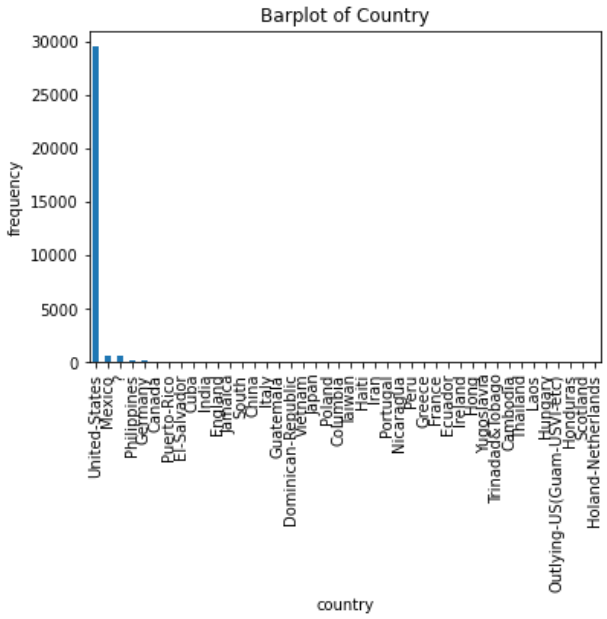
# IE30301-Final Report

Hypothesis and problem statement
In classification problem, an individual's annual income is predicted using 14 features, such as work class, education, marital and etc. I am trying to prove the following hypothesis: In the United States, the number of men and women who have annual income >50K is the highest, when the relationship status of man - husband, and woman-wife. By reading csv file and printing first 5 and last 5 rows, it is clearly seen, that annual income(our dependent variable) can have 2 values: either >50K or <=50K.

(1) Exploratory data analysis

By using the following plot, we observe distribution of values in feature ‘country':




It is clearly seen that the most frequent value is United States, and it takes the vast majority of all values. As our hypothesis states that we consider men and women in United States, we modify dataset so that it only contains rows for which country is United States, as following:

```
#Making dataset only with values for which, country is United-States
df = df[df['country'] == 'United-States']
```

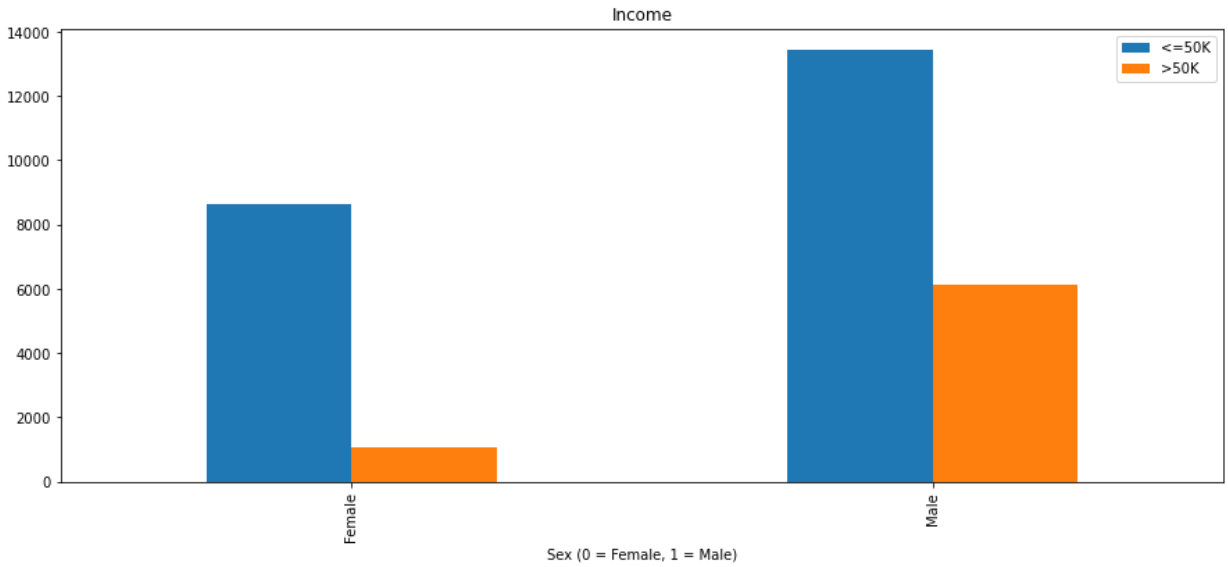
After it, we analyze whether our dataset is whether balanced or not. Using df[‘income’].value_counts(), we get that <=50K is 22269, while >50K is 7237, which means that our dataset is imbalanced and Ratio between >50K : <=50K = 1:3.0771.

Next step is to analyze data related to hypothesis. 2 features ‘sex’ and ‘relationship’ were grouped and using the following code, we observe distribution of income values for females and males with different relationship status:

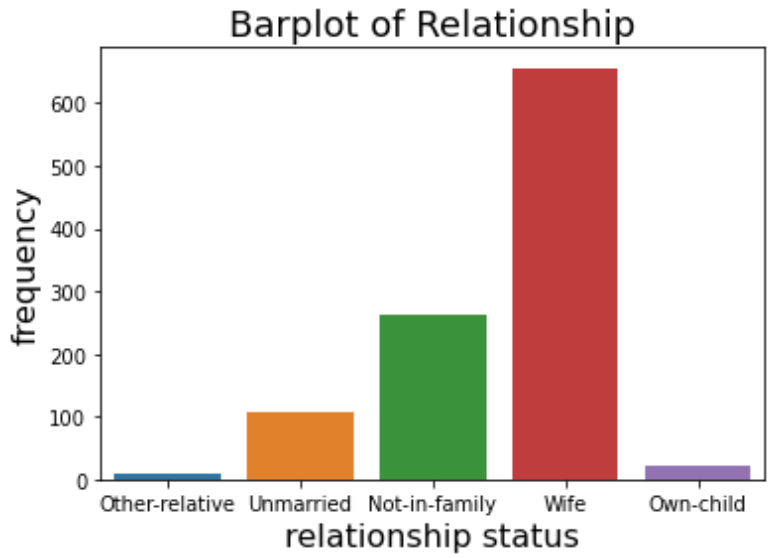
```
In [13]: df.groupby(['sex', 'relationship'])['income'].value_counts()

         #it shows one inconsistent data

Out[13]: sex     relationship    income
         Female  Not-in-family   <=50K     3333
                                 >50K       263
                 Other-relative  <=50K      309
                                 >50K        10
                 Own-child       <=50K     2012
                                 >50K        21
                 Unmarried       <=50K     2219
                                 >50K       107
                 Wife            <=50K      682
                                 >50K       656
         Male    Husband         <=50K     6492
                                 >50K      5399
                 Not-in-family   <=50K     3403
                                 >50K       510
                 Other-relative  <=50K      362
                                 >50K        17
                 Own-child       <=50K     2495
                                 >50K        34
                 Unmarried       <=50K      573
                                 >50K        92
                 Wife            >50K         1
         Name: income, dtype: int64
```

The following table clearly shows that number of females who earn >50K is the highest when their relationship status is wife - 656, while the highest number of females who earn <=50K is when they are not-in-family - 3333. Regarding males, the highest number for both earning <=50K and >50K is when male is husband - 6492 and 5399 respectively, which supports our hypothesis. Moreover, one irrelevant data was found, as male cannot be wife, and this data will be deleted in preprocessing part.

Now, using the following plot, we consider the ratio between women and men who earn <=50K and > 50K:

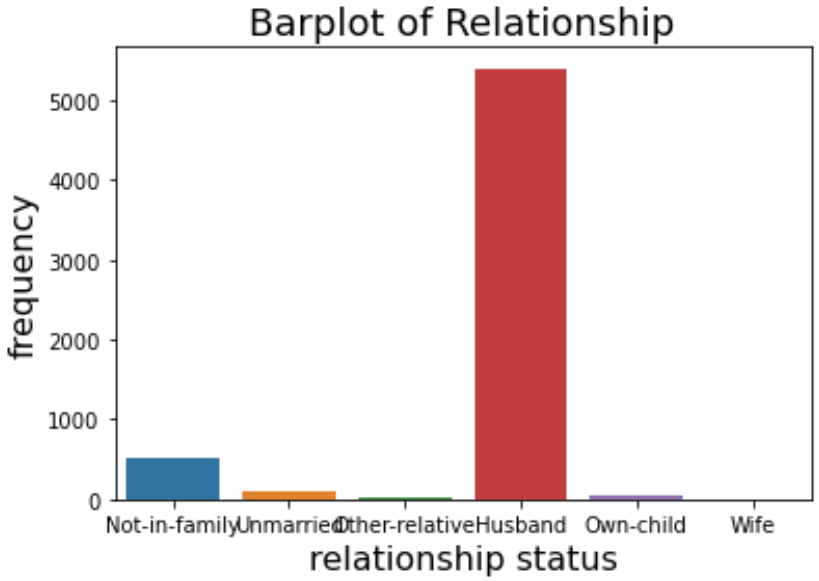



It is clearly seen that number of females and males who earn <=50K is quite larger than number of females and males who earn >50K.
Regarding women, the following plot was constructed to observe how ‘relationship’ feature’s values among females who earn >50K are distributed:

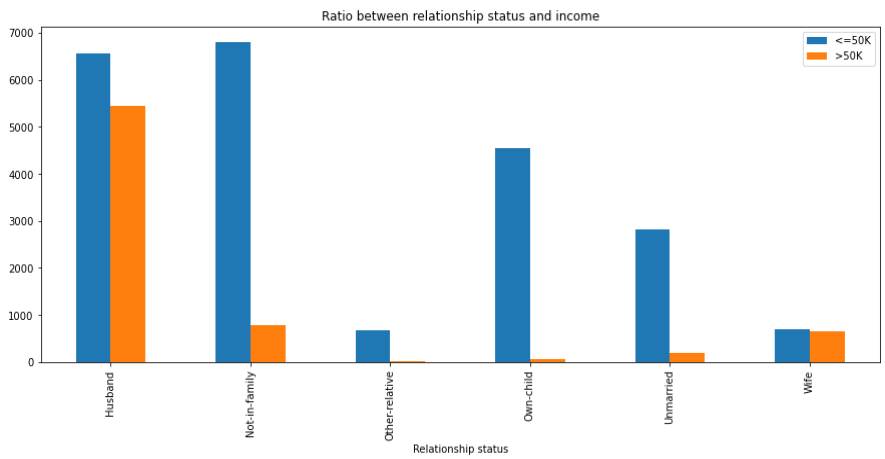



It clearly shows that number of women who have annual income >50K is the highest when relationship status of woman is wife, which totally supports our hypothesis.
The same was done for men, and the same trend was observed: number of man with >50K annual income is the highest with relationship status - husband.

But, in plot for females, not-in-family and unmarried status females also tend to earn >50K and they take quite sufficient part from total number of women who earn >50K, while for males, the gap between other relationship status and husband is very large so that they can be even not considered.



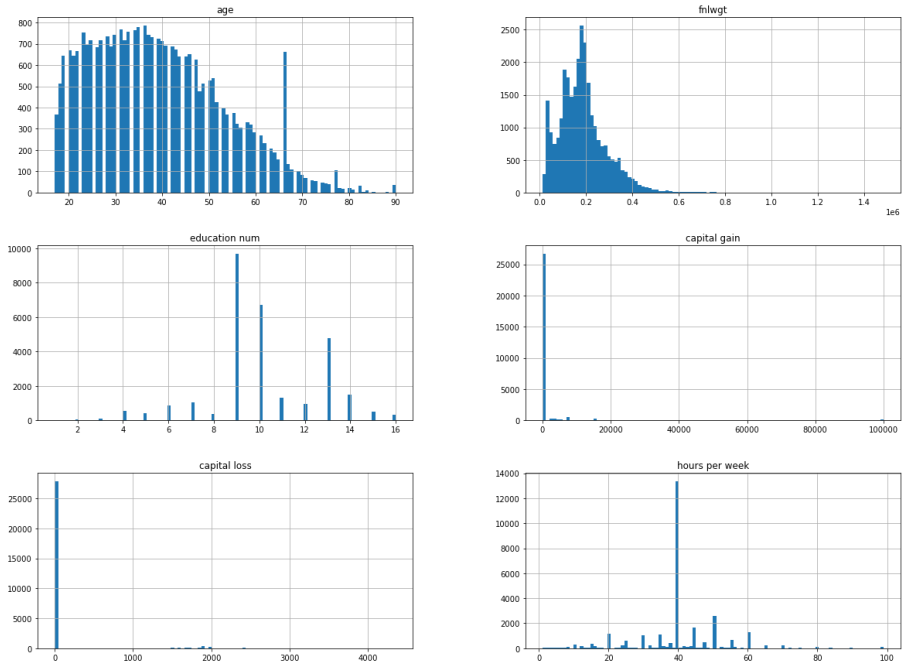

Consider the gap between numbers who earn <=50K and >50K being in one relationship status the following plot was constructed:



It showed that only in wife and husband relationship status the ratio between those who earn >50K and <=50K is approximately the same, while in other relationship statuses number of those who earn <=50K sufficiently outperforms number of those who earn >50K.

(2) Preprocessing

To observe outliers, and values that can be dropped, the following plot was obtained:




It shows that there is no any outliers that can be dropped. Studying the domain, I came to conclusion that capital gain and capital loss have to be included in annual come, so we do not drop these values.

Capital gain and capital loss mean the same, but with different sign, so we substract them and assign it to capital gain variable to reduce the dimensions and avoid multicollinearity problem, as following:

```
df['capital gain']=df['capital gain']-df['capital loss']
```

As, we made 2 features in one, we drop capital loss feature to avoid multicollinearity, as following:

```
df = df.drop(['capital loss'], axis=1)
```

Also, as for all rows, country value is United States, we can drop it to have less features in our model, by the same code as capital loss column was dropped.

Next, education num and education values were compared using following code:

```
pd.concat([df['education num'].value_counts() , df['education'].value_counts()], axis=0)
```

Code clearly showed that education and education num features represent the same, where each level of education corresponds to specific number in education num. That is why, we need to drop education column to avoid multicollinearity. If we would encode education as ordinal variable, we would get the same results as in education num, so we drop education column.
We use df.info to check what columns were dropped, and finally our model is left with 11 features.
Next, irrelevant data (male is wife) which was mentioned before is removed.
Now, let's take care of missing data. The following table represents % of missing data in each column:

| | Number of missing values | % |
|---|---|---|
| Capital gain | 591 | 2.0 |
| Workclass | 309 | 1.0 |
| Race | 293 | 1.0 |
| fnlwgt | 291 | 1.0 |
| marital | 290 | 1.0 |
| Hours per week | 289 | 1.0 |
| Education num | 287 | 1.0 |
| Age | 283 | 1.0 |
| Occupation | 276 | 0.9 |
| Relationship | 273 | 0.9 |
| sex | 245 | 0.8 |

It seems that each column has 1-2% of missing data. Next, we replace all ? signs in data with NaN, so that they would also be removed. In numeric variables such as age, hours per week and etc, all missing values were replaced by the mean of values in this feature, as following:

```
df.loc[df["age"].isnull(), "age"]=round(df["age"].mean(),0)
```

While, in all categorical variables such work class, marital and etc, all missing values were replaced by the mode, the most common category in this column, using the following function:

```
def miss_value(dataframe,colname):
     mode=dataframe[colname].mode()[0]
     dataframe[colname].fillna(mode,inplace=True)
```

```
for columns in cat_feat:
    miss_value(df,columns)
```

Now, we check percent of missing values in each column, and it turns out that finally there is no any missing values.

Next step is to check for duplicates, and using the following code, we drop duplicates, leaving the first duplicated value in our dataset. And finally we get 28918 entries.

```
feat = ['workclass','marital', 'occupation', 'relationship', 'race', 'sex','age','hours per
```

```
df.drop_duplicates(subset =feat,
                     keep = 'first', inplace = True)
```

Then, we split our data in test and train sets using train_test_split function from sklearn library. We use test size 20% and train set 80%, and use stratified sample so that ratio between <=50K and >50K in both test and train sets will be the same, as following:

```
from sklearn.model_selection import train_test_split
```

```
training_set, test_set = train_test_split(df, test_size=0.2, random_state=1, stratify=df[['i
```

And ratio of income in training set >50K : <=50K = 1:3.0672, while ratio of income in testing set >50K : <=50K = 1:3.0675.Then, we separate numerical and categorical(nominal/ordinal) columns, and obtain X_train, y_train from copy of training set, and X_test, y_test from copy of testing set. Then, one hot encoding was done to nominal columns, and as there is only one ordinal column which already in correct form, ordinal encoder was not applied:

```
from sklearn.preprocessing import OneHotEncoder, OrdinalEncoder
```

```
ohe = OneHotEncoder(sparse=False)
ohe.fit(X_train[nom_feat].values)
X_train_nom = ohe.transform(X_train[nom_feat].values)
X_test_nom=ohe.transform(X_test[nom_feat].values)
X_train_ord=X_train[ord_feat].values
X_test_ord=X_test[ord_feat].values
```

Then, Feature scaling was applied to numeric variables for both test and train sets, and then all 3 types of variables were connected together for both X_train and X_test using np.concatenate, as following:

```
scalar = StandardScaler()
scalar.fit(X_train[num_feat].values)
X_train_num = scalar.transform(X_train[num_feat].values)
X_test_num=scalar.transform(X_test[num_feat].values)
```

```
X_train = np.concatenate([X_train_num, X_train_ord, X_train_nom], axis=1)
X_test= np.concatenate([X_test_num, X_test_ord, X_test_nom], axis=1)
```

We get X_test with a shape (5784, 47) and X_train with a shape (23134, 47).

(3) Model train & test

The first model that was used is Logistic Regression, and as the dataset is imbalanced, we tune it with class weight='balanced', and with max_iter=800, whereas default was 100 and it was not enough.

```
from sklearn.linear_model import LogisticRegression
log_reg_cw = LogisticRegression(class_weight='balanced',max_iter=800)
log_reg_cw.fit(X_train, y_train)
accuracy_logreg=log_reg_cw.score(X_train, y_train)
```

Next model was built using Decision Tree algorithm, initially it was tuned only with random_state=0, but accuracy value on training set was equal to 1 which means overfitting. So, in order to prevent overfitting, max_depth=10 was used, as following:

```
from sklearn.tree import DecisionTreeClassifier
dectree = DecisionTreeClassifier(random_state=0,class_weight='balanced',max_depth=10)
dectree.fit(X_train, y_train)
accuracy_dectree=dectree.score(X_train, y_train)
```

And now value of accuracy on training set for decision tree became 0.82.

Random Forest was also tuned with several parameters to deal with imbalance data, and prevent overfitting. Number of estimators was assigned to 100, max_depth to 10 and class_weight='balanced', as following:

```
 from sklearn.ensemble import RandomForestClassifier
randfor = RandomForestClassifier(n_estimators=100, random_state=0,class_weight='balanced',ma
randfor.fit(X_train,y_train)
accuracy_randfor=randfor.score(X_train, y_train)
```

In SVM, 2 models were constructed. One with polynomial kernel and one with radial basis function kernel, and the only tuning parameter for both was class_weight='balanced'.
While in k-NN, model was tuned by 2 parameters, as following

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 15,weights='distance')
knn.fit(X_train, y_train)
accuracy_knn=knn.score(X_train,y_train)
```

And accuracy value for training set was 1.0, which means that model experiences overfitting.

(4) Result

For evaluation, 10-fold cross validation was used using the following function:

```
from sklearn.model_selection import cross_val_score
def validation(model, x, y):
    scores = cross_val_score(model, x, y, cv=10)
    return scores.mean()
```

And then, results of cross validation accuracy for all 6 models were summarized in one table as following:

| Model | Accuracy score on training set | Accuracy using 10-fold cross validation |
|---|---|---|
| Logistic Regression | 0.798435 | 0.795594 |
| Decision Tree | 0.819789 | 0.799501 |
| Random Forest | 0.803493 | 0.788125 |
| SVM(poly) | 0.803147 | 0.799917 |
| SVM(rbf) | 0.782441 | 0.779791 |
| k-NN | 1.000000 | 0.834359 |

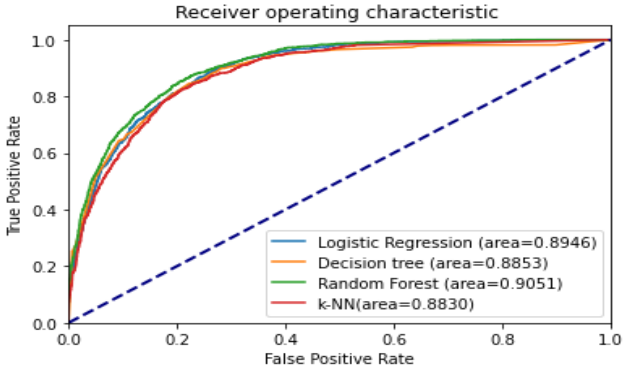
It clearly shows that k-NN has the highest accuracy usage 10-fold cross validation although it experiences overfitting. While comparing kernels for SVM, SVM with polynomial kernel performs better and gives a higher accuracy than SVM model with radial basis function kernel. Among models that were not overfitted, SVM with polynomial kernel has the highest accuracy, although they are quite the same.

The next evaluation metric that was used is ROC-AUC curve. It was plotted for 4 models, excluding SVM models. Curves were obtained using auc, roc_curve from sklearn.metrics, as following:

```
from sklearn.metrics import auc, roc_curve
#Logistic Regression
fpr, tpr, threshold = roc_curve(y_test, y_proba,pos_label='>50K')
rocauc = auc(fpr, tpr)
#Decision Tree
fpr1,tpr1,threshold1= roc_curve(y_test,y_proba1,pos_label='>50K')
rocauc1 = auc(fpr1, tpr1)
#Random Forest
fpr2,tpr2,threshold2= roc_curve(y_test,y_proba2,pos_label='>50K')
rocauc2 = auc(fpr2, tpr2)
#k-NN
fpr4,tpr4,threshold4= roc_curve(y_test,y_proba4,pos_label='>50K')
rocauc4 = auc(fpr4, tpr4)
```

And all 4 curves were combined in one plot so that they could be compared.

The following plot was obtained:



The area under 4 curves is approximately the same, where highest performance is from Random Forest(area=0.9051) and the lowest performance is from k-NN (area=0.8830) meaning that regardless of probability threshold, overall performance of these 4 models is quite high.

(5) Discussion & Conclusion

Insights from data: Data is concentrated mainly for United States, it considers lots of factors, where capital gain and capital loss, education num and education are correlated. Also, majority of samples belong to <=50K classification output, meaning dataset is imbalanced. In majority cases, man being a husband can earn >50K annual income, while for woman, highest number of women earning >50K is wife, but a sufficient part of women can be not-in-family or unmarried and still earn >50K. Mostly, in our samples capital loss and capital gain is 0, while most frequently people work 40 hours per week. Also, the number of female and male who earn >50K is the smallest when their relationship status- own-child, which is quite reasonable. The most frequent education status is HS-grad.

Insights from different results among each model: lots of parameters in each of the model can affect accuracy and performance of the model. For instance changing from polynomial kernel to radial basis function kernel changed the performance of model. It is always important to consider imbalanced dataset, especially for Logistic Regression, where probability threshold determines the classification output, and problem of overfitting, as it is undesirable when we construct model. Also, it is important to consider accuracy on both training set to see how data fits the model and on testing set to see how model performs on new unseen dataset. Models can be evaluated and compared using cross validation or ROC-AUC curve. Moreover, as in models hyper parameters were set by me, it is important to find best set of parameters that will give the best performance of model that is why lots of factors, parameters and conditions needs to be considered when constructing models. For instance: changing number of neighbors for k-NN theoretical would decrease overfitting, although changing it to 150 gave me 1.00 accuracy on training set.

Regarding hypothesis, it is valid, and it was supported by tables and visualizations above. It was clearly shown that in United States, the number of men and women who have annual income >50K is the highest, when the relationship status of man - husband, and woman-wife. One of the improvement to my hypothesis would be to consider other factors that affect income, as not relationship status alone determines annual income. And it would be great to consider feature importance so that I would understand for how much does relationship status affect annual income, and it will help to decide on validness of my hypothesis.