(Data Mining)

Inhalt

Einführung

Dieses Dokument enthält 32561 Stichproben und 15 Variablen und stellt eine Datenanalyse dar, die auf den Attributen Alter, Arbeitsbereich, Gewichtungszensus, Bildung, Bildungsnummer, Ehestatus, Beruf, Beziehung, Rasse, Geschlecht, Kapitalgewinn, Kapitalverluste, Stunden pro Woche, Land und Einkommen basiert. Dabei wurden ein Vergleich und eine Analyse der Einkommensspalte mit allen anderen Spalten durchgeführt.

- Alt : Alter einer Person
- Arbeitsbereich : Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked
- GewichtungsZensus : Das von der Volkszählungsbehörde vergebene Gewicht
- Bildung : Bachelor, Some-college, 11., HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9., 7. bis 8., 12., Master, 1. bis 4., 10., Doktorat, 5. bis 6., Vorschule
- Bildungsnum : Kategorische Variable der Bildung
- Ehestatus : Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse
- Beruf : Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspect, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces
- Beziehung : Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried
- Rasse : White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black
- Sex : Female, Male
- Kapitalgewinn : Kapital-Gewinn
- Kapitalverluste : Kapitalverlust
- Stundenprowoche : Anzahl der Stunden, die die Person in der Woche gearbeitet hat
- Land : United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands
- Einkommen : >50K, <=50K[1]

Folgende Darstellungen zeigen die alle Spalten.

	Alt	Arbeitsbereich	GewichtungsZensus	Bildung	Bildungsnum	Ehestatus	Beruf	Beziehung	Rasse	Sex	Kapitalgewinn	Kapitalverluste	Stundenprowoche	Land	Einkommen
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K

[1] Adult Data Set

Methoden

Die beliebtesten Vorverarbeitungsbibliotheken, die in der Programmiersprache Python geschrieben wurden, sind Pandas, Numpy und Sklearn. Die Daten werden dann in eine Tabellenkalkulation übertragen und mit Attributen versehen. Zur Analyse der Daten wurden statistische Methoden verwendet, um Korrelationen zwischen Variablen zu bestimmen. Der Zweck dieses Verfahrens besteht darin, ein tieferes Verständnis der Faktoren zu erlangen, die zu einem bestimmten Einkommen beitragen.

Die genutzten Methoden sind:

- train_test_split : zerlegt die Daten in Training- und Testdaten.
- k-Means : Datenpunkte in Gruppen basierend auf ähnlichen Merkmalen gruppiert.
- fit_transform : Transformieren der Parameter.
- fit : Das Training und Testdaten auf Modell anwenden.
- predict : mit dem KNN-Vorhersagen treffen werden.
- compile : Kompilieren des Modells mit Optimizer Methode.
- MinMaxScaler(): Normalisieren (Skalieren) die Datensätze besitzen erheblich große Werte.
- Apriori : Finden von Frequenzmuster in komplexen Datensätzen.
- DecisionTreeClassifier, scatter, pyplot : Diagramm zu erstellen.

Die meisten Datensätze enthalten Nullwerte, mit denen wir uns befassen müssen, bevor wir ein Modell erstellen. Die Handhabung dieser fehlenden Werte beinhaltet das Löschen von Proben mit fehlenden Werten, das Ersetzen durch Mittelwert, Median oder Modus und das Behandeln aller fehlenden Werte als neue Klasse (wahrscheinlich nur in wenigen Fällen). In diesem Datensatz gibt es Nullwerte mit '?'.

so behandeln wir dieses Problem wie in folgende Abbildung.

```
data['Arbeitsbereich'].fillna(data['Arbeitsbereich'].mode()[0], inplace=True)
data['Beruf'].fillna(data['Beruf'].mode()[0], inplace=True)
data['Land'].fillna(data['Land'].mode()[0], inplace=True)
data.isnull().sum()
```

Die Werte der Variablen können zwischen unterschiedlichen Bereichen variieren, und große Diskrepanzen bei der Größenordnung der Variablen können die Prognose der Zielvariablen beeinträchtigen. Um die Daten auf einen vereinheitlichten skalierten Wertebereich zu bringen, können MinMaxScaler, StandardScaler und Normalizer wirksam sein.

Wie wir sehen können, haben die unabhängigen Variablen des Datensatzes unterschiedliche Datentypen. Aus Konsistenzgründen wurden die Spalten [Berufsebene, Bildung, Familienstand, Beruf, Beziehung, Rasse, Geschlecht, Herkunftsland] in kategoriale Variablen umgewandelt. Auch die Zielgröße „Einkommen“ wird in zwei Kategorien eingeteilt: mehr als 50.000 pro Jahr oder weniger als 50.000 pro Jahr. Der Datensatz wird 70:30 in Trainings- und Testdatensätze aufgeteilt, und die unabhängigen Variablen werden unter Verwendung von Standard-Skalierungstechniken skaliert, um Daten mit Null-Mittelwert und Einheitsvarianz zu erhalten, und das steht im nächste Diagramm.

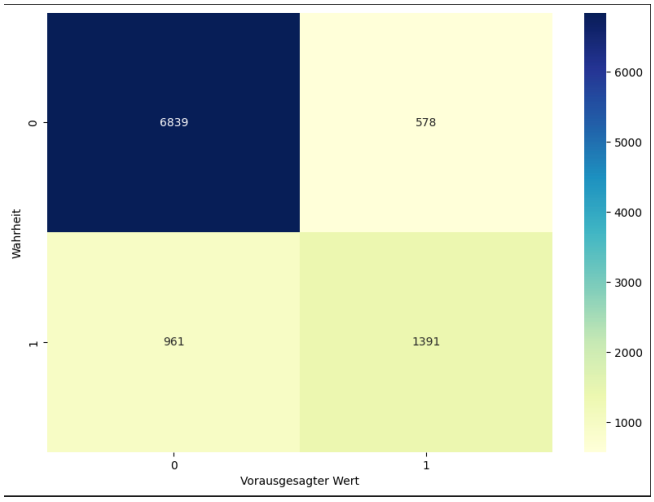
Modell Aufbau

1 Train_test_split

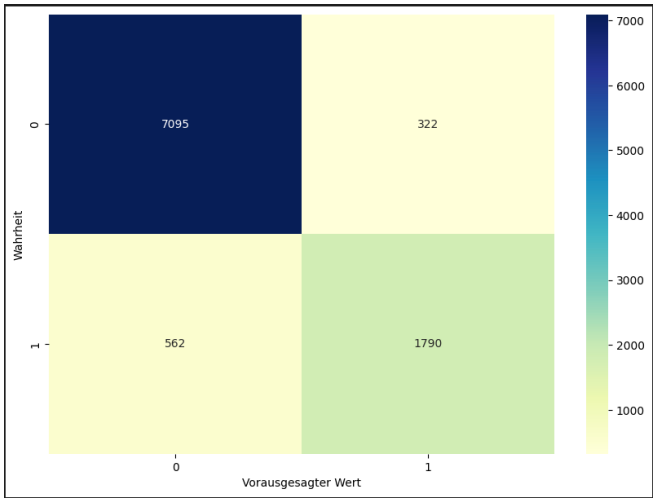
Train_test_split ist eine Funktion, die es ermöglicht, einen Datensatz in eine Trainings- und eine Testgruppe zu unterteilen. Dies ist ein wesentlicher Bestandteil des maschinellen Lernens, da es es ermöglicht, ein Modell zu trainieren und zu validieren.

Der Train_test_split-Funktion nimmt als Argumente den Datensatz und die Größen der Trainings- und Testgruppen (normalerweise 80% / 20%) an und gibt zwei Arten zurück - eine Trainingsgruppe und eine Testgruppe. Die Trainingsgruppe wird verwendet, um das Modell zu trainieren, und die Testgruppe wird verwendet, um das Modell zu validieren.

Die Daten wird trainiert und visualisieren, es wird die für 75% train und 25% test, In folgendem Diagramm wird das Train-data als Diagramm erstellen.



Und die nächste Diagramm wird das Test-data als Diagramm erstellen.



2.SVM

Eine Support Vector Machine (SVM) : ist ein überwachter maschineller Lernalgorithmus, der sowohl für Klassifizierungs- als auch für Regressionsprobleme verwendet werden kann. Es wird hauptsächlich zur Lösung von Klassifizierungsproblemen verwendet und stellt jedes Datenelement als Punkt im n-dimensionalen Raum dar, wobei der Wert jedes Merkmals der Wert einer bestimmten Koordinate ist. Die Klassifizierung wird dann durchgeführt, indem die Hyperebene gefunden wird, die die beiden Klassen am besten unterscheidet.

Der SVM-Kernel ist eine Funktion, die einen niedrigdimensionalen Eingaberaum in einen hochdimensionalen Raum i umwandelt. H. Sie macht aus einem unteilbaren Problem ein teilbares. Es ist besonders nützlich bei nichtlinearen Trennungsproblemen, da es einige äußerst komplexe Datentransformationen durchführt und getrennte Daten basierend auf von Ihnen definierten Beschriftungen oder Ausgaben findet, und das wird im folgenden Abbildungen zeigen.

```
#SVC
svc.fit(X_train,y_train)
y_pred_svc = svc.predict(X_test)
```

```
Support Vector Machine:
Genauigkeitsgrad: 59.21
F1 Score:  32.099
Mittlerer quadratischer Fehler:  40.79
```

3. Precision

Precision ist die Genauigkeit einer Zahl, die durch die Anzahl der Dezimalstellen bestimmt wird. Es gibt auch eine Funktion namens round(), die die Genauigkeit einer Zahl auf eine bestimmte Anzahl von Dezimalstellen einstellen kann.

4. Recall

Recall ist eine Funktion, die zur Berechnung der Erinnerungsgenauigkeit verwendet wird. Es berechnet das Verhältnis der Anzahl der vorhergesagten positiven Ergebnisse, die tatsächlich positiv sind, zu der Gesamtzahl der tatsächlich positiven Ergebnisse.

5. F1-score

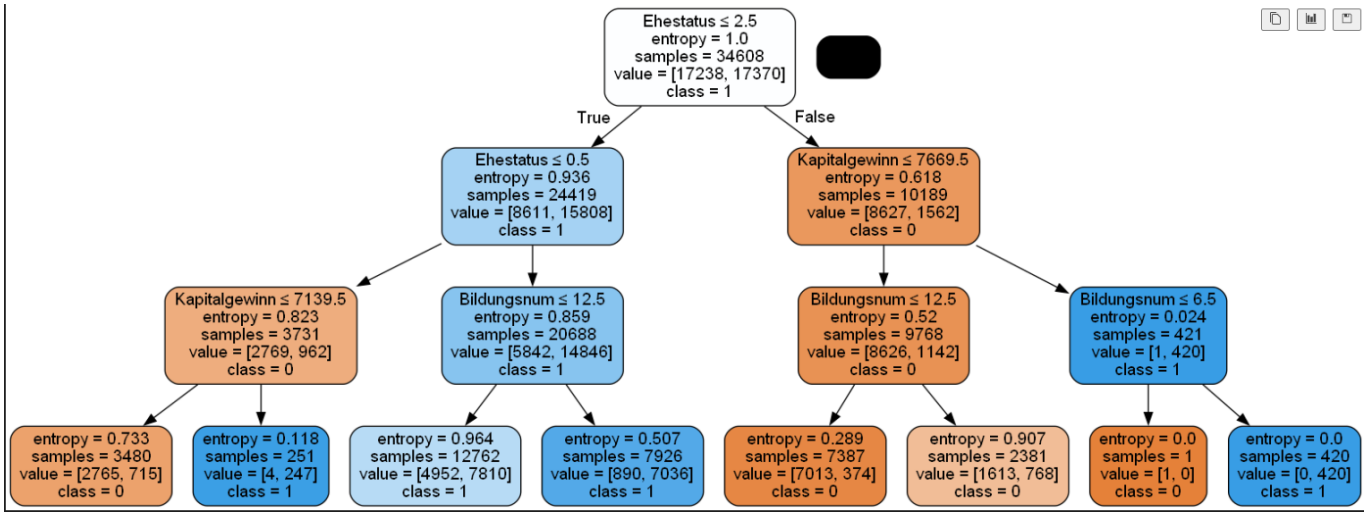
F1-score ist ein Messwert für die Qualität eines Klassifikators. Es wird verwendet, um zu messen, wie gut ein Klassifikator Vorhersagen basierend auf einer bestimmten Testdatenmenge macht. F1-score ist der harmonische Mittelwert von Präzision und Erkennung.

```
GaussianNB/ Naive Byers:
Genauigkeitsgrad: 62.999
F1 Score:  45.262
Mittlerer quadratischer Fehler:  37.001
RandomForestClassifier:
Genauigkeitsgrad: 92.496
F1 Score:  92.734
Mittlerer quadratischer Fehler:  7.504
DecisionTreeClassifier:
Genauigkeitsgrad: 90.844
F1 Score:  91.222
Mittlerer quadratischer Fehler:  9.156
Support Vector Machine:
Genauigkeitsgrad: 59.21
F1 Score:  32.099
Mittlerer quadratischer Fehler:  40.79
```

6. Entscheidungsbaum

Ein Entscheidungsbaum ist ein überwachter Lernalgorithmus (mit einer vordefinierten Zielgröße), der hauptsächlich für Klassifizierungsprobleme verwendet wird. Es funktioniert sowohl für kategoriale als auch kontinuierliche Eingabe- und Ausgabevariablen. Bei dieser Technik wird eine Population oder Stichprobe in zwei oder mehr

homogene Gruppen (oder Subpopulationen) unterteilt, basierend auf den charakteristischsten Merkmalen in den Eingabevariablen, wie im nächsten Foto.



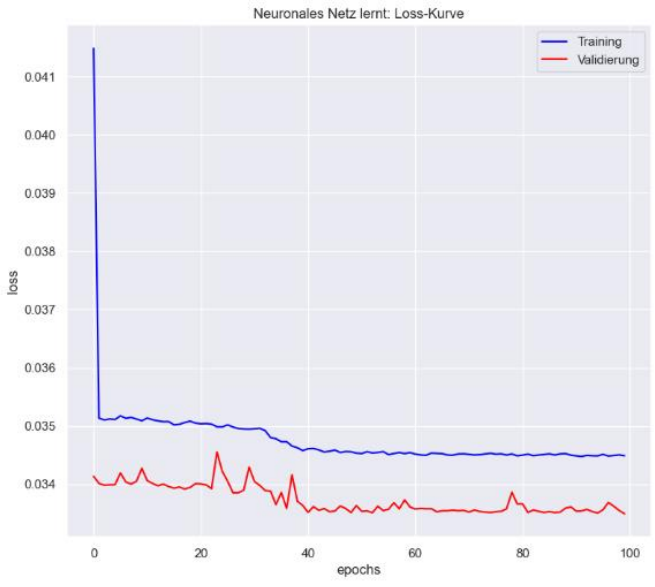
Neuronale Netze

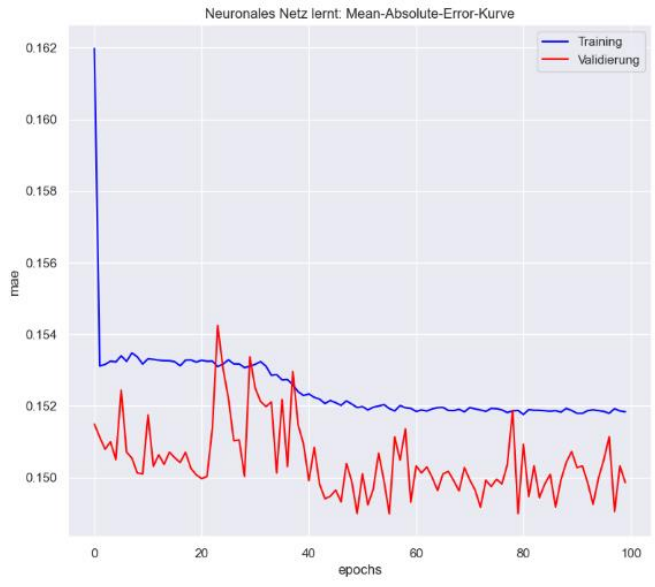
Neuronale Netze sind eine Form der künstlichen Intelligenz, die auf dem Konzept von künstlichen Neuronen Systemen basiert. Sie sind eine Form der maschinellen Lernmethode, die es ermöglicht, selbstorganisierende, mehrschichtige Netzwerke aufzubauen, um mittels iterativer Lernverfahren komplexe Aufgaben zu lösen.

Neuronale Netze in Python können verwendet werden, um Daten zu klassifizieren. Mit einem neuronalen Netzwerk habe ich Klassifizierungsmodelle aufgebaut, und eine hohe Genauigkeit und Präzision aufgewiesen, das wird in nächste Bild zeigen.

```
Trainingsdaten:
764/764 [==============================] - 1s 742us/step
loss: 0.03
mae: 0.15
R2: 0.02
Testdaten:
255/255 [==============================] - 0s 823us/step
loss: 0.03
mae: 0.15
R2: 0.02
```

Und am Ende das Ergebnis visualisieren.

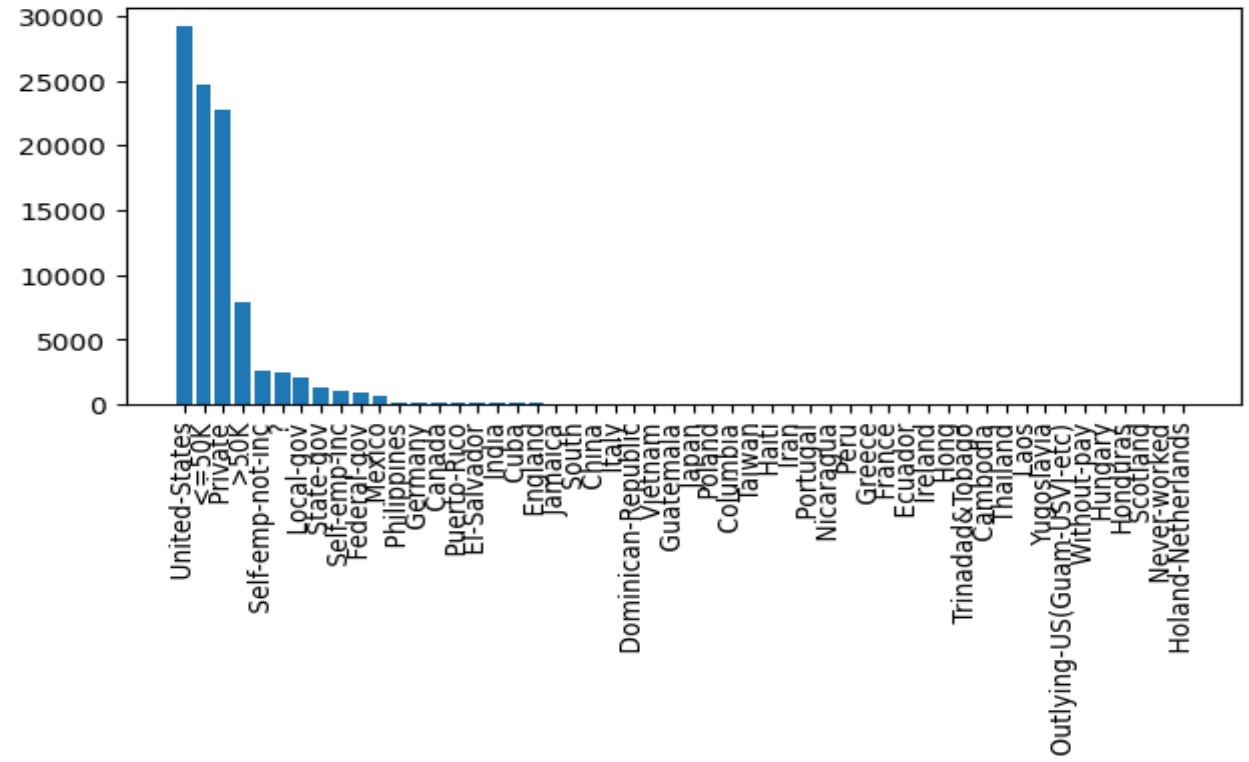




Assoziationsanalyse

Assoziationsanalyse in Python ist ein Verfahren, das verwendet wird, um Zusammenhänge zwischen Variablen in einem Datensatz zu untersuchen. Es kann verwendet werden, um zu sehen, ob es einen signifikanten Zusammenhang zwischen zwei oder mehr Variablen gibt, die sich auf eine bestimmte Ausgabe auswirken. Es kann verwendet werden, um Muster in komplexen Datensätzen zu erkennen, die durch andere Methoden möglicherweise übersehen werden. Es kann auch verwendet werden, um die Abhängigkeiten zwischen Variablen zu verstehen und zu veranschaulichen.

und in Projekts Fall werden die relevanten Spalten "Arbeitsbereich", "Land" und "Einkommen" aus dem Datensatz extrahiert. Danach wird der Datensatz in einem neuen Datensatz namens 'df2' zusammengefasst, wodurch die Spaltennamen und die Werte zu einer Spalte zusammengefasst werden. Als Nächstes wird die Anzahl der Artikel ermittelt, indem die Funktion "value_counts" verwendet wird, um ein Diagramm zu erstellen.



Als Nächstes werden die Duplikate aus dem Datensatz entfernt. Dann wird die Spalte "Anzahl" mit dem Wert 1 aufgefüllt. Danach wird der Datensatz in einen Pivot-Datensatz umgewandelt. Dann wird die Funktion "apriori" verwendet, um die häufigsten Elemente zu finden, die die Mindestunterstützung von 0,005 haben. Die Ergebnisse werden dann nach Unterstützung absteigend sortiert. Als Nächstes wird die Funktion "association_rules" verwendet, um die Regeln für eine Mindestvertrauenswürdigkeit von 0,25 zu erhalten. Die Ergebnisse werden dann nach Vertrauen absteigend sortiert. Schließlich werden die ersten 12 Regeln angezeigt.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
0	(<=50K)	(State-gov)	0.020408	0.020408	0.020408	1.0	49.0
1	(State-gov)	(<=50K)	0.020408	0.020408	0.020408	1.0	49.0
2	(United-States)	(<=50K)	0.020408	0.020408	0.020408	1.0	49.0
3	(<=50K)	(United-States)	0.020408	0.020408	0.020408	1.0	49.0
4	(South)	(?)	0.020408	0.020408	0.020408	1.0	49.0
5	(?)	(South)	0.020408	0.020408	0.020408	1.0	49.0
6	(United-States)	(State-gov)	0.020408	0.020408	0.020408	1.0	49.0
7	(State-gov)	(United-States)	0.020408	0.020408	0.020408	1.0	49.0
8	(United-States, <=50K)	(State-gov)	0.020408	0.020408	0.020408	1.0	49.0
9	(United-States, State-gov)	(<=50K)	0.020408	0.020408	0.020408	1.0	49.0
10	(<=50K, State-gov)	(United-States)	0.020408	0.020408	0.020408	1.0	49.0
11	(United-States)	(<=50K, State-gov)	0.020408	0.020408	0.020408	1.0	49.0