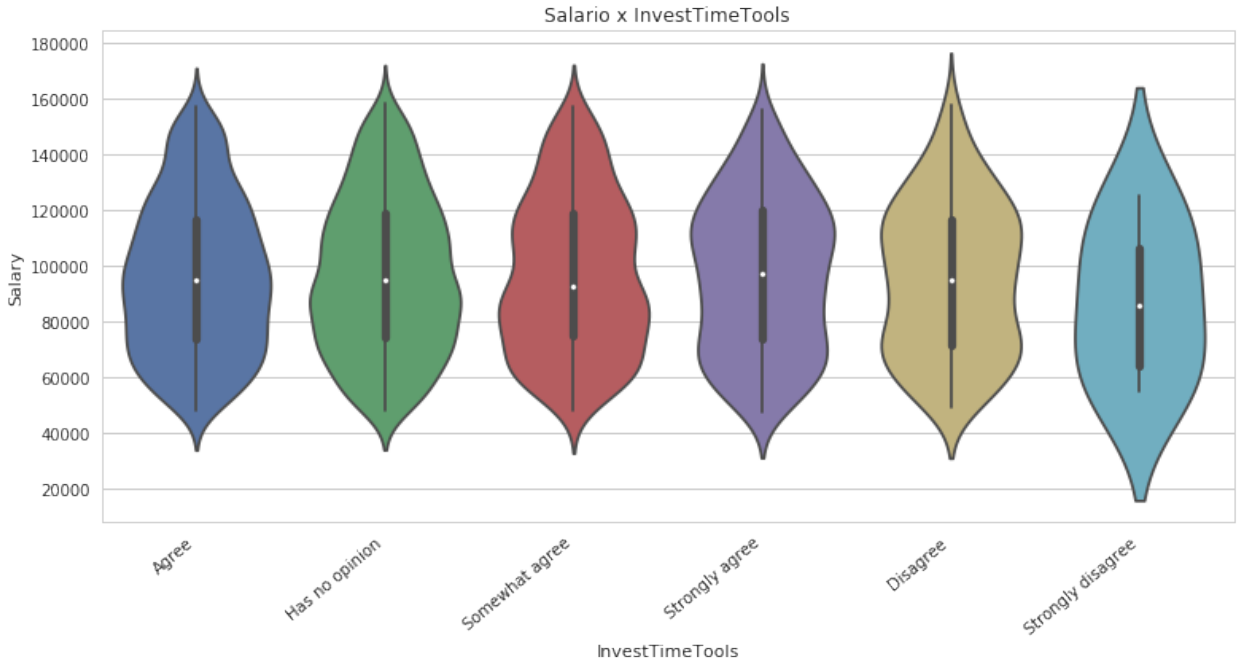

Salario x AnnoyingUI
Salary
175000
150000
125000
100000
75000
50000
25000
Agree
Somewhat agree
Has no opinion
Strongly agree
Disagree
Strongly disagree
AnnoyingUI

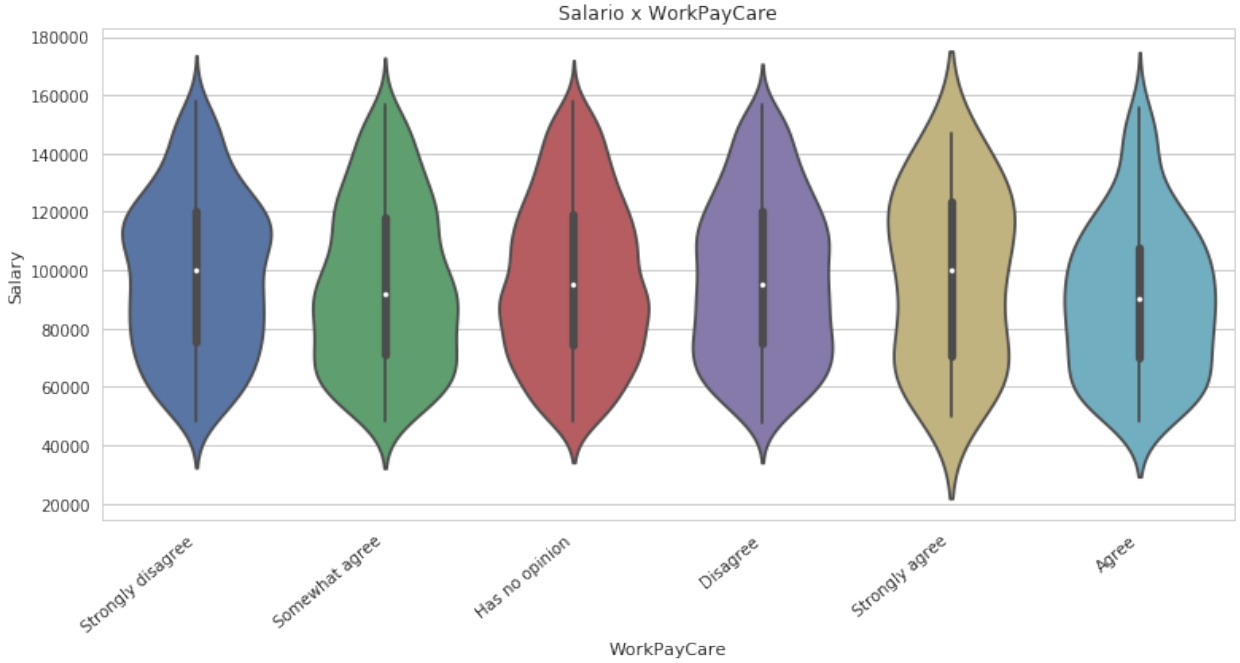

Salario x FriendsDevelopers
Salary
180000
160000
140000
120000
100000
80000
60000
40000
Disagree
Agree
Has no opinion
Somewhat agree
Strongly disagree
Strongly agree
FriendsDevelopers

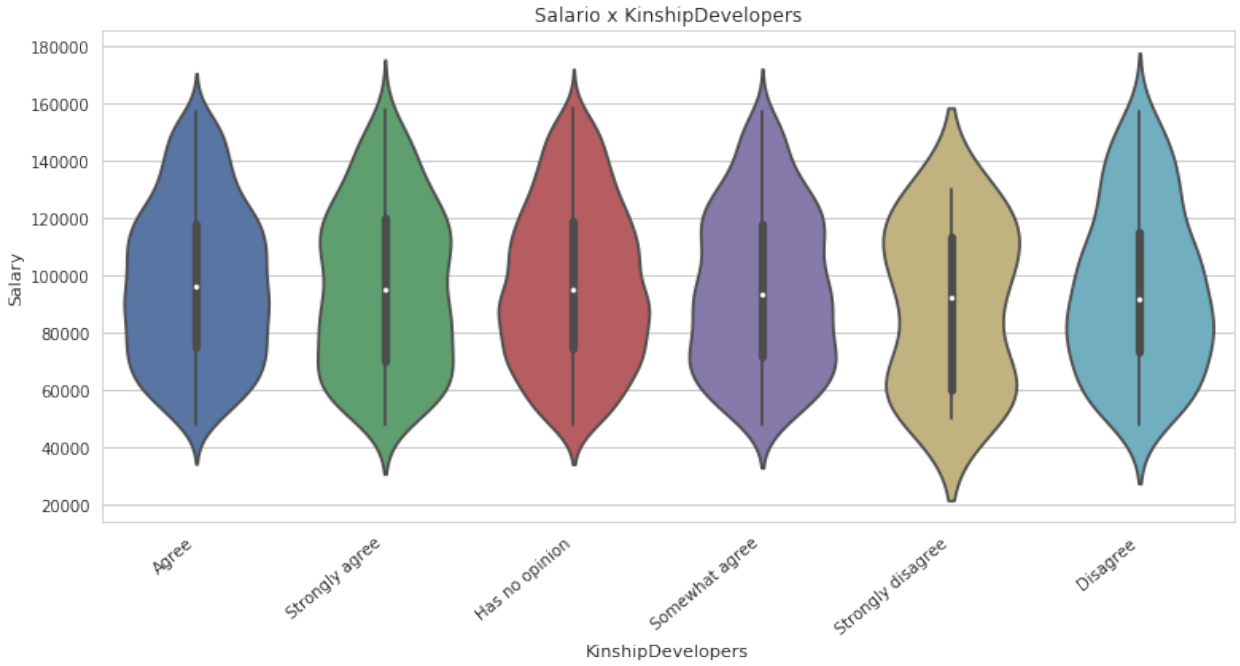

Salario x RightWrongWay
Salary
180000
160000
140000
120000
100000
80000
60000
40000
Disagree
Has no opinion
Somewhat agree
Agree
Strongly agree
Strongly disagree
RightWrongWay

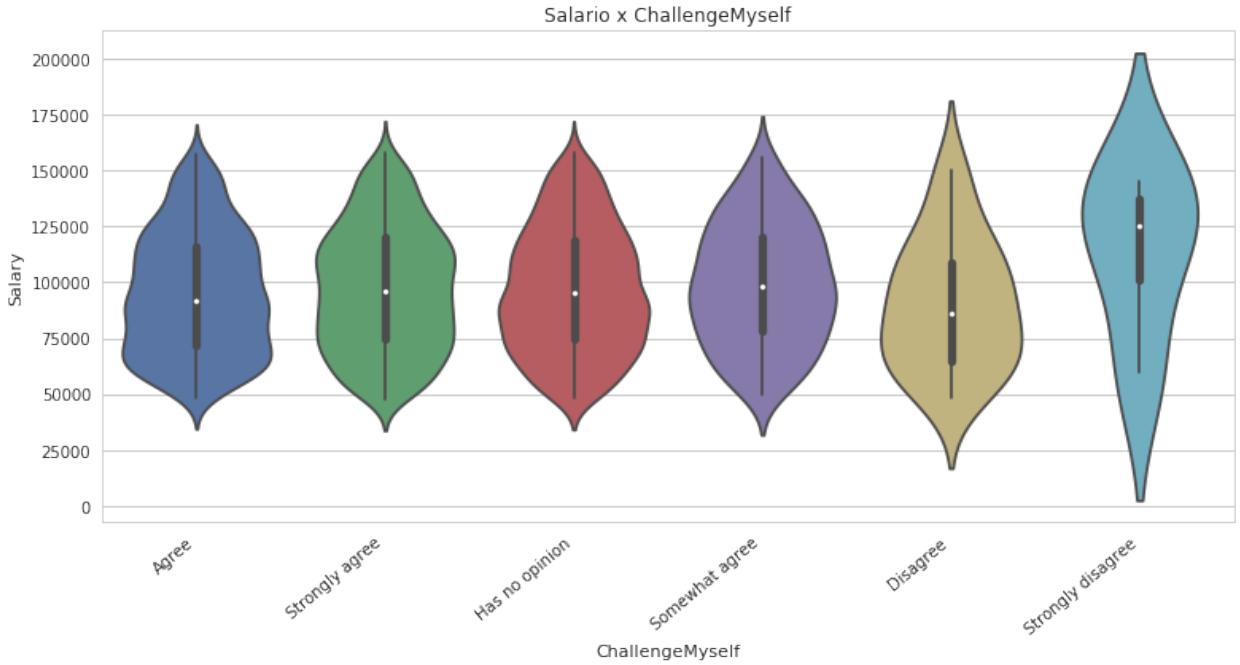

Salario x UnderstandComputers
Salary
180000
160000
140000
120000
100000
80000
60000
40000
Strongly agree
Disagree
Has no opinion
Somewhat agree
Strongly disagree
Agree
UnderstandComputers

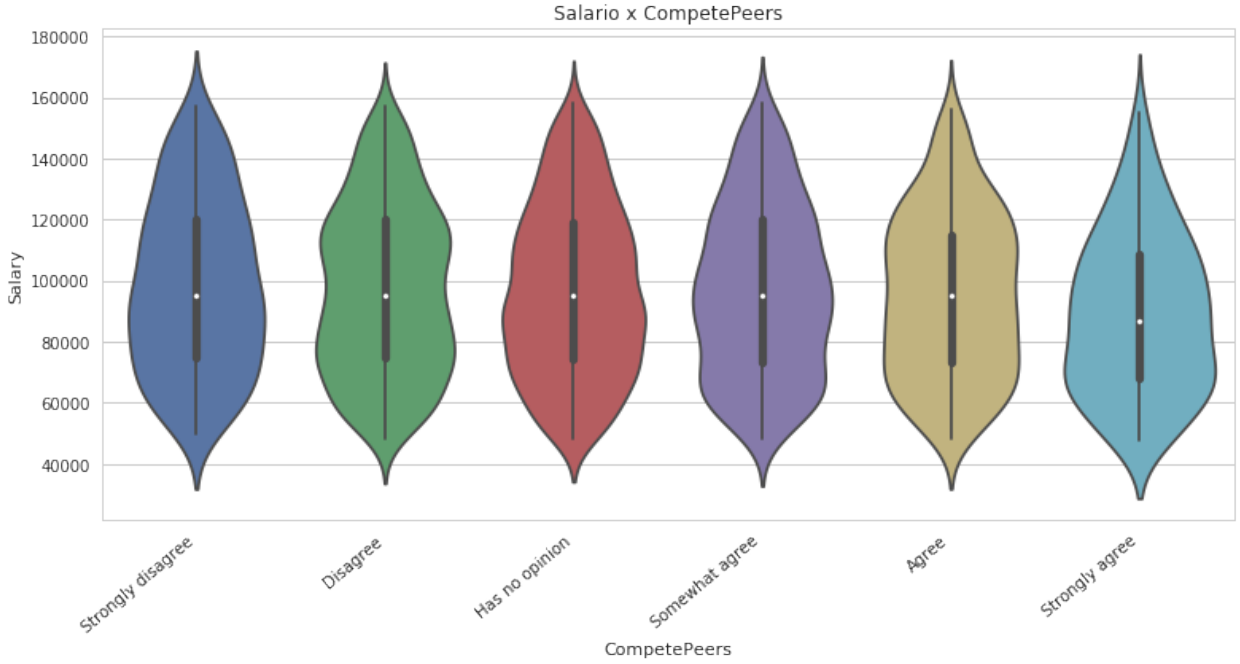

Salario x SeriousWork
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Agree
Has no opinion
Strongly agree
Somewhat agree
Disagree
Strongly disagree
SeriousWork

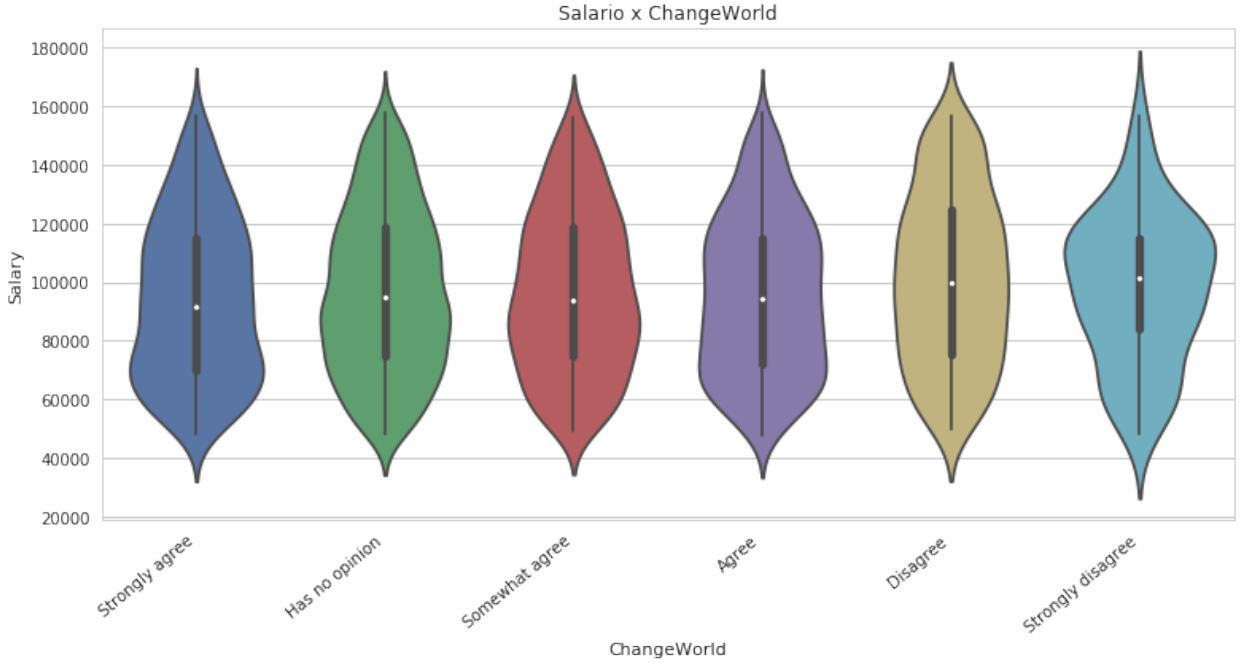

Salario x InvestTimeTools
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Agree
Has no opinion
Somewhat agree
Strongly agree
Disagree
Strongly disagree
InvestTimeTools

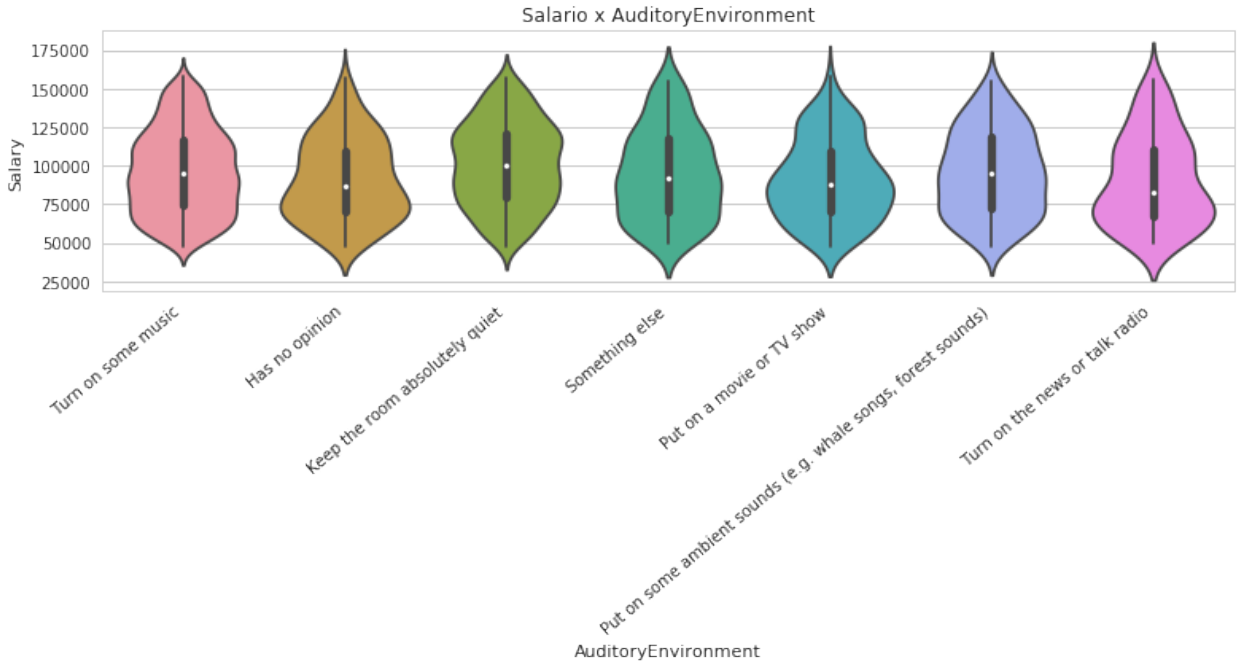

Salario x WorkPayCare
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Strongly disagree
Somewhat agree
Has no opinion
Disagree
Strongly agree
Agree
WorkPayCare

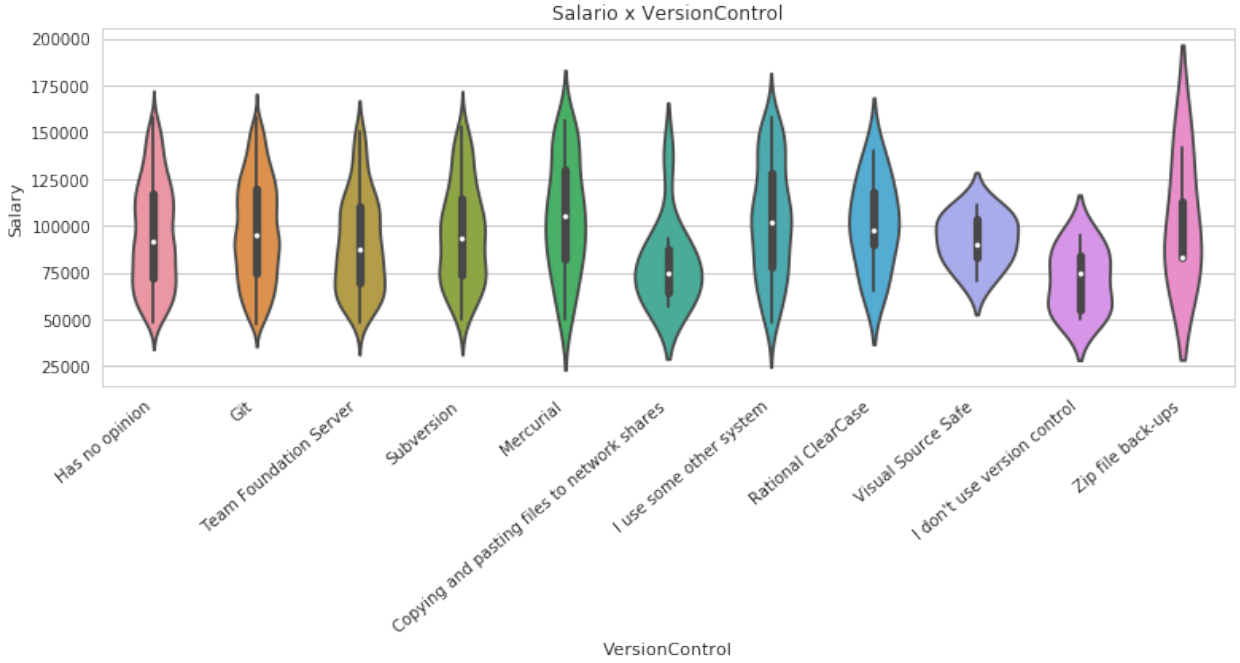

Salario x KinshipDevelopers
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Agree
Strongly agree
Has no opinion
Somewhat agree
Strongly disagree
Disagree
KinshipDevelopers

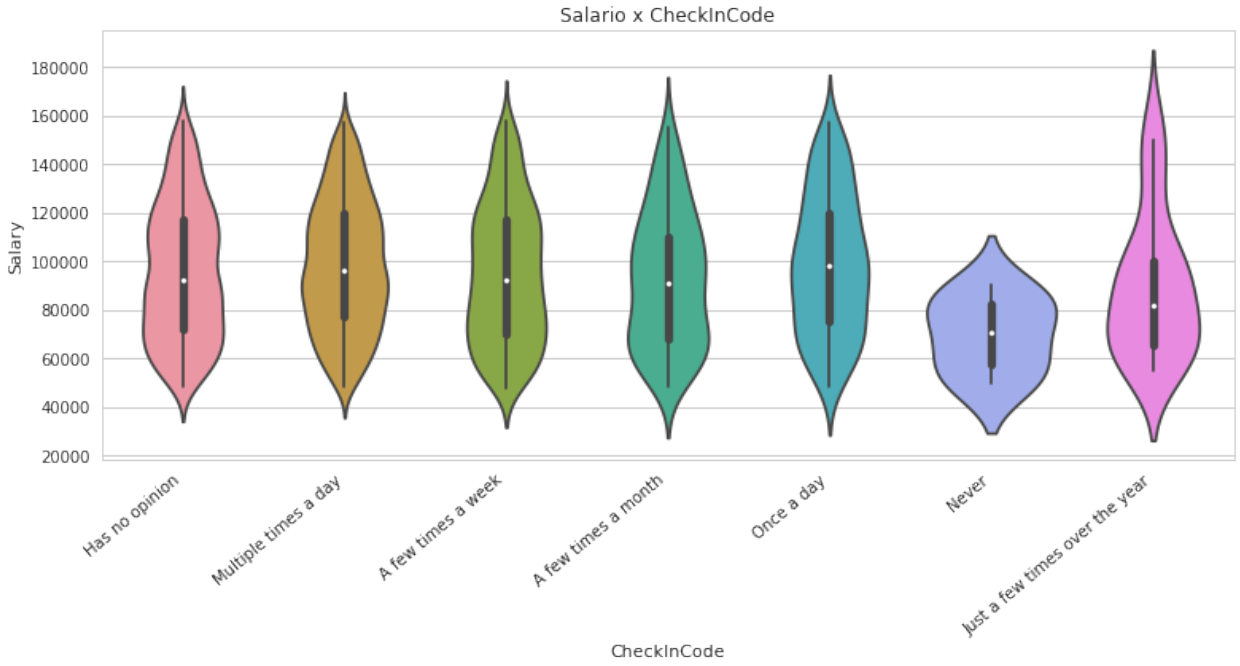

Salario x ChallengeMyself
Salary
200000
175000
150000
125000
100000
75000
50000
25000
0
Agree
Strongly agree
Has no opinion
Somewhat agree
Disagree
Strongly disagree
ChallengeMyself

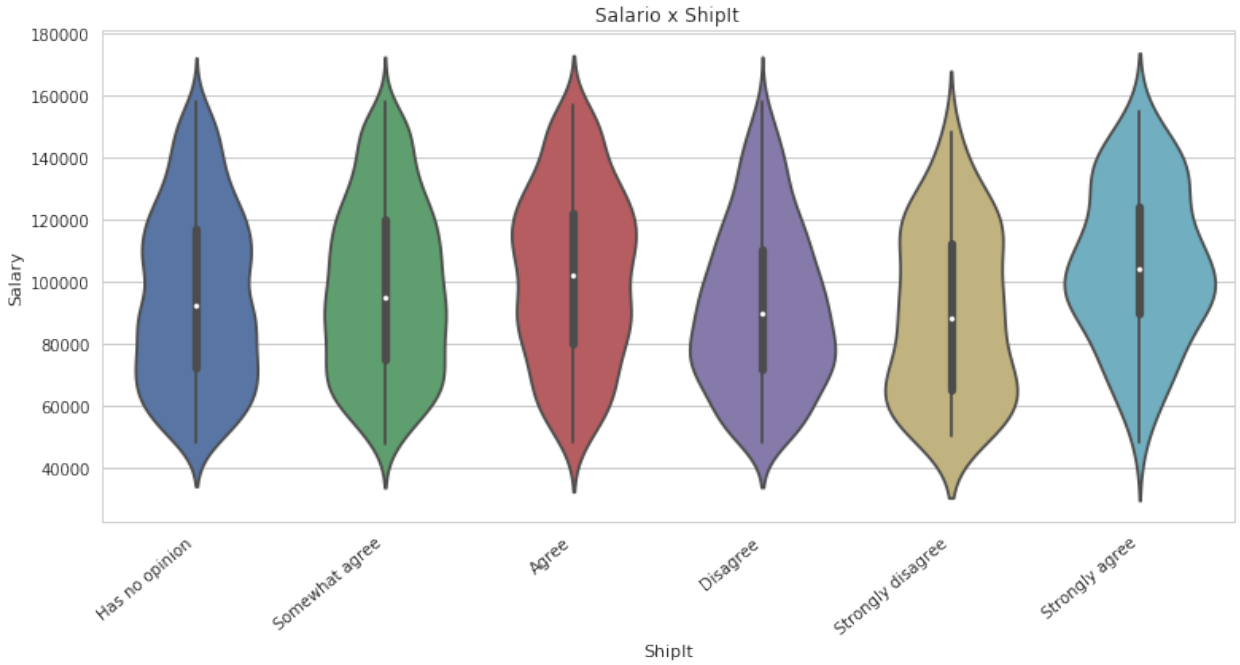

Salario x CompetePeers
Salary
180000
160000
140000
120000
100000
80000
60000
40000
Strongly disagree
Disagree
Has no opinion
Somewhat agree
Agree
Strongly agree
CompetePeers

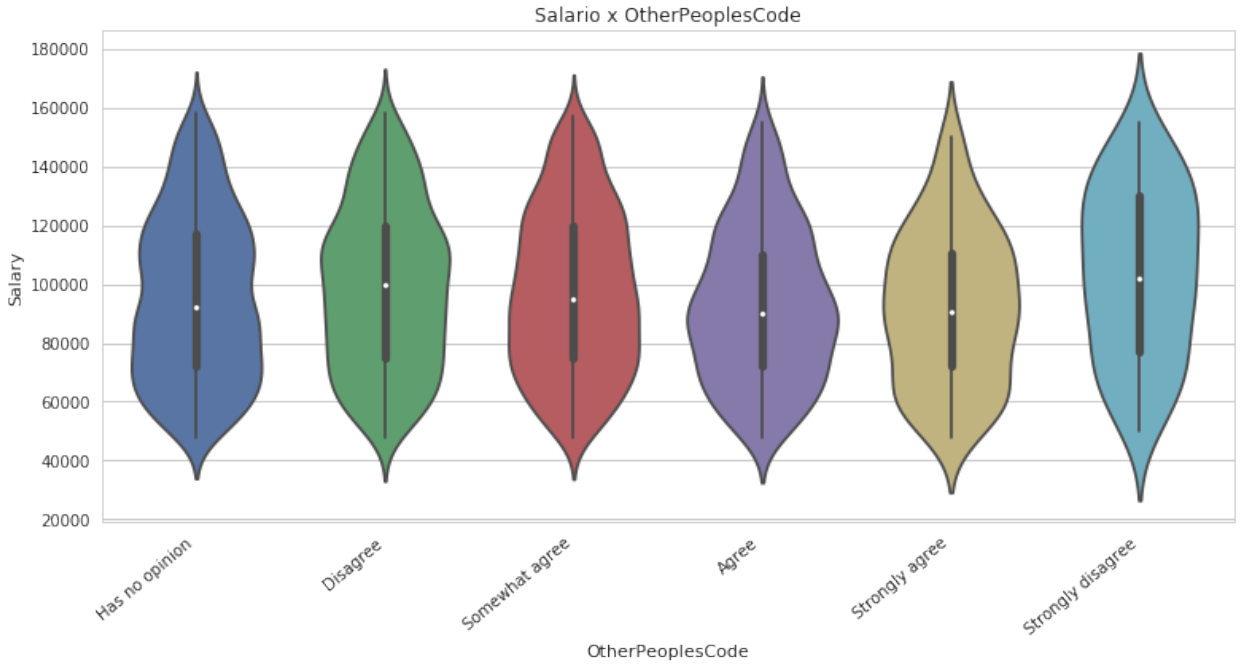

Salario x ChangeWorld
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Strongly agree
Has no opinion
Somewhat agree
Agree
Disagree
Strongly disagree
ChangeWorld

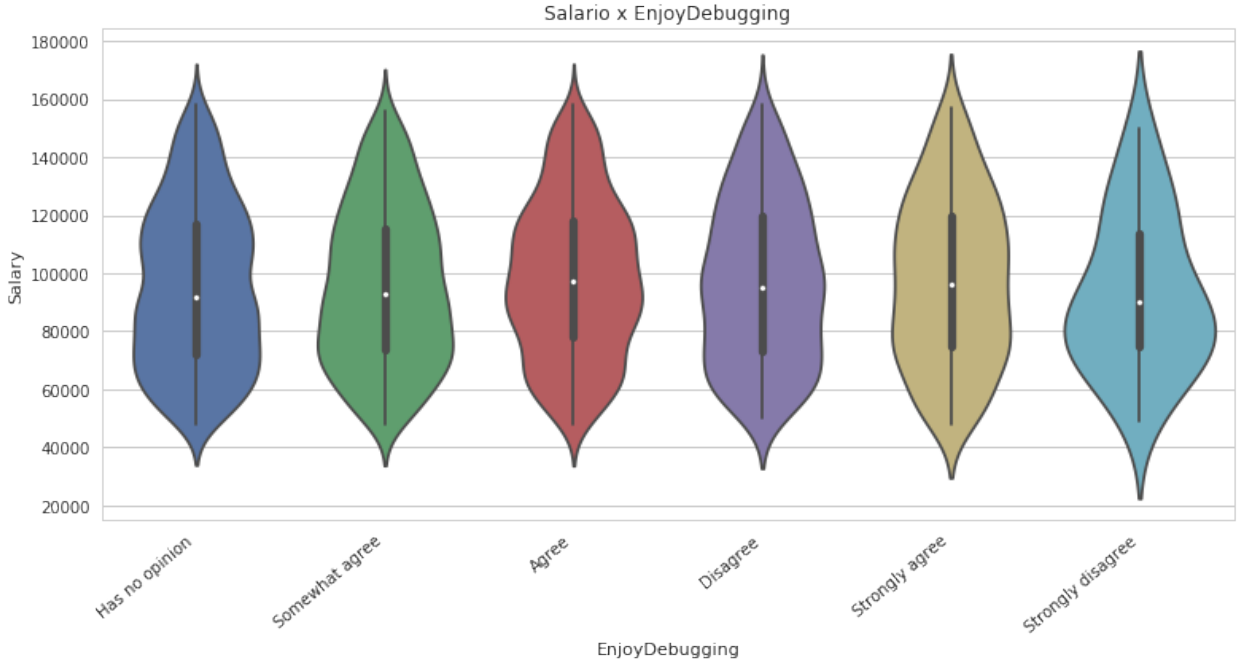

Salario x AuditoryEnvironment
Salary
175000
150000
125000
100000
75000
50000
25000
Turn on some music
Has no opinion
Keep the room absolutely quiet
Something else
Put on a movie or TV show
Put on some ambient sounds (e.g. whale songs, forest sounds)
Turn on the news or talk radio
AuditoryEnvironment

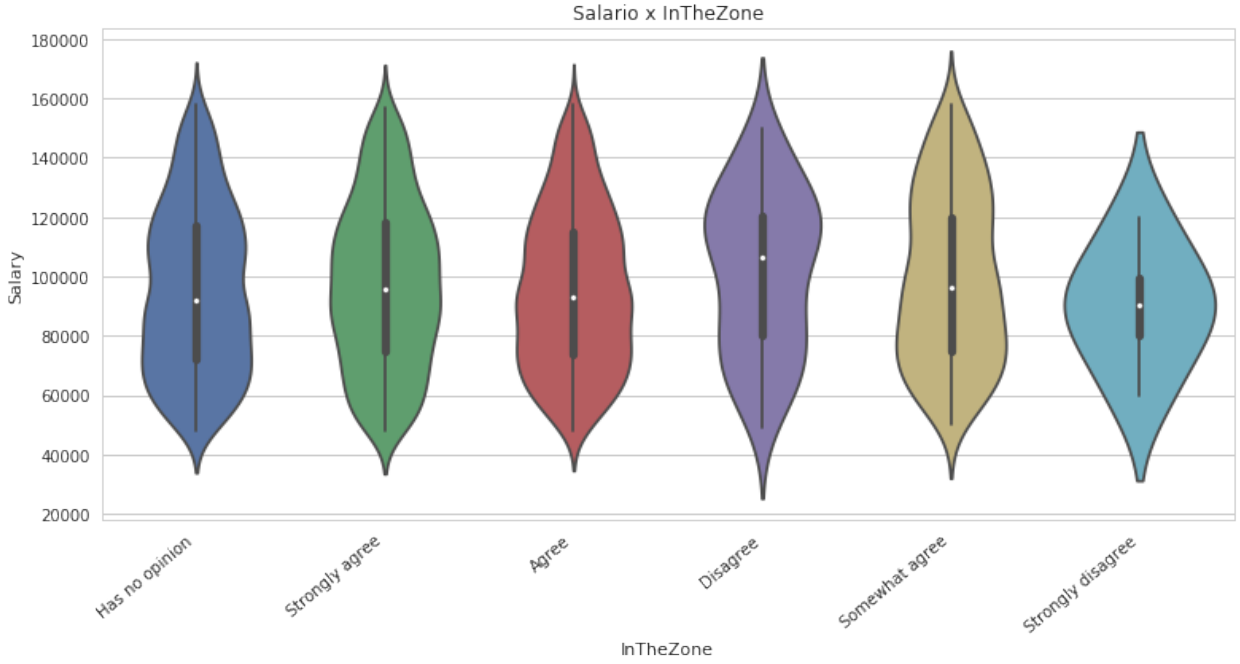

Salario x VersionControl
Salary
200000
175000
150000
125000
100000
75000
50000
25000
Has no opinion
Git
Team Foundation Server
Subversion
Mercurial
Copying and pasting files to network shares
I use some other system
Rational ClearCase
Visual Source Safe
I don't use version control
Zip file back-ups
VersionControl

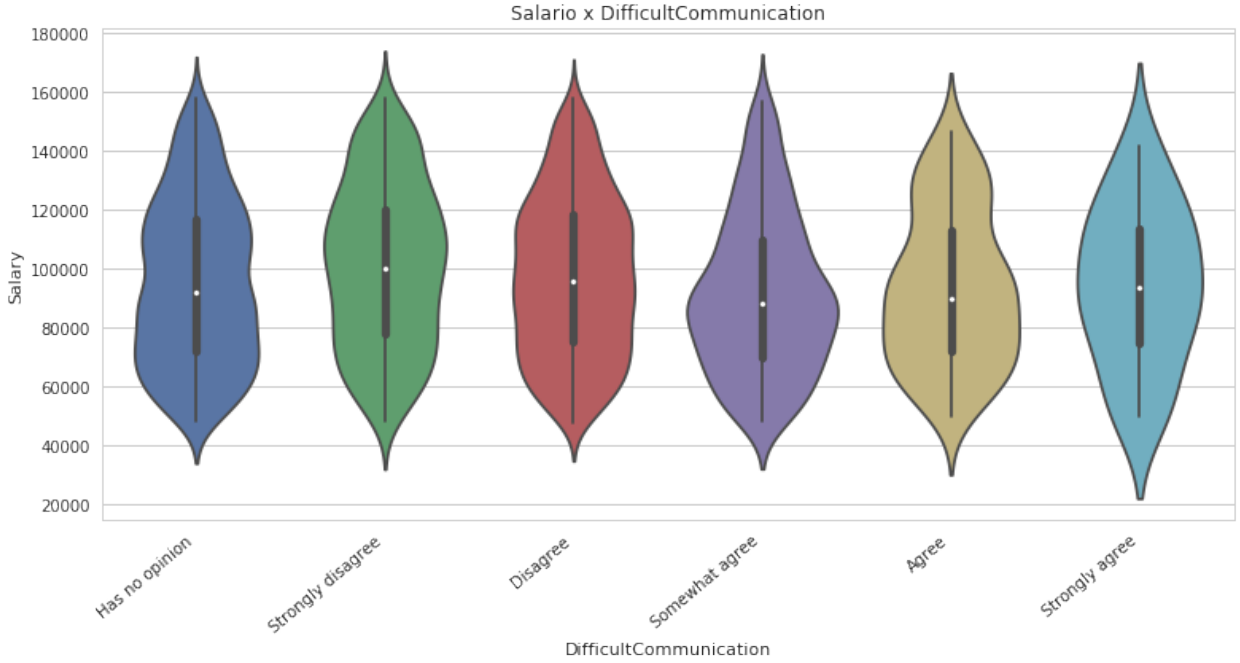

Salario x CheckInCode
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Has no opinion
Multiple times a day
A few times a week
A few times a month
Once a day
Never
Just a few times over the year
CheckInCode

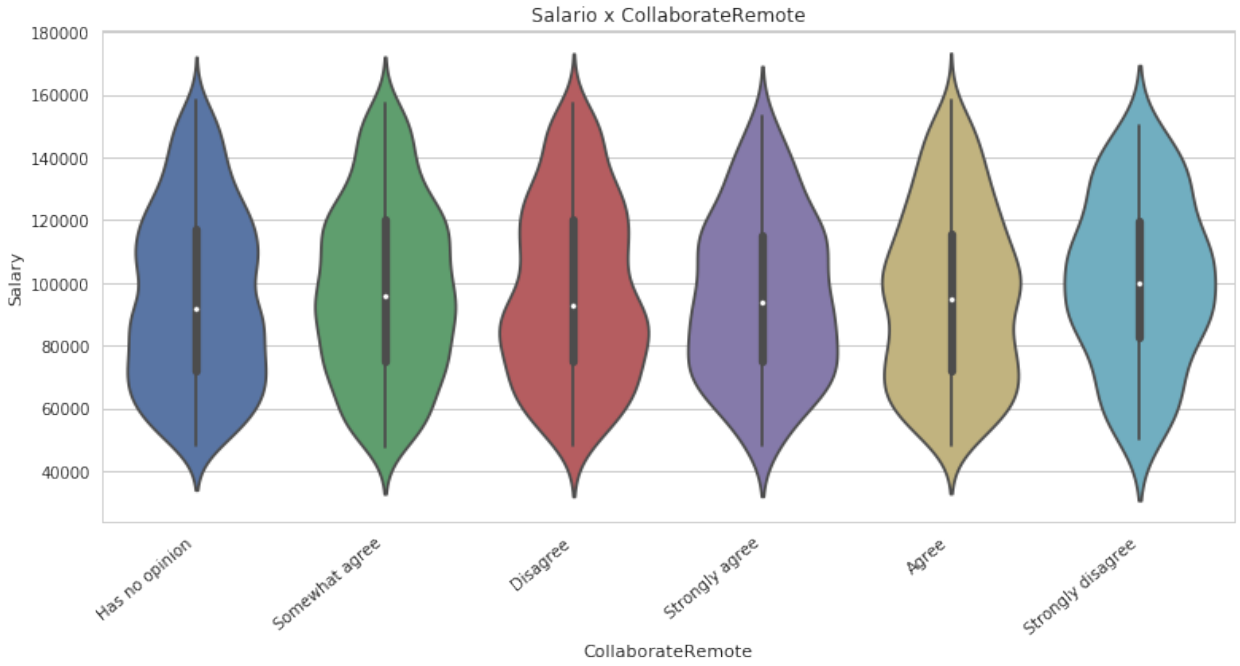

Salario x ShipIt
Salary
180000
160000
140000
120000
100000
80000
60000
40000
Has no opinion
Somewhat agree
Agree
Disagree
Strongly disagree
Strongly agree
ShipIt

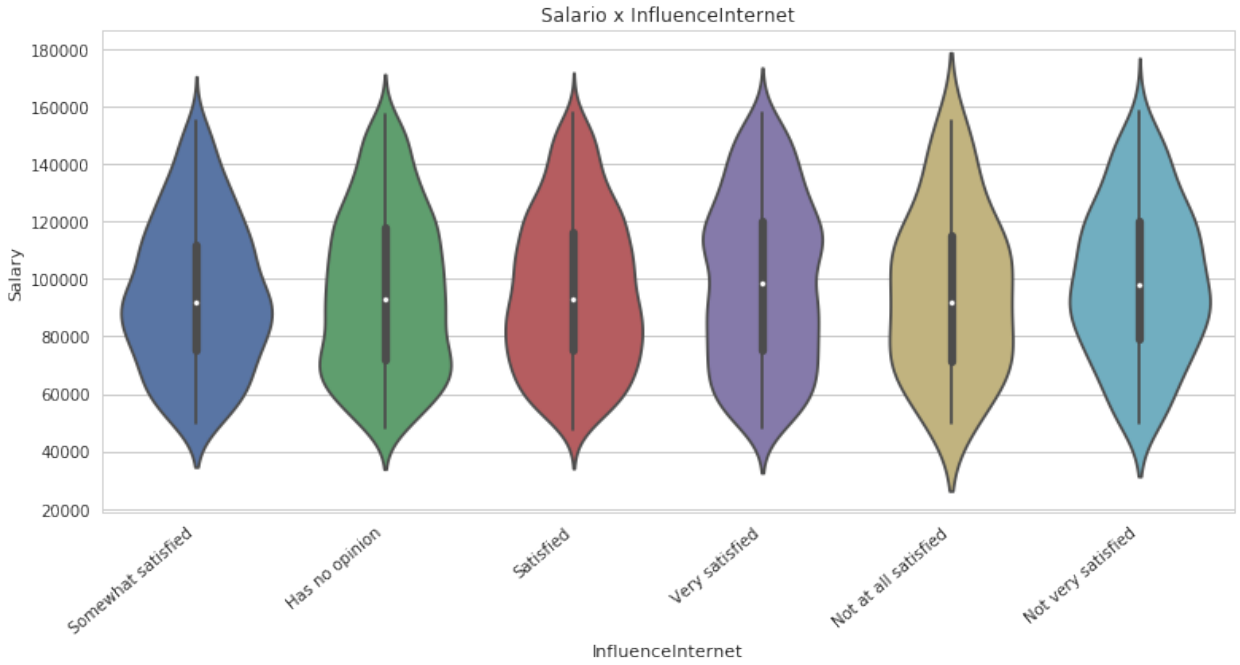

Salario x OtherPeoplesCode
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Has no opinion
Disagree
Somewhat agree
Agree
Strongly agree
Strongly disagree
OtherPeoplesCode

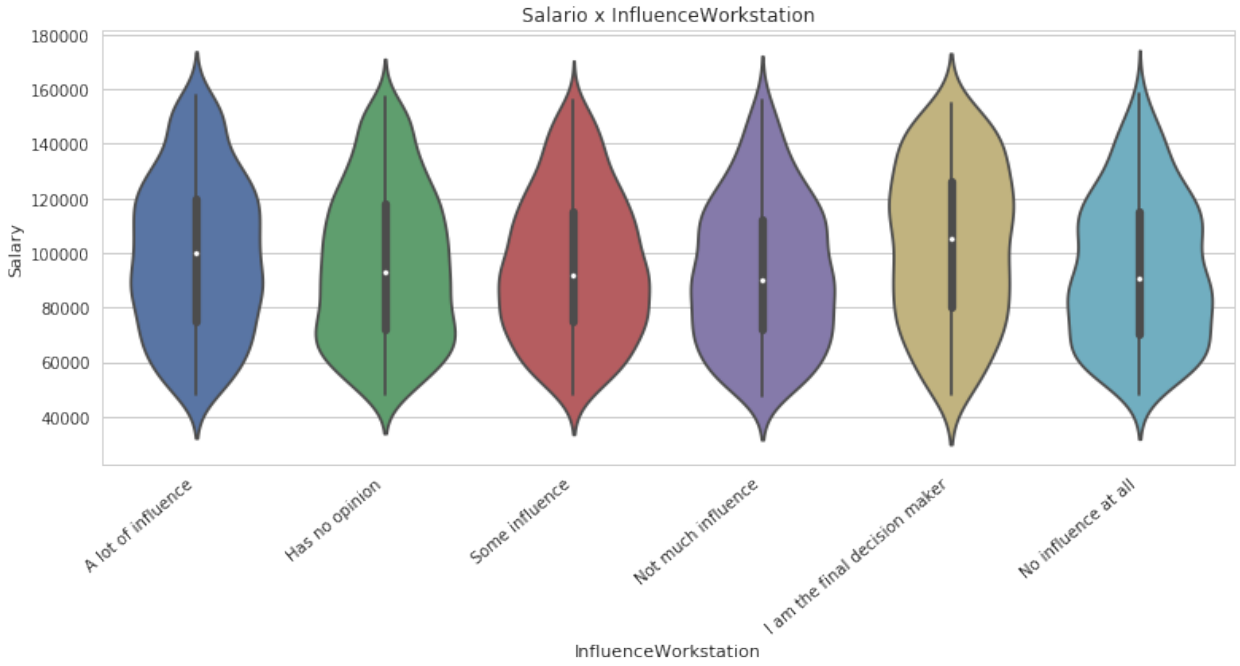

Salario x EnjoyDebugging
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Has no opinion
Somewhat agree
Agree
Disagree
Strongly agree
Strongly disagree
EnjoyDebugging

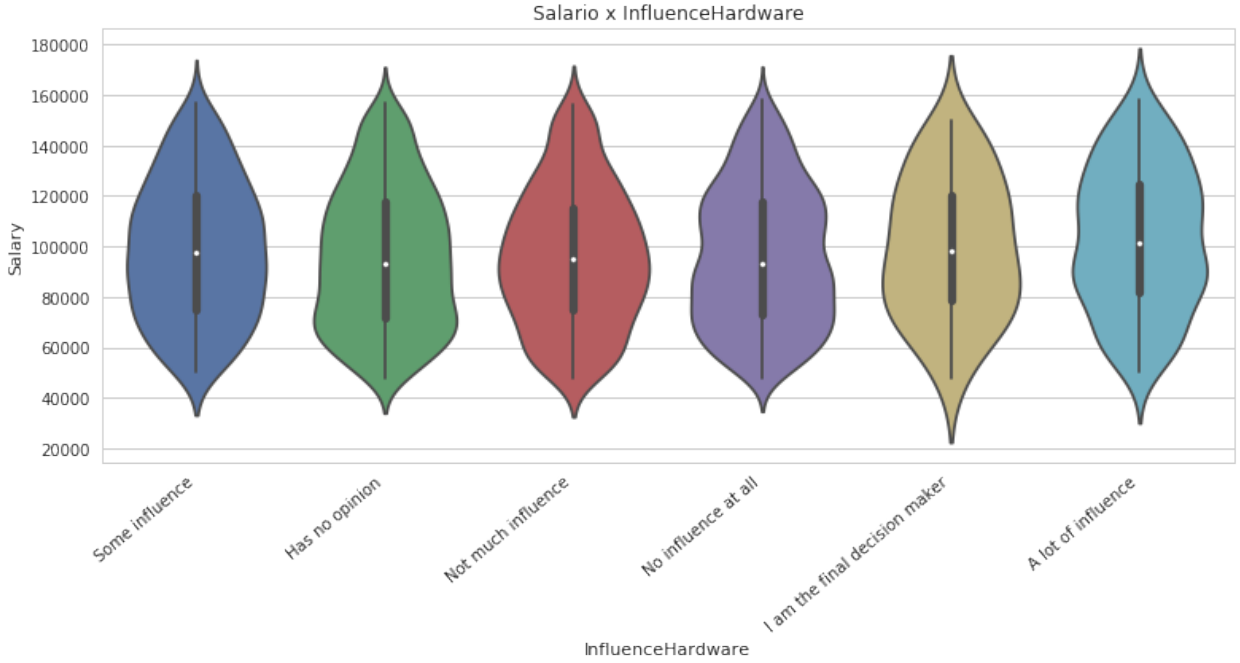

Salario x InTheZone
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Has no opinion
Strongly agree
Agree
Disagree
Somewhat agree
Strongly disagree
InTheZone

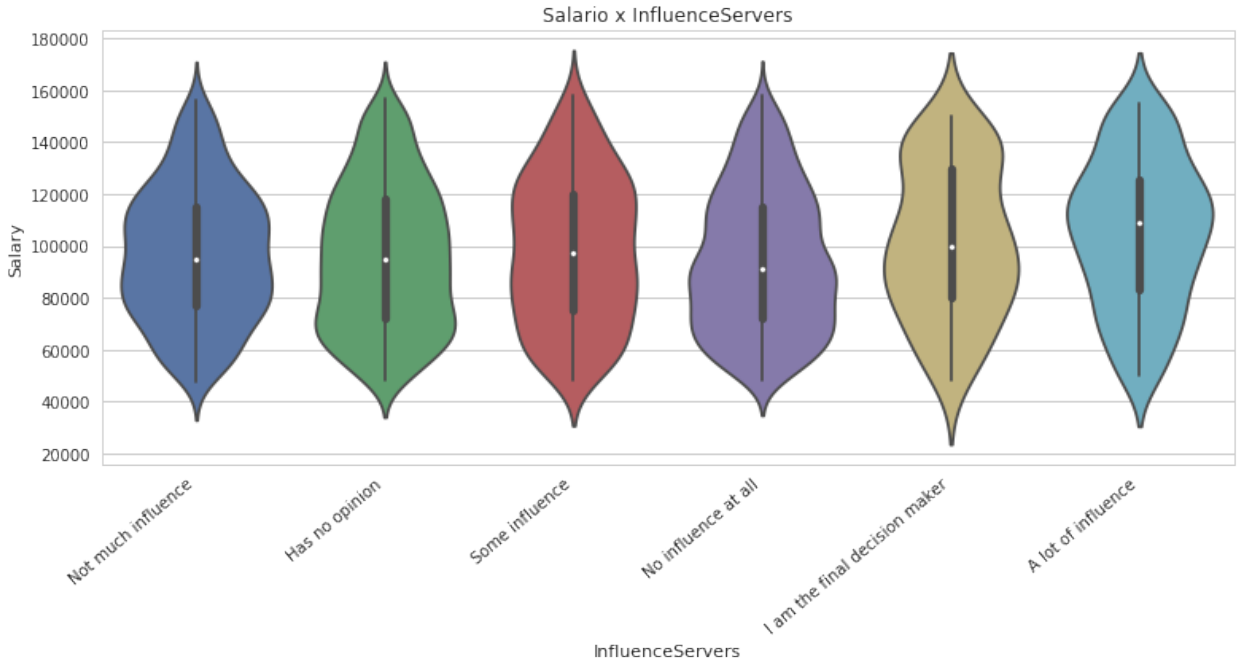

Salario x DifficultCommunication
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Has no opinion
Strongly disagree
Disagree
Somewhat agree
Agree
Strongly agree
DifficultCommunication

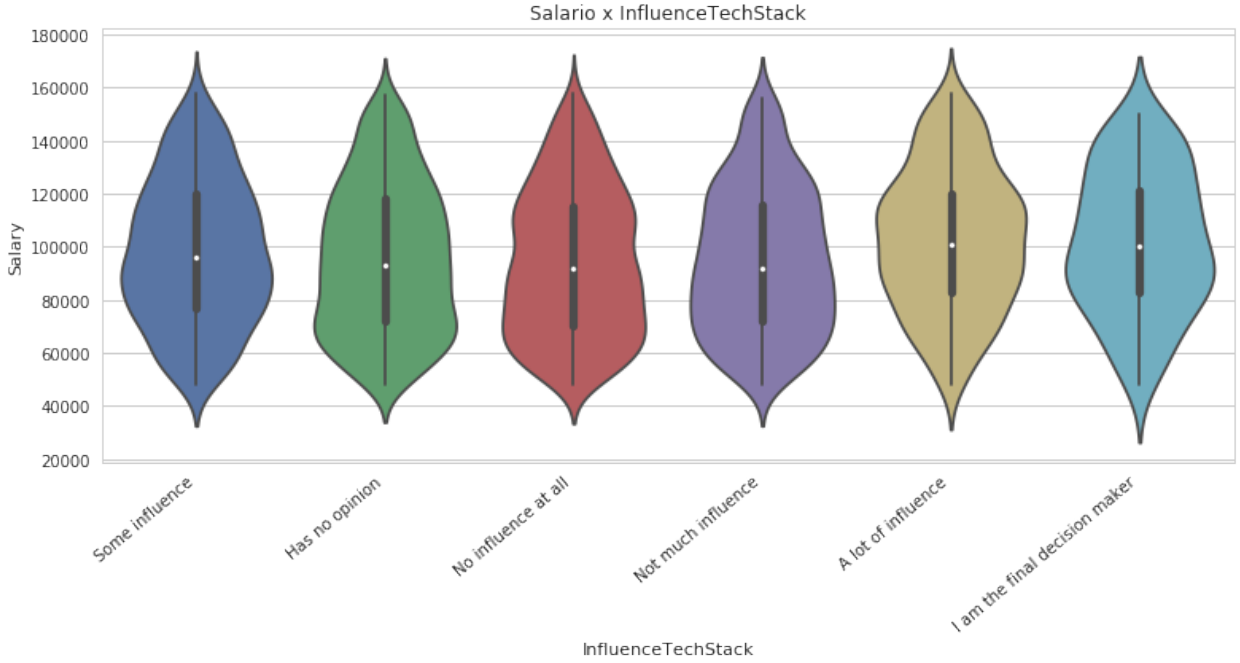

Salario x CollaborateRemote
Salary
180000
160000
140000
120000
100000
80000
60000
40000
Has no opinion
Somewhat agree
Disagree
Strongly agree
Agree
Strongly disagree
CollaborateRemote

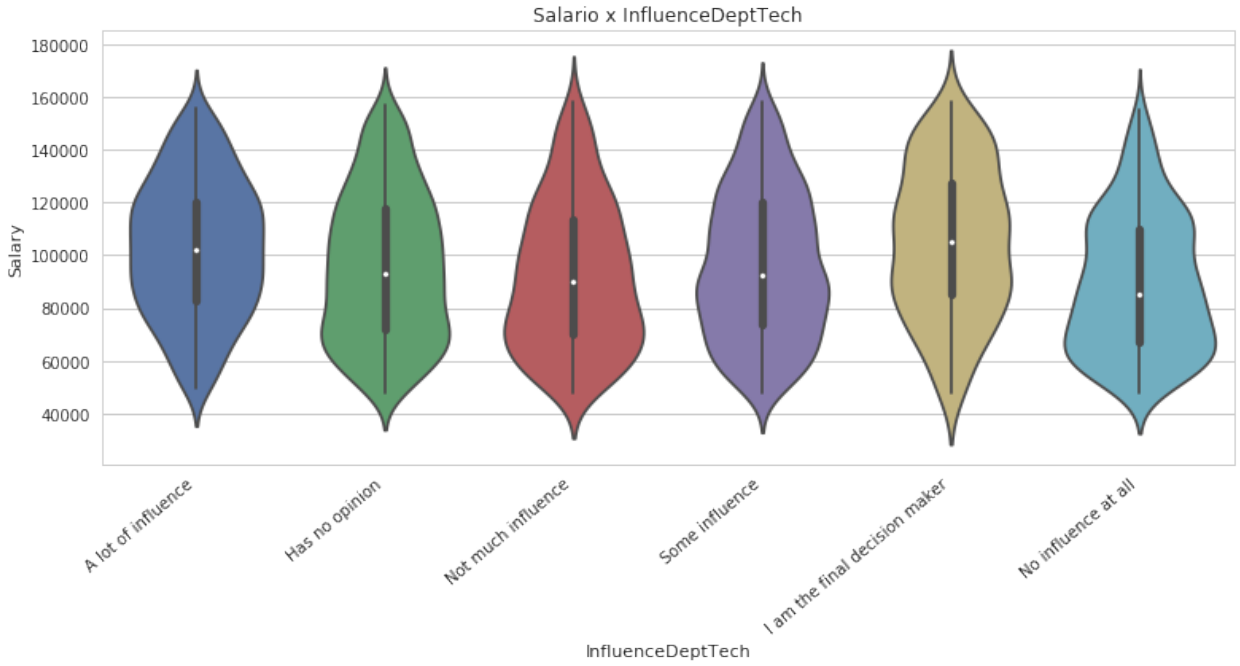

Salario x InfluenceInternet
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Somewhat satisfied
Has no opinion
Satisfied
Very satisfied
Not at all satisfied
Not very satisfied
InfluenceInternet

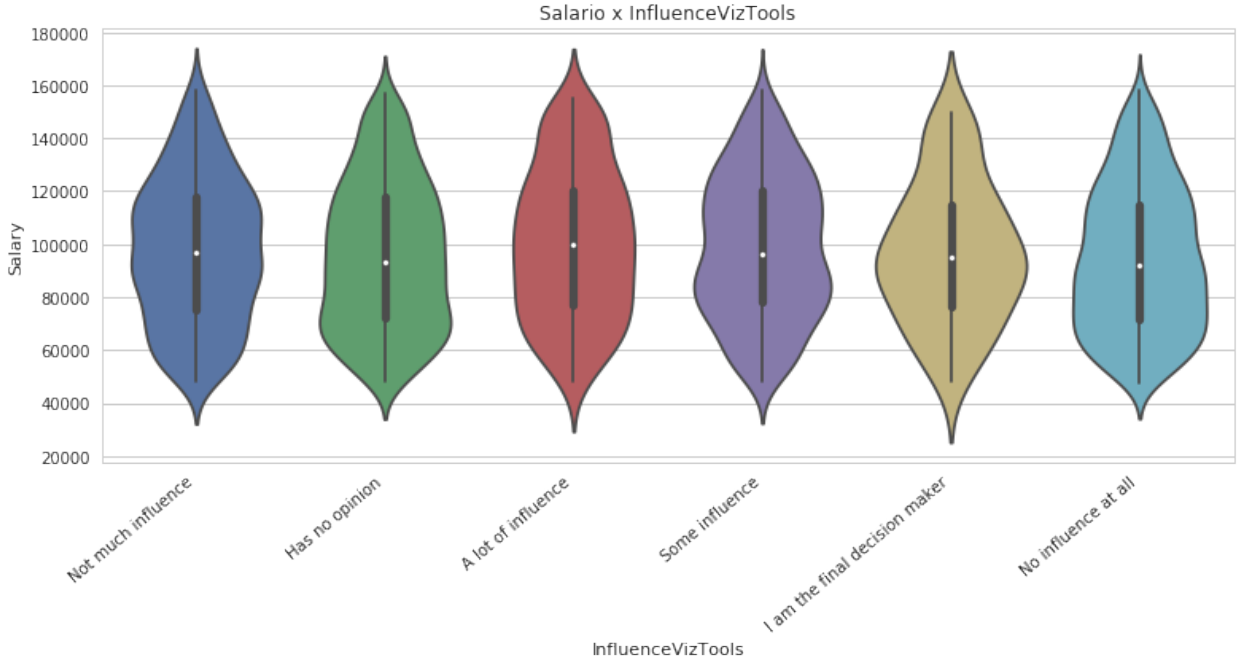

Salario x InfluenceWorkstation
Salary
180000
160000
140000
120000
100000
80000
60000
40000
A lot of influence
Has no opinion
Some influence
Not much influence
I am the final decision maker
No influence at all
InfluenceWorkstation

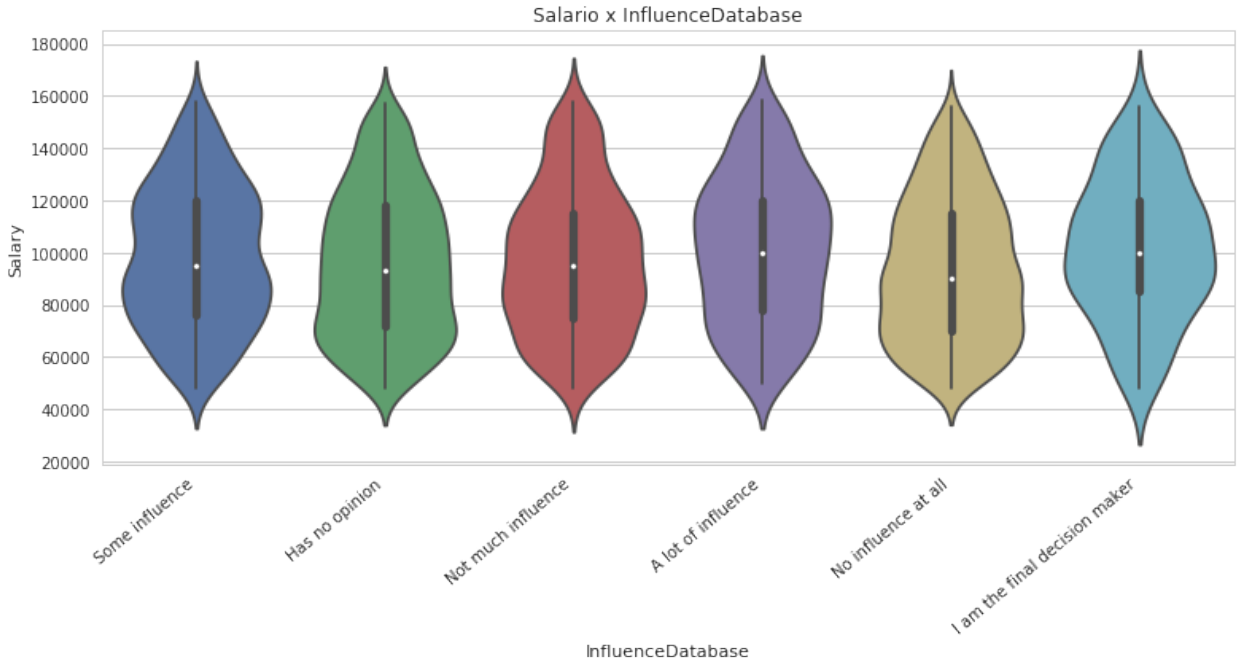

Salario x InfluenceHardware
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Some influence
Has no opinion
Not much influence
No influence at all
I am the final decision maker
A lot of influence
InfluenceHardware

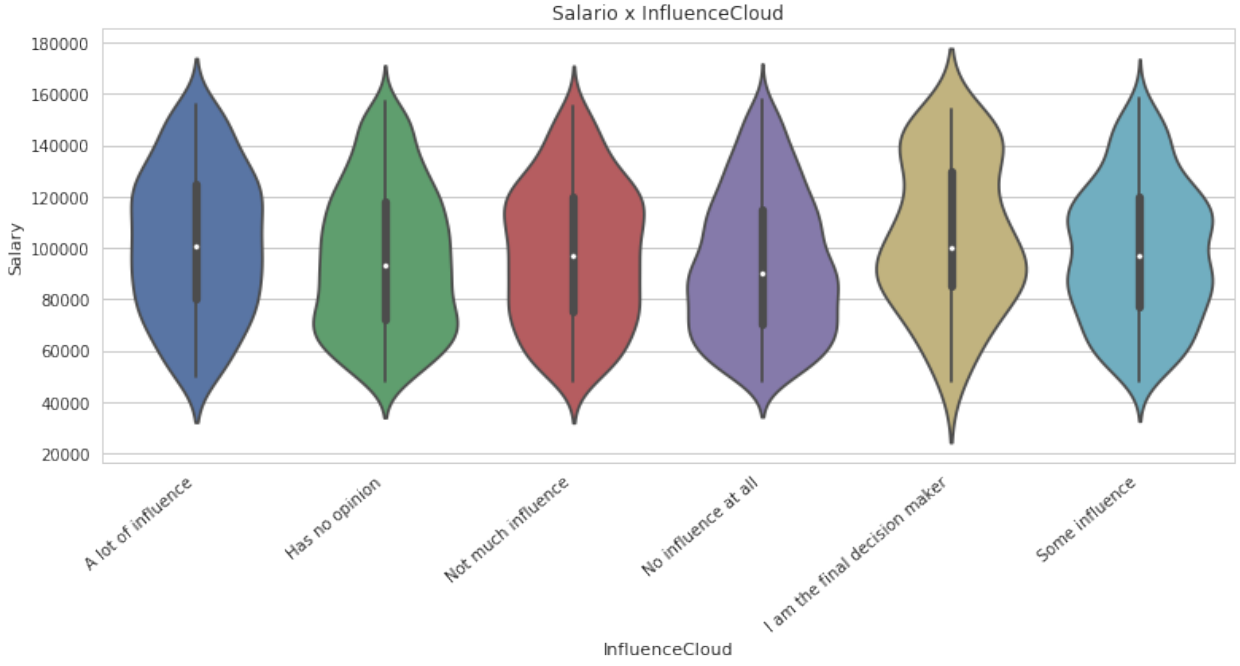

Salario x InfluenceServers
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Not much influence
Has no opinion
Some influence
No influence at all
I am the final decision maker
A lot of influence
InfluenceServers

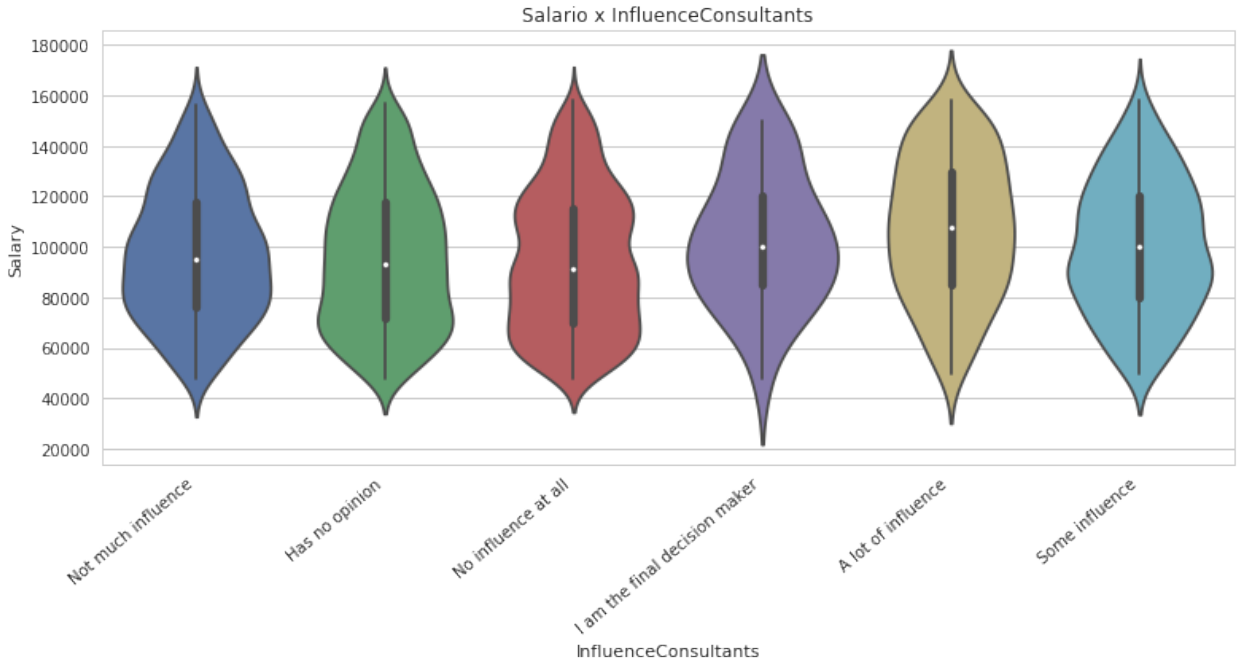

Salario x InfluenceTechStack
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Some influence
Has no opinion
No influence at all
Not much influence
A lot of influence
I am the final decision maker
InfluenceTechStack

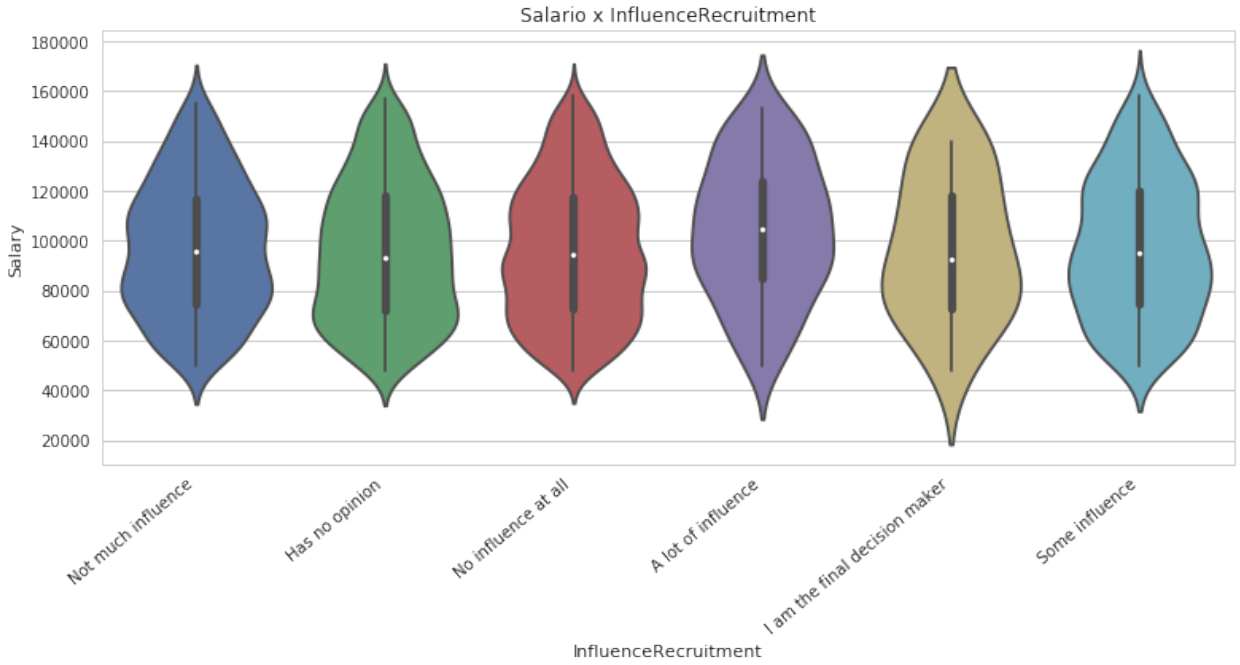

Salario x InfluenceDeptTech
Salary
180000
160000
140000
120000
100000
80000
60000
40000
A lot of influence
Has no opinion
Not much influence
Some influence
I am the final decision maker
No influence at all
InfluenceDeptTech

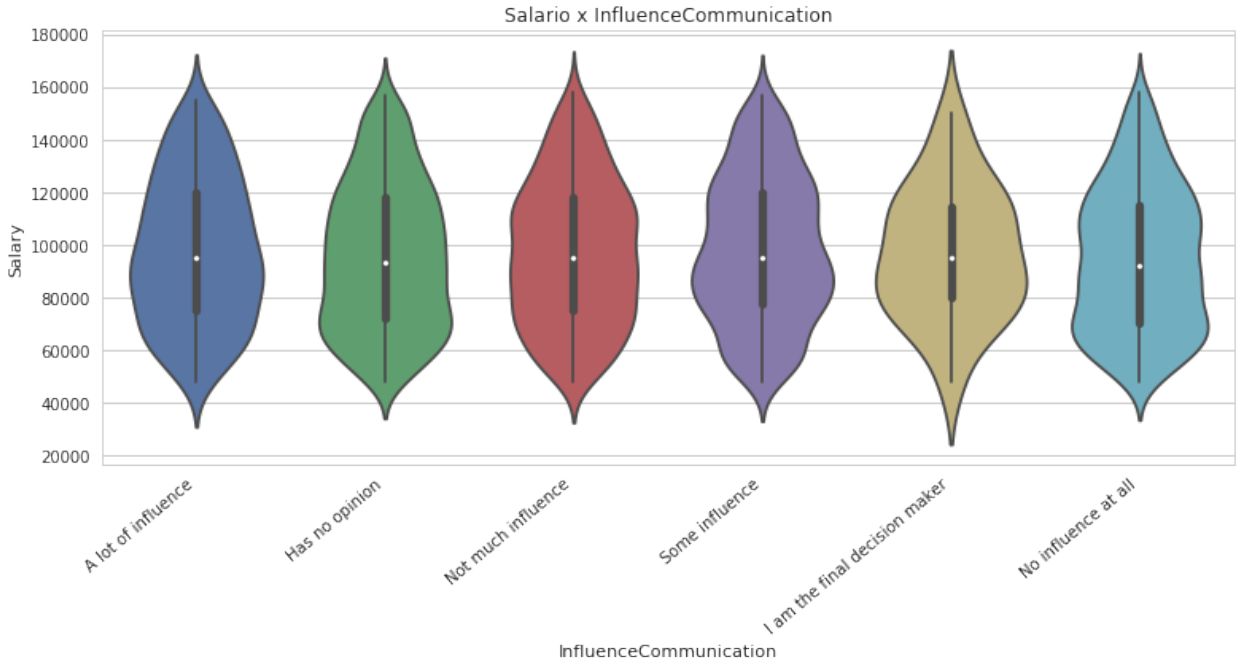

Salario x InfluenceVizTools
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Not much influence
Has no opinion
A lot of influence
Some influence
I am the final decision maker
No influence at all
InfluenceVizTools

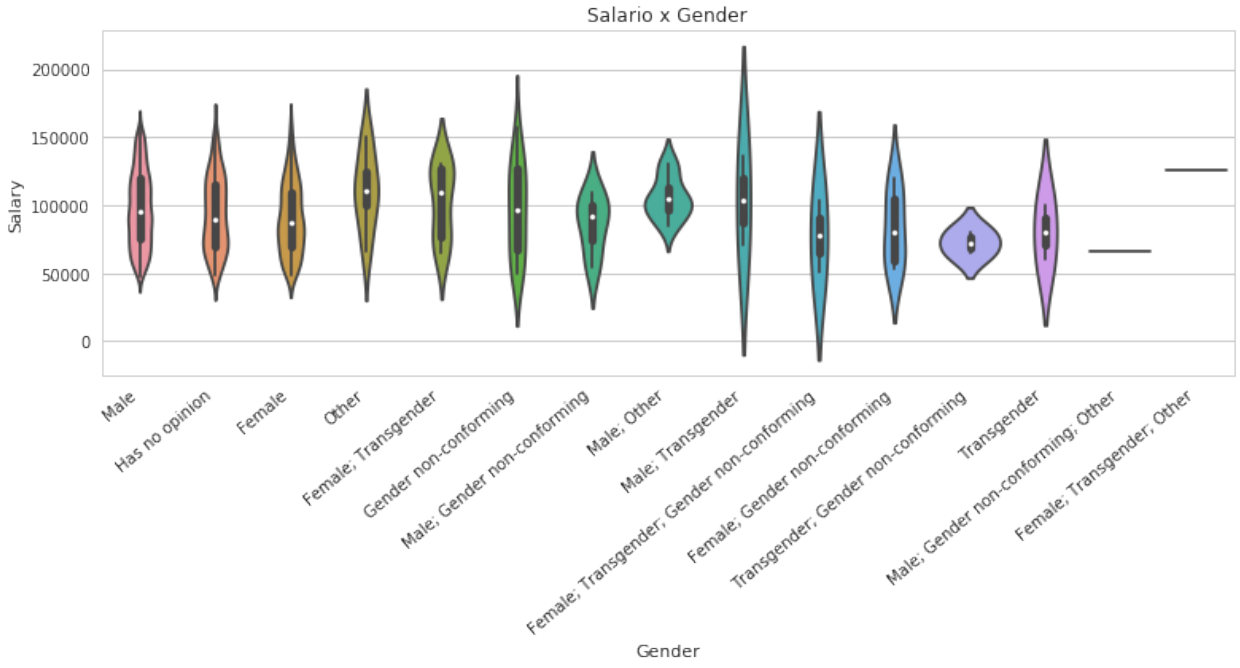

Salario x InfluenceDatabase
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Some influence
Has no opinion
Not much influence
A lot of influence
No influence at all
I am the final decision maker
InfluenceDatabase

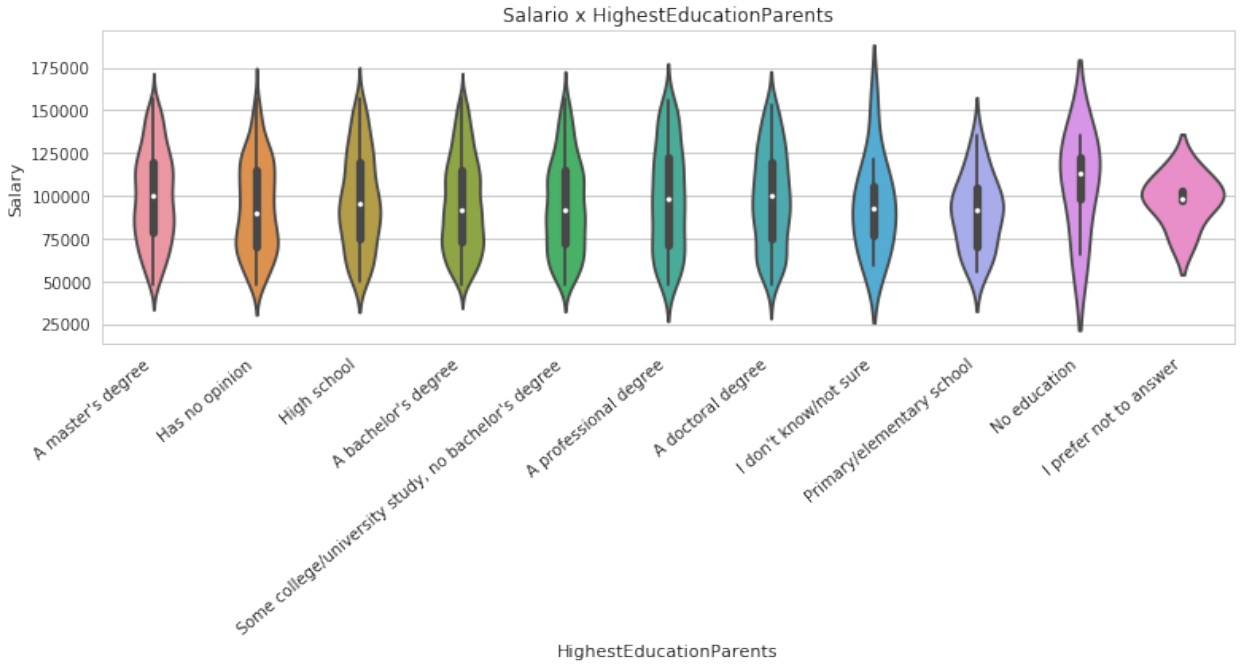

Salario x InfluenceCloud
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
A lot of influence
Has no opinion
Not much influence
No influence at all
I am the final decision maker
Some influence
InfluenceCloud

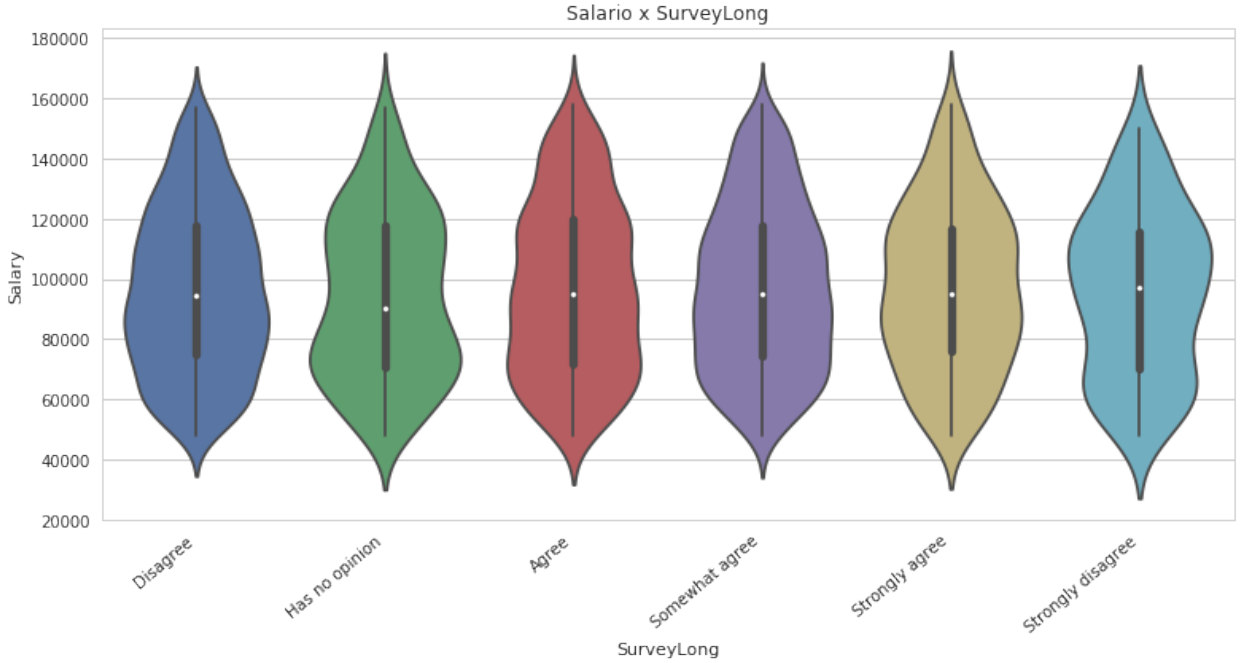

Salario x InfluenceConsultants
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Not much influence
Has no opinion
No influence at all
I am the final decision maker
A lot of influence
Some influence
InfluenceConsultants

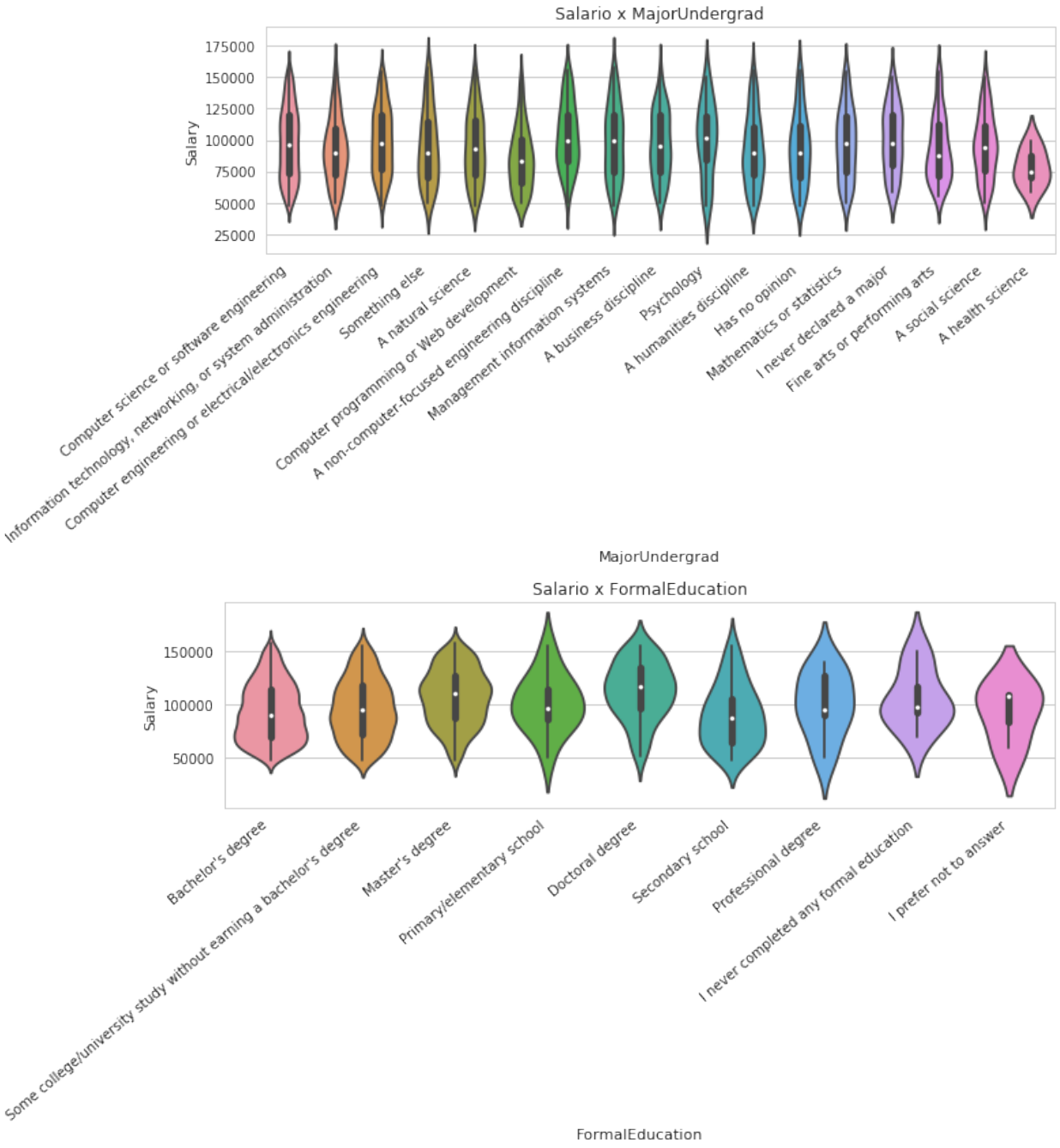

Salario x InfluenceRecruitment
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Not much influence
Has no opinion
No influence at all
A lot of influence
I am the final decision maker
Some influence
InfluenceRecruitment


Salario x InfluenceCommunication
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
A lot of influence
Has no opinion
Not much influence
Some influence
I am the final decision maker
No influence at all
InfluenceCommunication


Salario x Gender
Salary
200000
150000
100000
50000
0
Male
Has no opinion
Female
Other
Female; Transgender
Gender non-conforming
Male; Gender non-conforming
Male; Other
Male; Transgender
Female; Transgender; Gender non-conforming
Female; Gender non-conforming
Transgender; Gender non-conforming
Transgender
Male; Gender non-conforming; Other
Female; Transgender; Other
Gender


Salario x HighestEducationParents
Salary
175000
150000
125000
100000
75000
50000
25000
A master's degree
Has no opinion
High school
A bachelor's degree
Some college/university study, no bachelor's degree
A professional degree
A doctoral degree
I don't know/not sure
Primary/elementary school
No education
I prefer not to answer
HighestEducationParents


Salario x SurveyLong
Salary
180000
160000
140000
120000
100000
80000
60000
40000
20000
Disagree
Has no opinion
Agree
Somewhat agree
Strongly agree
Strongly disagree
SurveyLong



Temos um número bem grande de features (52) que somente tem perguntas com uma única resposta, assim gerando bastante gráficos.

A maioria das perguntas tem a mediana e a distribuição iguais entre as respostas, os únicos que podemos fazer algumas pontuações são:

- YearsProgram: A quantidade de anos programando implica diretamente na senioridade que implica no salario ganho, então podemos ver nas pessoas que tem mais anos programando tem os maiores salários.
- YearsrCodedJob: É a mesma coisa que o YearsProgram, porém já é diretamente nos anos de trabalho, segundo a logica e o gráfico, pessoas que tem mais experiência no trabalho ganham mais.
- FormalEducation: É o nível de escolaridade do usuário, no gráfico mostra que realmente temos uma pequena diferença quando o usuário tem mestrado ou um doutorado, apesar de não ser grande, tem.

Algoritmos e técnicas

Como é um problema de regressão, vamos usar modelos de regressão fornecidos pela biblioteca Scikit-learn, primeiramente vou detalhar mais o que é, e como são os modelos de regressões.

Regressão é uma analise que cria uma estimativa com o relacionamento da variavel alvo com uma ou mais variavies independentes, existem diversos tipos de regressão e elas são definidas principalmente por três características: Número de variáveis independentes, tipo da variável independente e como é feito a forma da linha da regressão (na qual será a “estimativa”)

Agora vou entrar nos algoritmos que vamos utilizar e o porquê deles:

- Regressão Linear é a mais famosa das regressões, vamos utilizar por ser a mais “simples e rápida” de se testar, ela contém uma variável dependente que será nesse caso o salário, e irá conter todas as nossas questões escolhidas para fazer uma reta e assim dar a nossa estimativa, as questões serão totalmente independentes entre elas. Um grande problema desse modelo é que ele é altamente sensível aos outliers.
- Regressão polinomial é bem parecido com a regressão linear, mas invés de uma reta, ela pode fazer uma linha que não seja uma reta, porém pode gerar um grande overfitting.
- Regressão Ridge é uma regressão onde o seu método de regularização é suavizar atributos que tem correlação com outros e que aumentam o ruído no modelo em geral (multicolinearidade), com a retiradas desses atributos com multicolinearidade o modelo tende a ter uma estabilidade melhor.
- Regressão de Lasso segue quase o mesmo método de regularização do Ridge, porém ele penaliza os coeficientes dos atributos com o seu valor absoluto, se a penalização for tão grande que o coeficiente chega no zero, o atributo não terá mais valor e será eliminado do modelo.
- Por ultimo temos um Ensemble, o Random Forest, onde usa várias arvores de decisão em conjunto para montar o melhor modelo.

Não faremos uma escolha agora de qual será o modelo, isso será feito com a busca em matriz (GridSeach, também fornecido pelo Scikit-learn), então não a necessidade de escolher um previamente, vamos escolher qual se adapta melhor aos nossos dados.

Também utilizaremos algumas técnicas de selecionamento de atributos (feature selection) que vamos decidir qual será utilizada, porém a que vamos testar será:

- PCA: É usado para reduzir dimensões, fazendo junções de features que tem um significado. Para as junções são pegos as features com dimensões correlacionadas positivamente ou negativamente.
- F_Regression: Usada para calcular a importância de cada feature. É feito um calculo onde basicamente correlaciona a feature com a variável alvo.

Como mencionado acima, vamos utilizar a busca em matriz tanto para a escolha do melhor modelo de regressão, melhor selecionador de atributos, e também para a quantidade de atributos que devemos ter, e consequentemente para determinar melhores valores de ajuste para o modelo escolhido.

Benchmark

No benchmark vamos ter seguintes considerações:

- Foi rodado em um MacBook Pro (13-inch, Mid 2010), então o tempo pode ser variado dependendo da maquina.
- Vamos pegar usuários somente nestas seguintes condições:
    - Esteja nos Estados Unidos
    - Trabalha em tempo integral
    - Ganha em Dólares
- Foram aplicados uma remoção de outliers dos salários (> 5% & < 95% da distribuição)
- Foram utilizados todas as features disponível, e que faziam um mero sentido.
- Todas as features que eram categóricas passaram por transformação de dummies.
- Colocamos uma normalização nos salários com RobustScaler do ScikitLearn.

Com as considerações acima, tivemos 1.386 amostras de 3.343, 970 amostras (70%) para treinamento e 416 amostras (30%) para teste

Vamos fazer o benchmark rodando um Ridge, especificamente um Ridge Bayesiano, já que foi um dos únicos que conseguiu rodar nessa quantidade de features e pontos.
Segue os resultados:

- No treino:
    - Tempo de previsão foi 0.0020
    - R2 Score: 0.59 (Não está fazendo overfitting)

    - Media do erro ao quadrado: 0.17
- No teste:
    - Tempo de previsão foi 0.0008
    - R2 Score: 0.39
    - Media do erro ao quadrado: 0.25

Também rodamos um algoritmo burro como outro benchmark. Calculamos o R2 Score e a media do erro ao quadrado com a média dos salários em relação a todos salários, os resultados foram:

- R2 score: 0.00
- Media do erro ao quadrado: 0.41

Metodologia

Pré-processamento de dados

O primeiro passo que deveríamos ter no pré-processamento seria a retirada das linhas que contiverem valores NA descrito anteriormente, porém se aplicarmos isso na nossa base de dados vai nos restar somente 800 amostras, se colocamos o mesmo modelo que o benchmart o R2 score cai para 0.35. Podemos remover as features, mas assim não vamos conseguir testar todas as features no feature selection.

Se retirarmos as features numéricas (que retiravam linhas que continha nulos do mesmo), temos 3343 amostras no total e conseguimos aumentar só com isso o R2 score para 0.41

Com essas considerações, vamos para o pré-processamento de fato:

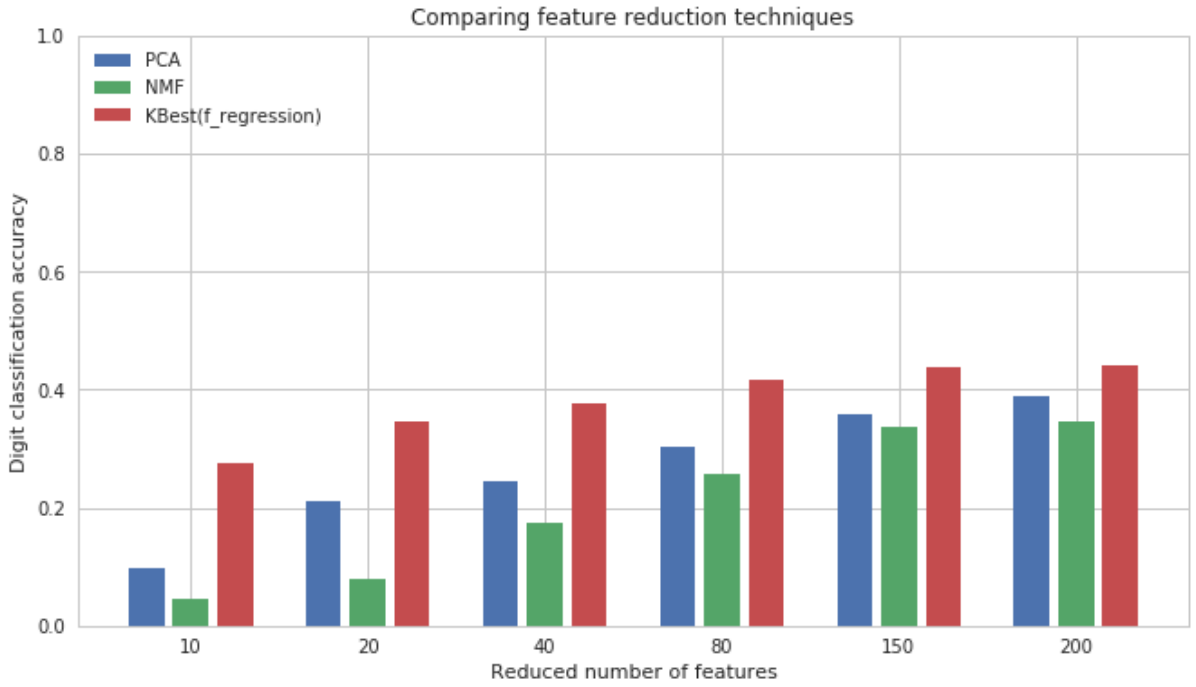
Pegamos todos os dados que contém algum NA e colocamos como “Has no option”.

Em seguida selecionamos somente quem tem salário definido, sem nulos para a variável alvo.

Por ultimo fazemos a mesma normalização nos salários feito no benchmark, usando o RobustScaler, uma normalização para dados que tem outliers.

Depois pegamos todas as features que tem múltiplas respostas, fazemos uma varredura em todas e colocamos uma dummie para cada resposta. Fazemos um dummie simples nas features categóricas restantes, assim nos dando 509 features no total.

Uma parte que teve bastante processamento de maquina assim levando bastante tempo, foi a busca em matriz de qual feature selection seria melhor, testamos PCA, NMF e KBest com f_regression.



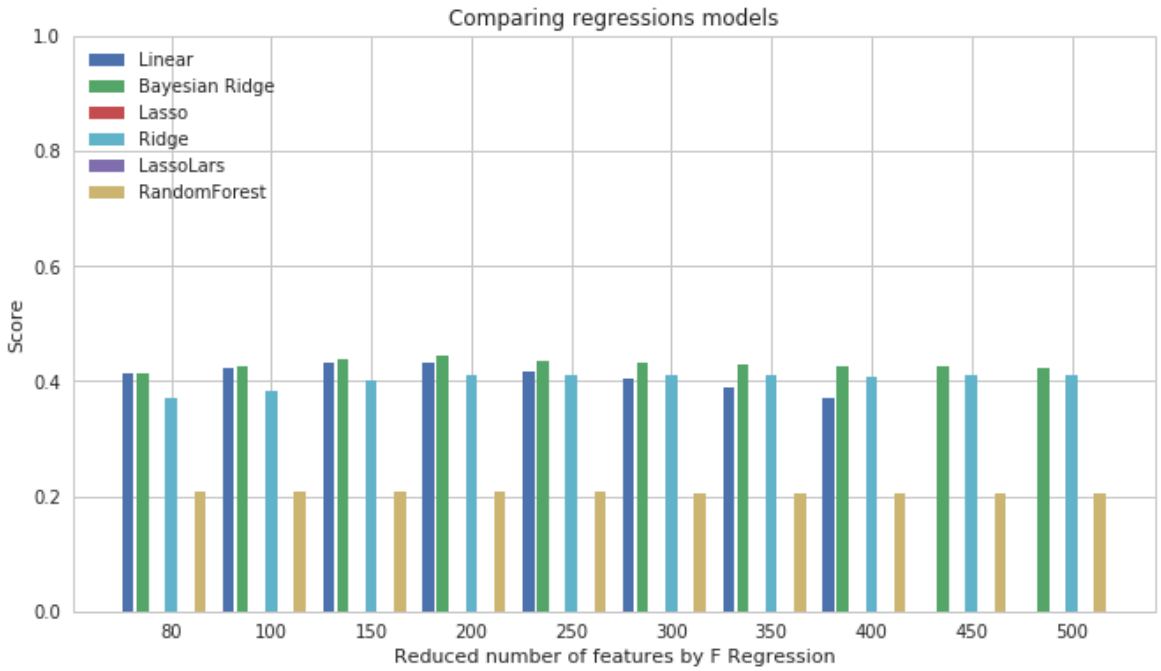
F_regression se mostrou melhor em todos os casos, infelizmente a nossa maquina não tem processamento o suficiente para fazer a busca em matriz com numero de features melhores, por isso vamos escolher F_regression (que por sinal, ela é bem rápida) e assim testar a quantidade ideal de features com F_regression junto com o melhor modelo de regressão.

Implementação

Primeiramente devemos selecionar qual é a melhor regressão entre:

- Linear
- Ridge normal
- Ridge Bayesiano
- Lasso
- LassoLars
- Random Forest

Vamos decidir isso por uma busca em matriz (Grid Search) em conjunto de uma quantidade ideal de atributos pelo F Regression, já que no Grid Search para escolher o melhor feature selection não conseguimos colocar números grandes de atributos.



Observando o gráfico podemos fazer algumas afirmações:

- Nenhum modelo conseguiu melhorar muito o R2 Score, já que ainda estamos na barreira de 0.4, mesmo com o selecionamento das melhores features.
- O modelo linear não conseguiu mais correlacionar a partir de 400 features.
- LassoLars não conseguiu se desenvolver.
- Lasso é melhor para poucas features e Ridge é melhor com o aumento das features, assim Lasso começa melhor e Ridge ultrapassa ao longo do aumento das features.
- O que se desenvolveu um pouco melhor e não teve muita mudança com o aumento de features foi o mesmo modelo que utilizamos no benchmark, o Ridge Bayesiano, fazendo seu pico em 200 features. Esse será o modelo escolhido.
- Random Forest não teve um bom desempenho, talvez colocando os parâmetros melhore e muito, mas não vamos acatar ele e sim o Ridge Bayesiano.

Para não ter que ficar processando isso novamente, pois o carregamento é bem demorado, e para economizar recurso em si, vamos usar isso apenas como dado estatístico, processar o número de features exato e usar o modelo Ridge Bayesiano
Ao treinar com o Ridge Bayesiano com 200 features dadas pelo SelecKBest da biblioteca Scikit-learn.

Para deixar o nosso modelo robusto, vamos utilizar uma validação cruzada com K-Fold, onde temos K divisões exatas no dataset (no nosso caso três), logo é feito as interações e selecionado um melhor conjunto de dados.

Temos os seguintes resultados:

- R2 Score de Treino: 0.53
- Média do erro ao quadrado: 0.19
- R2 Score de Teste: 0.45
- Média do erro ao quadrado: 0.21

Refinamento

Novamente utilizaremos o Grid Search para escolher os melhores valores dos parâmetros do Ridge Bayesiano. Os parâmetros serão:

- Normalize, se deve normalizar o dado
- N Iter, máximo de número de interações
- Fit Intercept, se deve calcular a interceptação do modelo

Poderíamos ter testado outros parâmetros, mas pelo pequeno poder de processamento que temos não conseguimos emular tudo.

Segundo os melhores parâmetros foram:

- FitIntercept: True
- Normalize: True
- Max Iter: 100

O resultado com os parâmetros ajustados foram:

- R2 Score de Treino: 0.54
- R2 Score de Teste: 0.44

Não tivemos nenhuma mudança, nem no treino e nem no teste, provavelmente deve ser o motivo de não conseguirmos emular os outros parâmetros.

Resultados

Modelo de avaliação e validação

Para testa a robustez de um modelo, precisamos colocar algum ruído nele.

Vamos testar o nosso modelo colocando algumas linhas no dataset que não tem muita coerência com a nossa proposta, onde as pessoas estão no Estados Unidos, ganham em dólares e são empregados em tempo integral.

O dataset completo tem dados de diversos países, podemos utilizar esses dados para criar ruído no nosso modelo, então vamos colocar linhas que não são do pais dos nossos estudos, Estados Unidos.

Vamos acrescentar ao nosso dataset os usuários do Brasil com qualquer categoria de trabalho e com qualquer tipo de moeda. Fazendo a limpeza de salários nulos, temos 237 linhas, que da em torno de 7% das linhas que tivemos anteriormente.

Utilizaremos as mesmas 200 features selecionadas no processo anterior para essas novas linhas.

Vamos treinar o modelo tanto com os dados que usamos neste projeto mais o nosso ruído, que será os dados brasileiros.

Fazendo a regressão, temos agora os seguintes resultados:

- R2 Score no treino: 0.50
- Média do erro ao quadrado de treino: 0.27
- R2 Score no teste: 0.42
- Média do erro ao quadrado de teste: 0.31

Tivemos uma pequena queda de 2 pontos no score de teste e tivemos uma redução na média do erro ao quadrado, acho esse resultado favorável, já que colocamos 237 linhas de ruído, se o modelo tivesse em overfitting ele não ia conseguir generalizar isso e teria uma queda brusca no score.

Justificativa

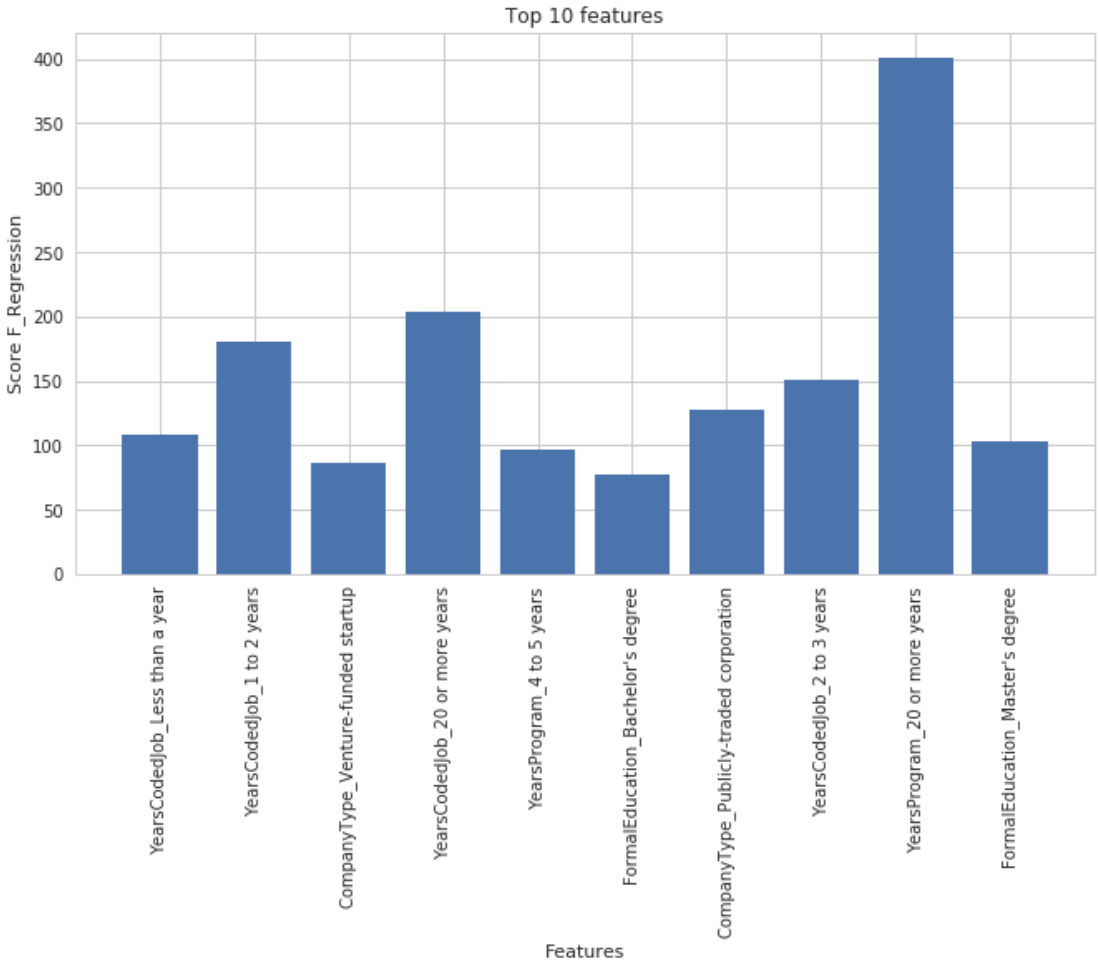
- Benchmark:
    - R2 Score de treino: 0.59
    - Media do erro ao quadrado no treino: 0.17
    - R2 Score de teste: 0.39
    - Media do erro ao quadrado no teste: 0.25
- No teste:
    - R2 Score de treino: 0.53
    - Media do erro ao quadrado no treino: 0.19
    - R2 Score de teste: 0.44
    - Media no erro ao quadrado no teste: 0.21

Bem, tivemos uma pequena melhora comparado ao benchmark, creio que não aumentamos muito o valor da diferença pois já no benchmark usávamos o melhor modelo para essa proposta e selecionando um numero de features conseguimos aumentar 5 pontos no R2 Score. Nossa média de erro ao quadrado não chega ser muito bom, porém conseguimos melhorar do benchmark alguns pontos.

Creio que se tivéssemos como ter um processamento melhor para emular outros parâmetros conseguíramos bater no 0.48 – 0.50, mas não mais do que isso.

Conclusão

Forma livre de visualização



Podemos ver que a feature mais importante disparadamente que determina seu salario realmente é a sua experiência, tanto que 20 anos ou mais de experiência foi a feature com mais pontos.

O tipo da companhia e a escolaridade foi um influenciador determinante.

Reflexão

Esse projeto foi bem desafiador, principalmente por uma quantidade absurda de atributos, e todos eles serem categóricas.

No começo do projeto decidimos pegar apenas dados de pessoas que moram no Estados Unidos, trabalham em tempo integral e ganham em dólares, com essas afirmações fizemos dois benchmarks, um com um modelo de Ridge Bayesiano que era um dos poucos modelos de regressão que aguentavam a quantidade grande de atributos e um algoritmo burro que sempre dava a média dos salários, assim comparando eles com os resultados. O primeiro teve um R2 score de 0.39 e uma média de erro ao quadrado de 0.25 enquanto o segundo zerou no score e teve uma média de erro ao quadrado de 0.41.

No pre-processamento limpamos os dados que não tinha salario, pegamos todas as colunas categóricas e colocamos em dummies, também fizemos dummies para atributos que vinham de perguntas com múltiplas respostas.

Na implementação, depois de limpar a nossa base rodamos um grid search para procurar o melhor feature selecition / feature reducer, o vencendor foi o F Regression. Por motivos de pouco processamento colocamos poucas opções de números de features no grid search, agora com um feature selecion escolhido podemos fazer outro grid search em busca do melhor modelo de regressão e a quantidade ideal de feature, o vencedor foi o mesmo do benchmark, o ridge bayesiano com 200 features.

No refinamento tentamos realizar outro grid search para pegar os melhores valores dos parâmetros do nosso modelo escolhido, porém por ter pouco poder computacional não conseguimos estimar muitas possibilidades e muitos parâmetros, então os resultados não ajudaram em praticamente nada.

Os resultados tiveram uma melhora significativa nas duas métricas, provando ser um modelo satisfatório.

Finalizando, testamos a nossa robustez colocando dados fora do proposto incialmente e conseguimos bons resultados provando que nosso modelo consegue aguentar sim algum ruído.

Esse projeto me fez aprender dois temas principais:

- Feature selection: Pela quantidade gigante de atributos, tive que aprender formas de escolher quais seriam as melhores features, comecei a estudar os selecionadores de forma geral até chegar nos que são especifico para regressões. Neste projeto apliquei diversas técnicas e o resultado foi mostrado nesse estudo
- Grid Search: Aprendi grid search pelo próprio nanodegree, mas esse projeto me fez aprofundar bastante nesse assunto, já que são diversos tipos de modelos de regressão e uma quantidade de features a ser escolhida.

Quase nenhuma feature tinha uma correlação extremamente forte, e isso dificultou bastante o projeto. Acho que isso é explicado pela forma que o mercado de desenvolvimento trabalha, não temos um norte especifico e uma base salarial, onde uma pessoa com X habilidades trabalha para uma empesa por um salario Y, em outra empresa ela vai receber um salario Z completamente diferente.

Fico feliz que consegui criar um modelo que consiga pelo menos correlacionar pequenas coisas, acho que 0.4 é uma pontuação alta para um universo onde a escolha do salário não tem uma forma clara e definida, claro que com mais experiência com machine learning e um processamento melhor essa pontuação pode ser facilmente aumentada.

Melhorias

Sei que temos algoritmos como ensembles e rede neural como modelos de regressão também, porém não consegui ter nenhum resultado significantes com ele pelo motivo de pouca experiência com os mesmos, por isso não foram utilizados nesse projeto, preferi manter no campo das regressões “nativos”.

Com certeza podemos melhorar o algoritmo com técnicas mais sofisticadas na qual não tenho experiência ainda. Como citado algumas vezes, o fator do processamento limitou a gente em algumas partes na qual precisávamos de um processamento um pouco mais robusto para ter uma melhor busca em matriz.