Author – Disoj Neupane
Analysis and Prediction of the Fraudulent Firm

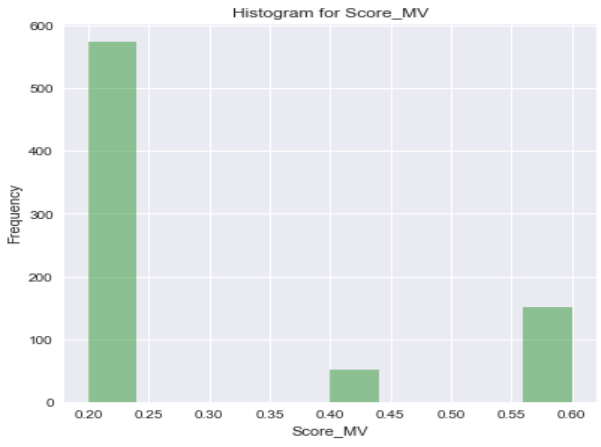
Analysis and prediction of fraudulent firm is very important these days because the examination of risk factors based on the historical data helps to reduce the risks of firms and enhance the profit and the reputation of the firms.

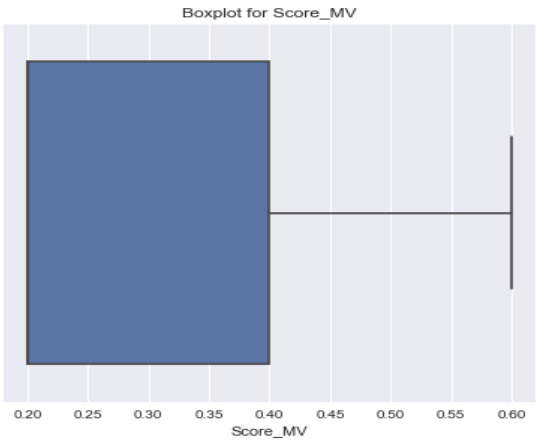
Exploratory Data Analysis:

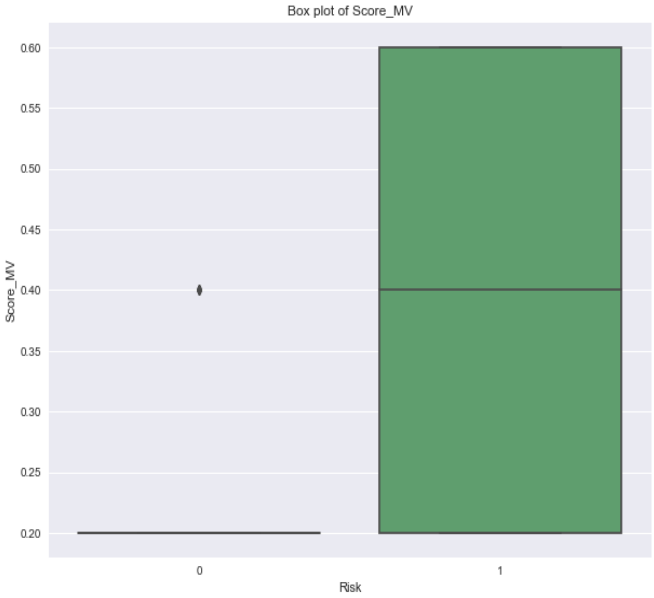
```
def boxplot_variables(var):

    """Function to plot histogram and boxplots"""

    #create 2 subplots for 2 columns
    fig,ax = plt.subplots(ncols=2, sharey= False, figsize = (14,6))

    #distribution plots
    sns.distplot(a = A_R[var], hist=True, kde=False, ax = ax[0], color = "green")
    ax[0].set_title("Histogram for {}".format(var))
    ax[0].set_ylabel("Frequency")

    #barplots
    sns.boxplot(x = var,data=A_R,ax=ax[1])
    ax[1].set_title("Boxplot for {}".format(var))

    plt.show()
```

Analysis based on Score_MV:

- Firms out of 776 having score_MV within the range of 0.20-0.40 are in safe.
- Firms above score_MV of 0.40 are in risk.

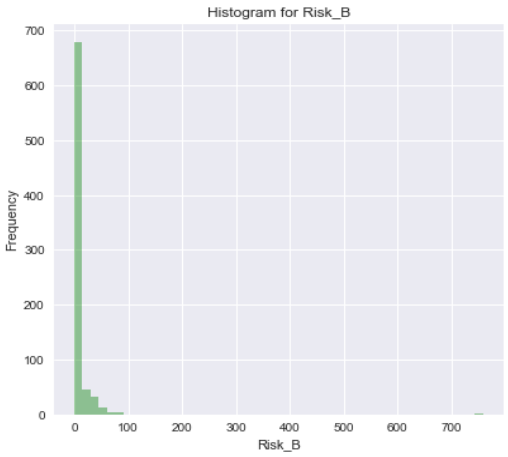


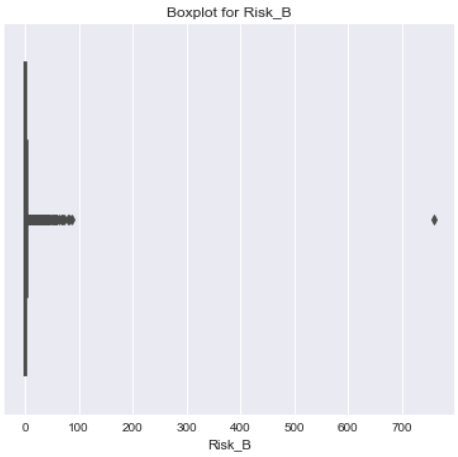


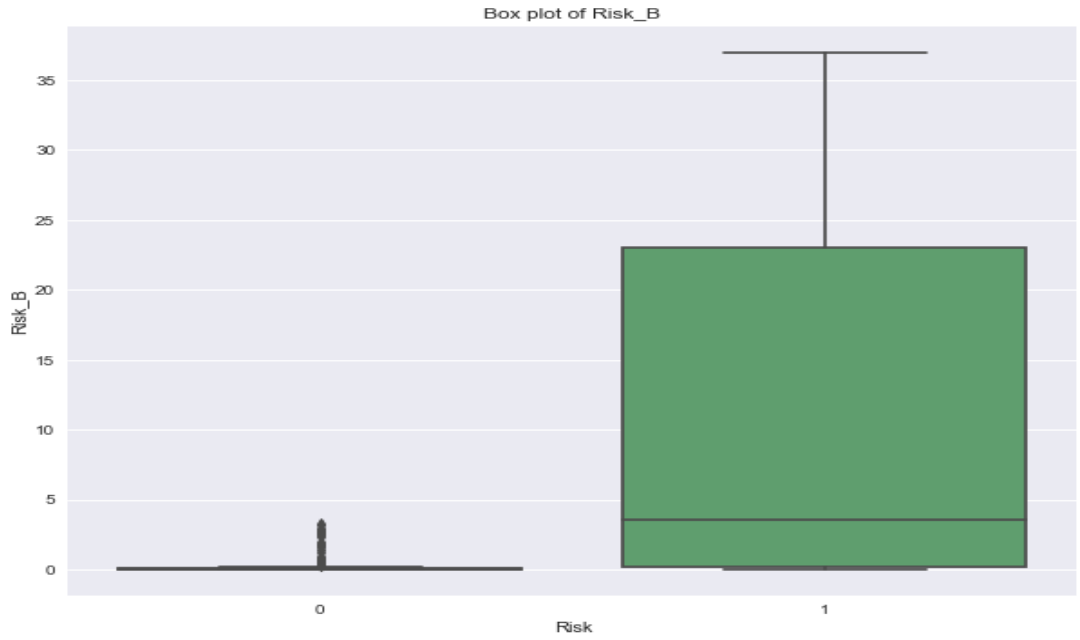


Analysis based on Risk_B:

Firms are at no risk with Risk_B factors within the range of 0-4, whereas firms with above this score are at risk.

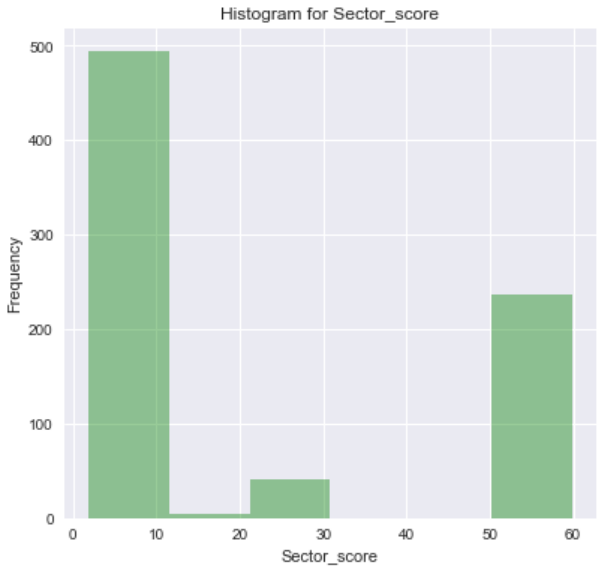


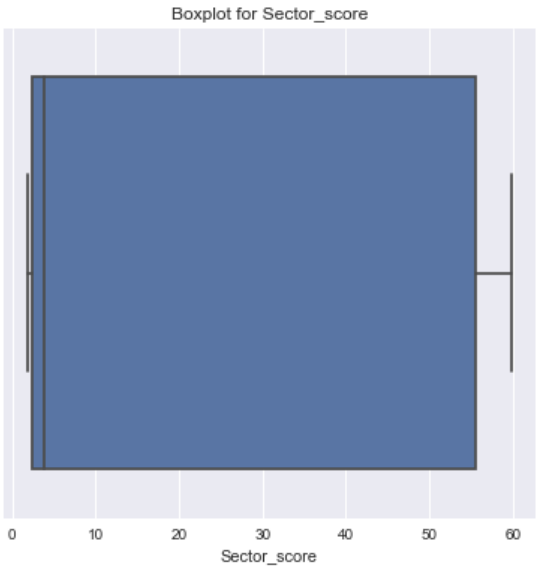


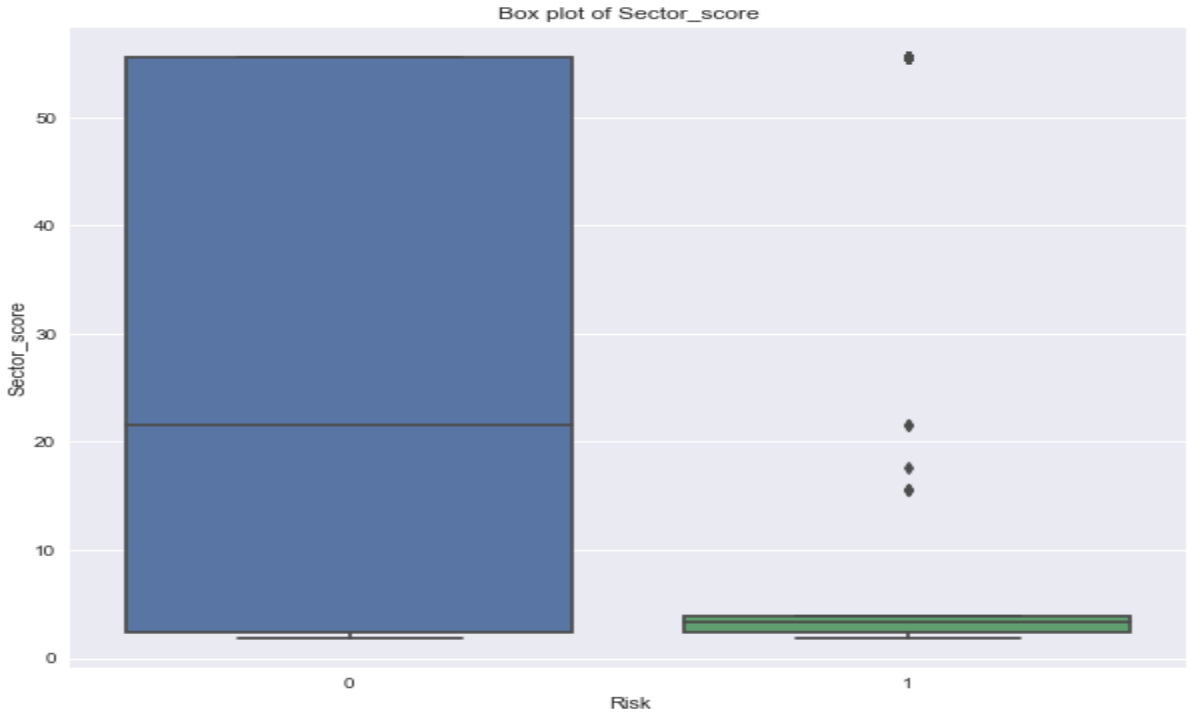


Analysis based on Sector_Score:

Majority of firms (almost 500) have low Sector_scores, and firms with Sector_score within the range of 2-22 and below are under risk.

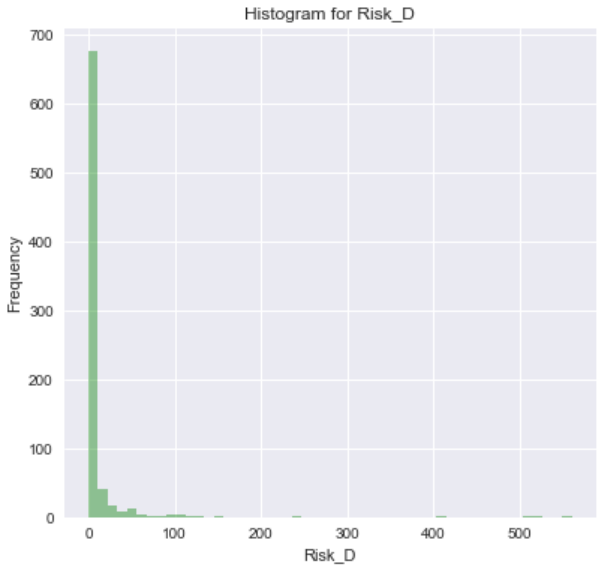


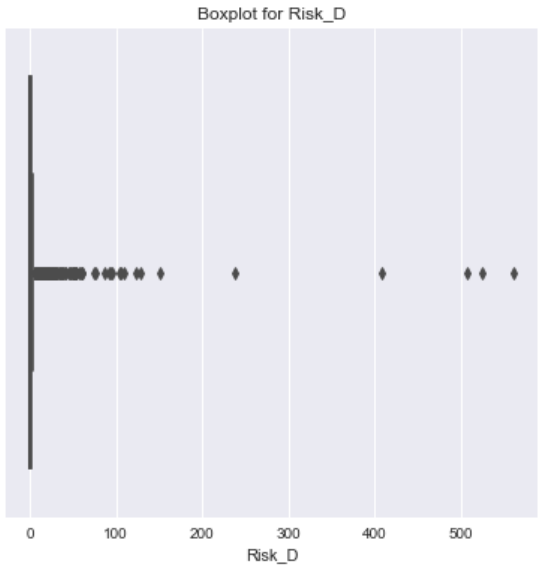


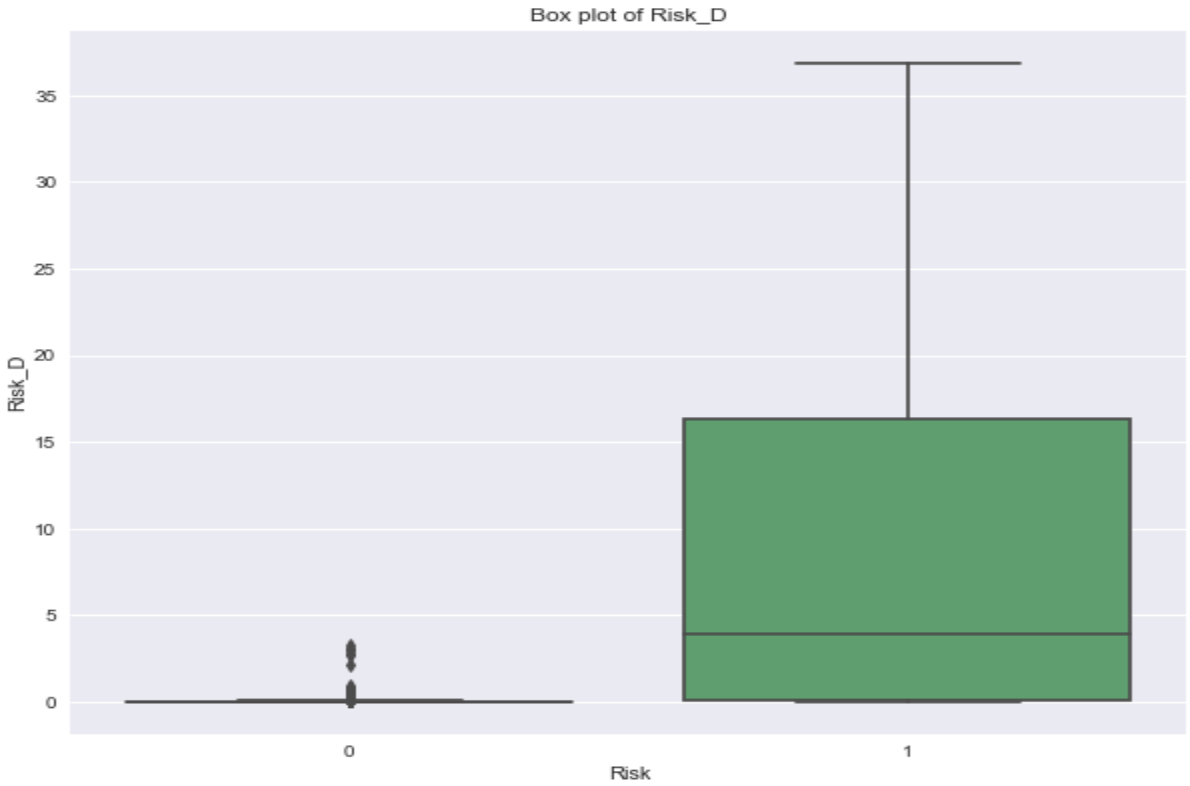


Analysis based on Risk_D:

Firms are at no risk with Risk_D factors within the range of 0-3.5, whereas firms with above this score are at risk. Majority of firms (almost 6800) are not affected by Risk_D factors.

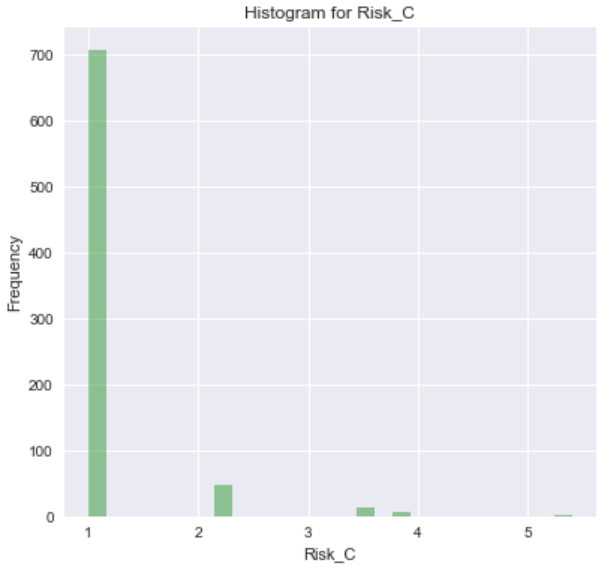


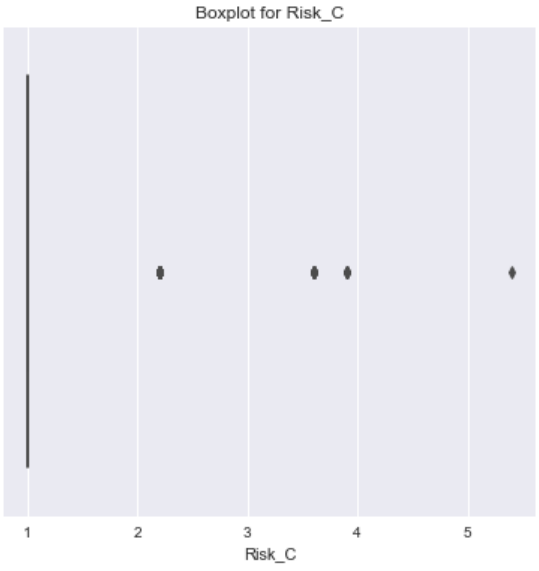


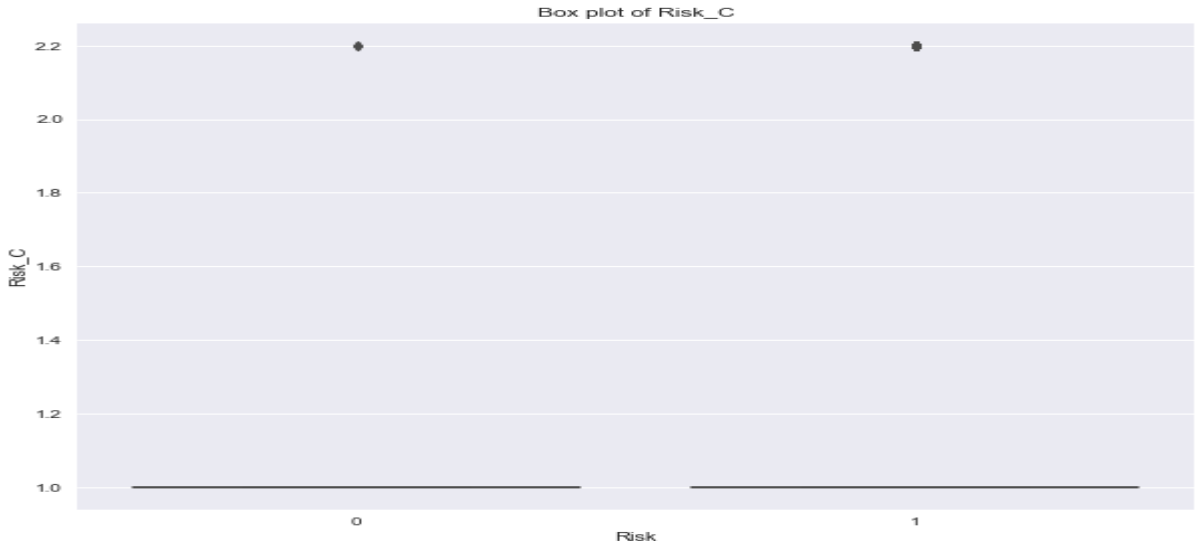


Analysis based on Risk_C:

Majority of firms are not affected by Risk_D factors.





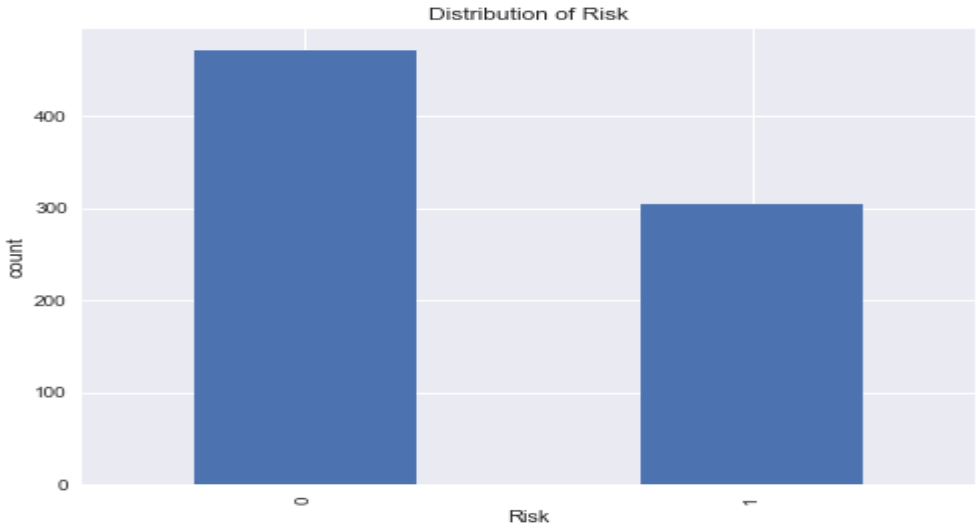


Similarly, most of the firms are having low values of prob, PROB and Audit_Risk, whereas all firms have the average value of the Detection_Risk. In addition, Majority of firms are lacking of history.

Risk Analysis:

Out of 776 firms, 305 firms are at high risk, where as 471 firms are risk free as shown in the following histogram.
A_R.Risk.value_counts()
0 471 1 305 Name: Risk, dtype: int64

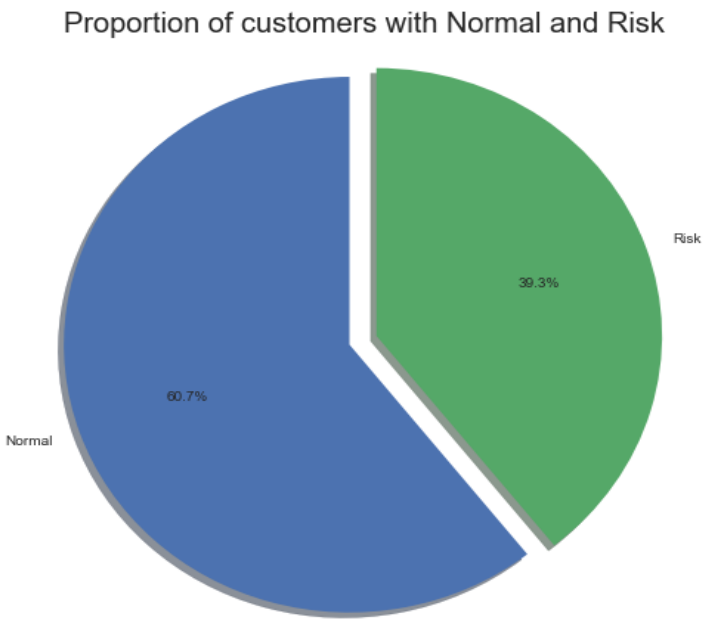
A_R.Risk.value_counts().plot.bar()
plt.xlabel("Risk")
plt.ylabel("count")
plt.title("Distribution of Risk")
plt.show()



Pie Chart:

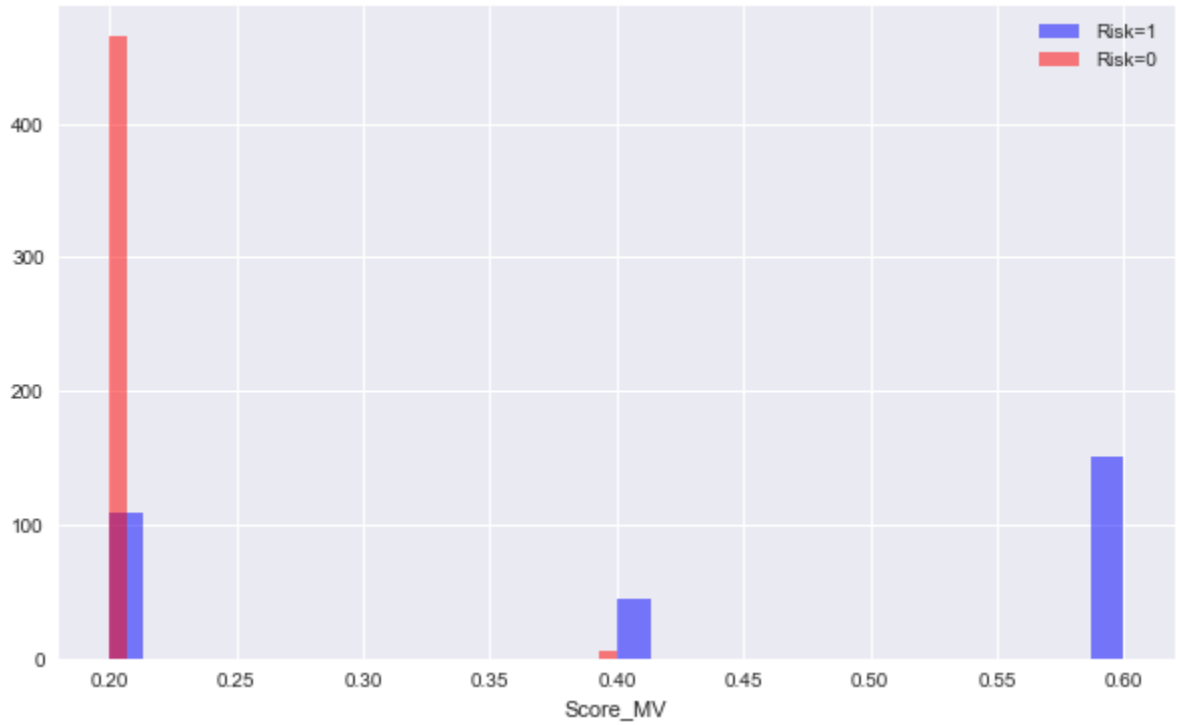
60.7 % firms are normal(risk free), whereas 39.3% are suffering from risk.

```
labels = 'Normal','Risk'
sizes = [A_R_sel.Risk[A_R_sel['Risk']==0].count(), A_R_sel.Risk[A_R_sel['Risk']==1].count()]
explode = (0, 0.1)
fig1, ax1 = plt.subplots(figsize=(10, 8))
ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')
plt.title("Proportion of customers with Normal and Risk", size = 20)
plt.show()
```



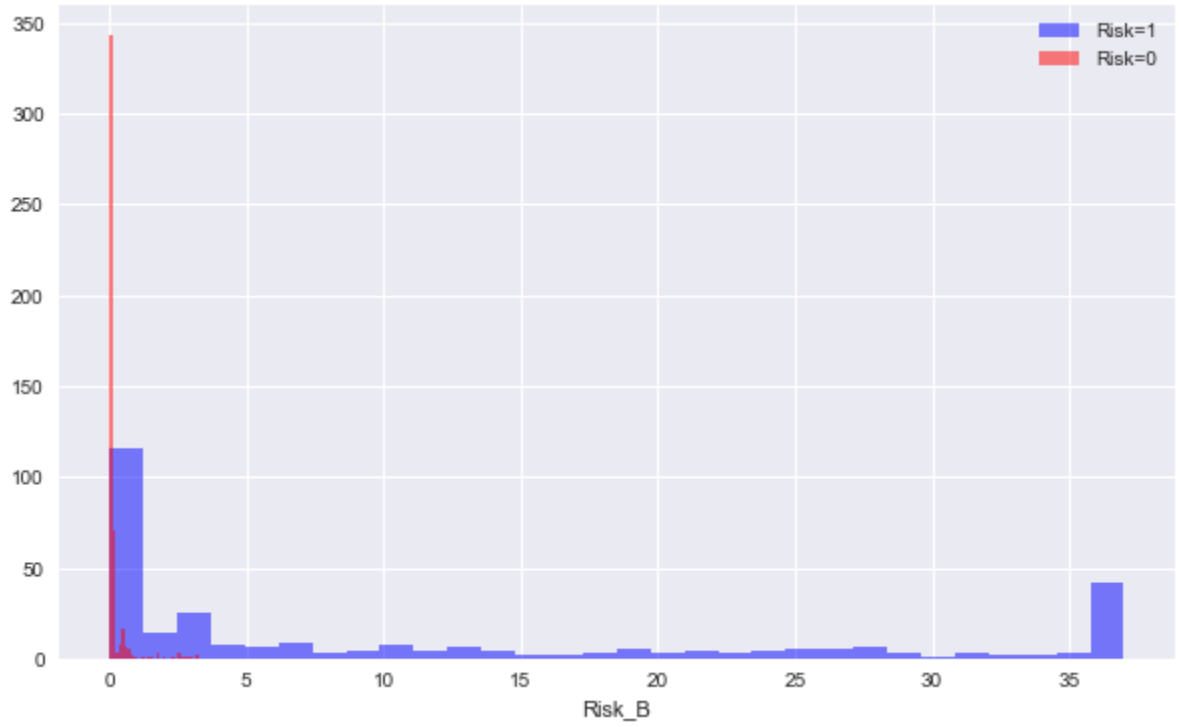
Histogram of two Score_MV distributions on top of each other, one for each Risk outcome:

```
plt.figure(figsize=(10,6))
A_R_sel[A_R_sel['Risk']==1]['Score_MV'].hist(alpha=0.5,color='blue',bins=30,label='Risk=1')
A_R_sel[A_R_sel['Risk']==0]['Score_MV'].hist(alpha=0.5,color='red',bins=30,label='Risk=0')
plt.legend()
plt.xlabel('Score_MV')
```



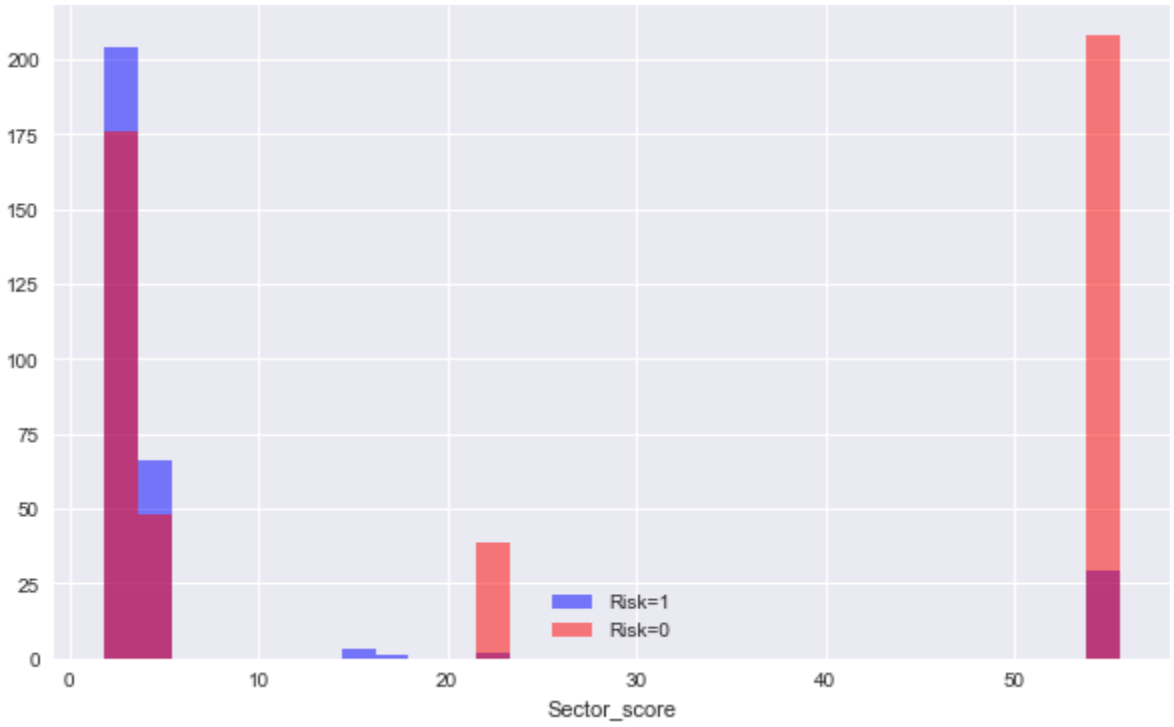
Histogram of two Risk_B distributions on top of each other, one for each Risk outcome

```
plt.figure(figsize=(10,6))
A_R_sel[A_R_sel['Risk']==1]['Risk_B'].hist(alpha=0.5,color='blue',bins=30,label='Risk=1')
A_R_sel[A_R_sel['Risk']==0]['Risk_B'].hist(alpha=0.5,color='red',bins=30,label='Risk=0')
plt.legend()
plt.xlabel('Risk_B')
```



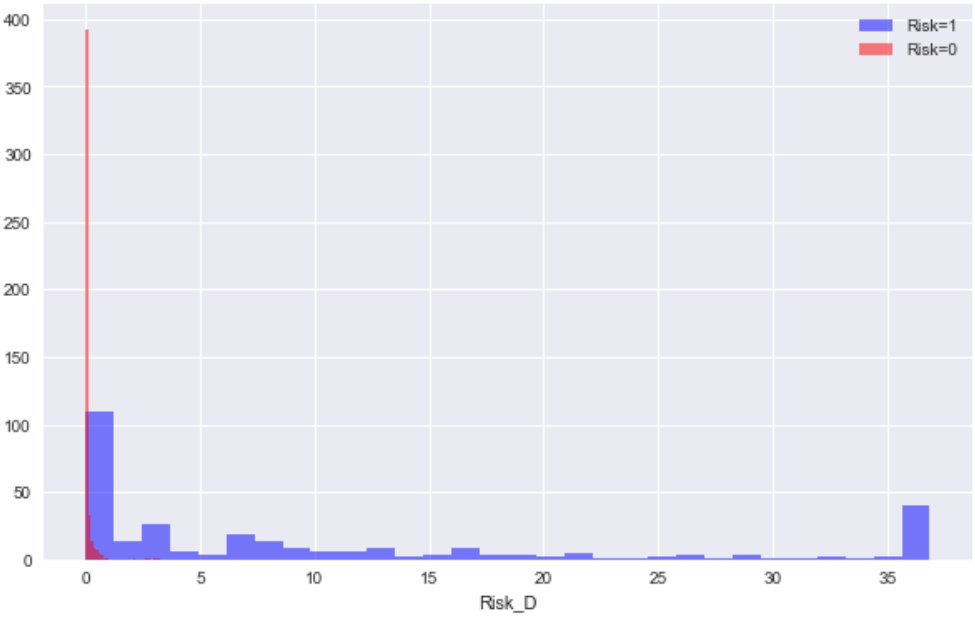
Histogram of two Sector_score distributions on top of each other, one for each Risk outcome

```
plt.figure(figsize=(10,6))
A_R_sel[A_R_sel['Risk']==1]['Sector_score'].hist(alpha=0.5,color='blue',bins=30,label='Risk=1'
)
A_R_sel[A_R_sel['Risk']==0]['Sector_score'].hist(alpha=0.5,color='red',bins=30,label='Risk=0')
plt.legend()
plt.xlabel('Sector_score')
```



Histogram of two Risk_D distributions on top of each other, one for each Risk outcome

```
plt.figure(figsize=(10,6))
A_R_sel[A_R_sel['Risk']==1]['Risk_D'].hist(alpha=0.5,color='blue',bins=30,label='Risk=1')
A_R_sel[A_R_sel['Risk']==0]['Risk_D'].hist(alpha=0.5,color='red',bins=30,label='Risk=0')
plt.legend()
plt.xlabel('Risk_D')
```

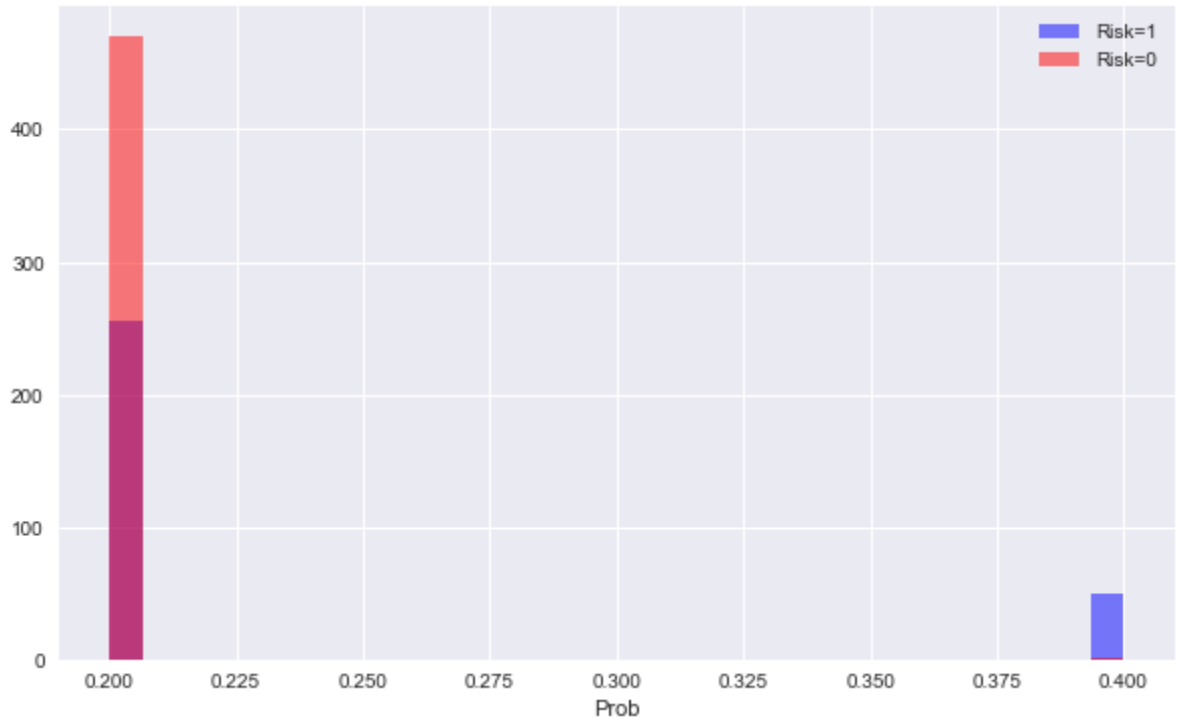


Histogram of two Risk_C distributions on top of each other, one for each Risk outcome

```
plt.figure(figsize=(10,6))

A_R_sel[A_R_sel['Risk']==1]['Risk_C'].hist(alpha=0.5,color='blue',bins=30,label='Risk=1')

A_R_sel[A_R_sel['Risk']==0]['Risk_C'].hist(alpha=0.5,color='red',bins=30,label='Risk=0')

plt.legend()

plt.xlabel('Risk_C')
```

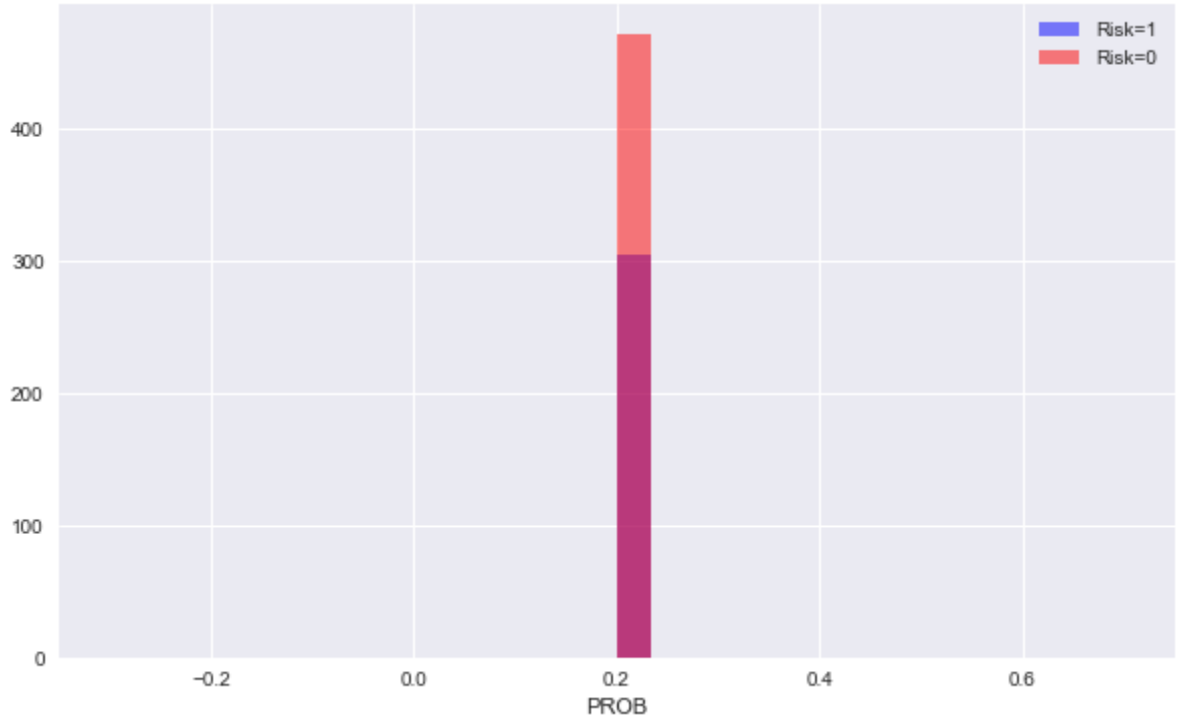
Histogram of two Prob distributions on top of each other, one for each Risk outcome

```
plt.figure(figsize=(10,6))
A_R_sel[A_R_sel['Risk']==1]['Prob'].hist(alpha=0.5,color='blue',bins=30,label='Risk=1')
A_R_sel[A_R_sel['Risk']==0]['Prob'].hist(alpha=0.5,color='red',bins=30,label='Risk=0')
plt.legend()
plt.xlabel('Prob')
```



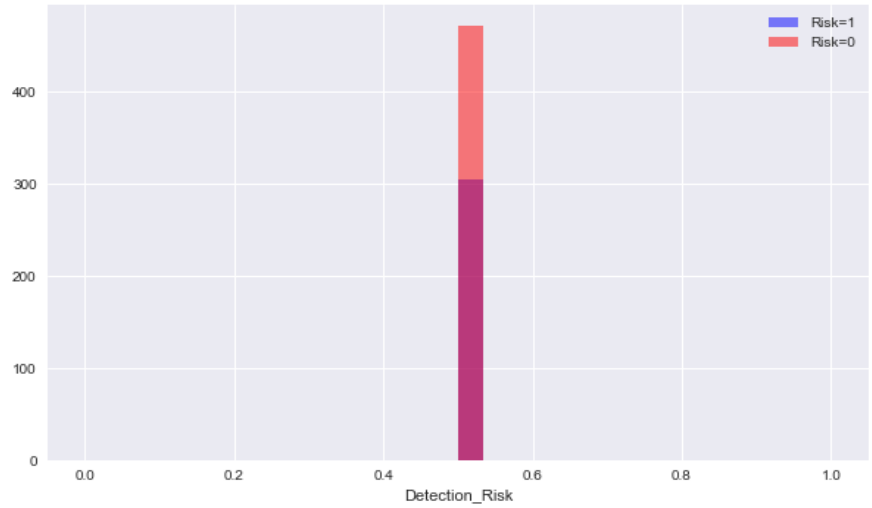
Histogram of two PROB distributions on top of each other, one for each Risk outcome

```
plt.figure(figsize=(10,6))
A_R_sel[A_R_sel['Risk']==1]['PROB'].hist(alpha=0.5,color='blue',bins=30,label='Risk=1')
A_R_sel[A_R_sel['Risk']==0]['PROB'].hist(alpha=0.5,color='red',bins=30,label='Risk=0')
plt.legend()
plt.xlabel('PROB')
```



Histogram of two Detection_Risk distributions on top of each other, one for each Risk outcome

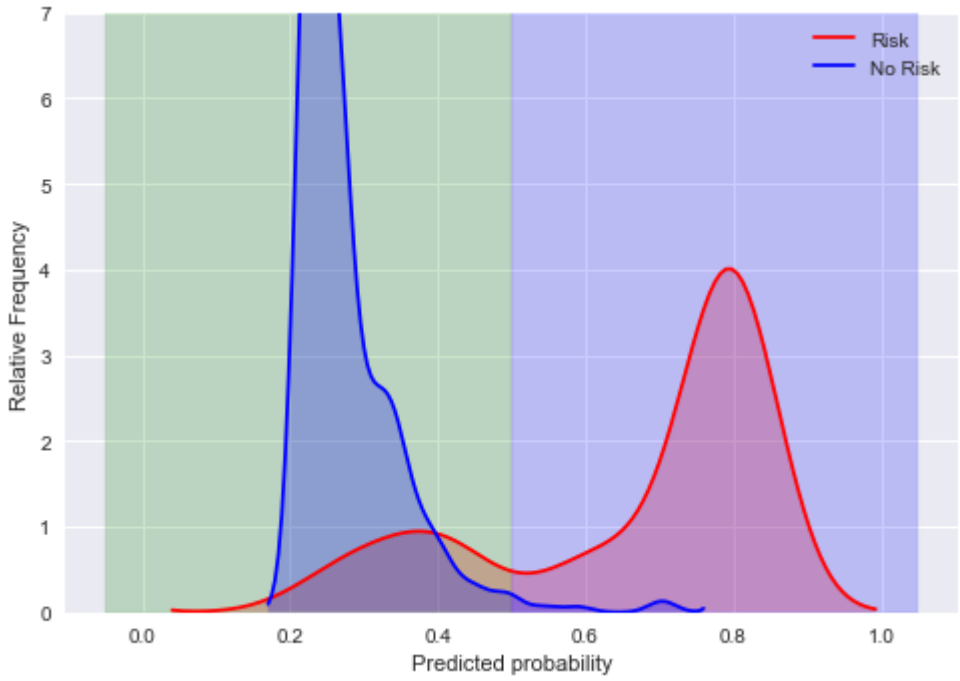
```
plt.figure(figsize=(10,6))

A_R_sel[A_R_sel['Risk']==1]['Detection_Risk'].hist(alpha=0.5,color='blue',bins=30,label='Risk=1')

A_R_sel[A_R_sel['Risk']==0]['Detection_Risk'].hist(alpha=0.5,color='red',bins=30,label='Risk=0')

plt.legend()

plt.xlabel('Detection_Risk')
```



Predictive Modeling:

```
y_pred_prob = best_model.predict_proba(train_X)[:,1]
ypp = pd.DataFrame(data = y_pred_prob, columns = ['predicted'])
ypp['Risk'] = train_Y.values

ypp1 = ypp[ypp.Risk == 1]
ypp0 = ypp[ypp.Risk == 0]
ax = sns.kdeplot(ypp1.predicted, shade=True, color="r", label = 'Risk')
plt.fill_between([-0.05,0.5], 0, 10, color = 'g', alpha = 0.2)
plt.fill_between([0.5,1.05], 0, 10, color = 'b', alpha = 0.2)
plt.ylim(0, 7)
sns.kdeplot(ypp0.predicted, shade=True, color="b", ax = ax, label = 'No Risk')
plt.xlabel('Predicted probability')
plt.ylabel('Relative Frequency')
plt.show()
```

I used supervised learning to build a model that predicts weather a firm is at risk or not based on the historical data. I tested several learning methods to achieve a better prediction. All models are performing well, but I predicted the risk analysis finally based on winning model, xgboost. The calculated probability distribution of a model xgboost classifier is shown below. The red colored distribution represents the data originally labeled as ‘Risk’. The ‘blue’ color represents the subset of data labeled as ‘No Risk’. My model performs fairly well in predicting the risk probability of most of the data.

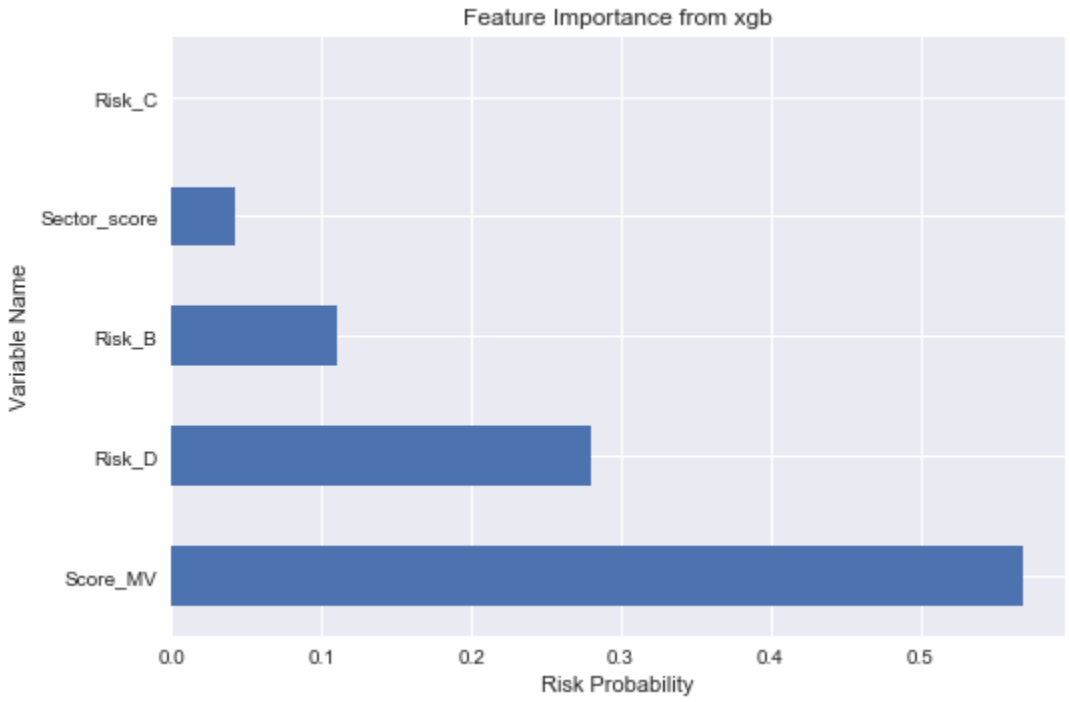


Calculated probability distribution of test data labeled as No Risk (blue) and Risk (red).

RECOMMENDATIONS:

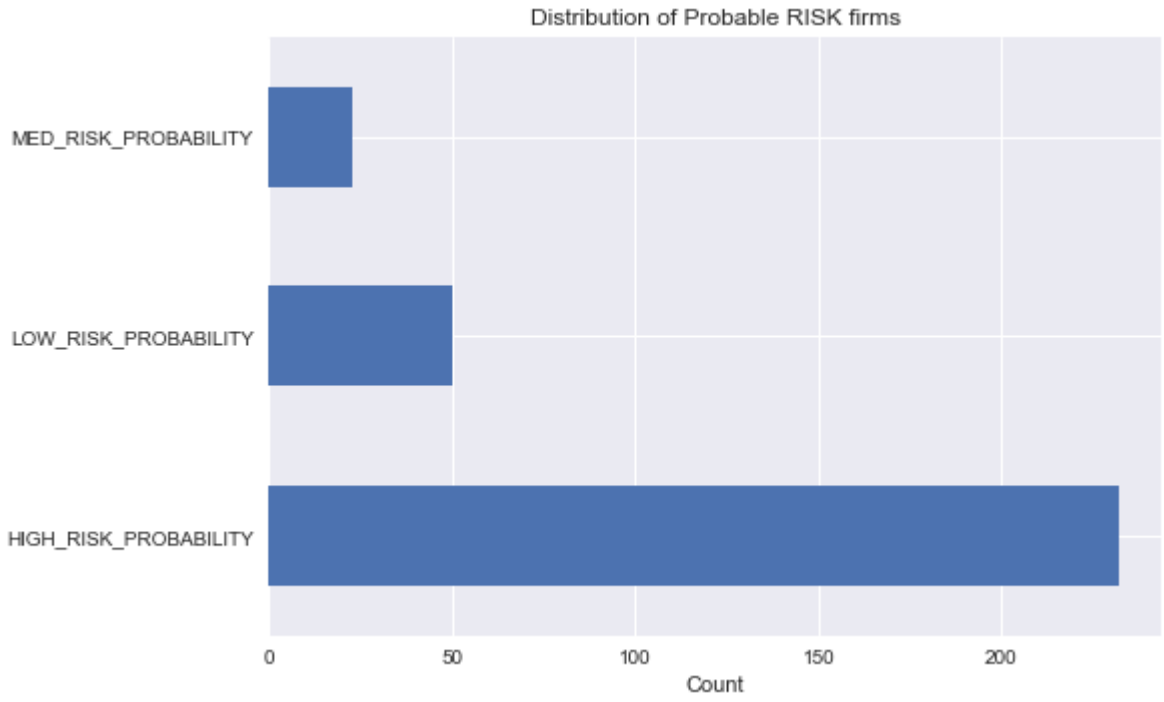
(A) Regarding to the risk analysis of firms, it is very important to understand the causes of the probability of high risk. In the figure below, I have shown the common factors which are responsible for the high risk of probability of firms. Among the selected factors, Score_MV is almost prominent factor to make the firms at high risk. Likewise, Risk_D and Risk_B are other two effective predictors, which are significantly correlated with the response variable 'Risk'. Therefore, reducing Score_MV, Risk_D, and Risk_B is extremely important in order to be safe from high risk for the firms.

```
feat_importances = pd.Series(best_model.feature_importances_, index=train_X.columns)
feat_importances.nlargest(5).plot(kind='barh')

plt.title("Feature Importance from xgb")
plt.xlabel("Risk Probability")
plt.ylabel("Variable Name")
plt.show()
```



Probability Distribution of Risks:

```
def risktype(prediction_of_actve_df):
   if ((prediction_of_actve_df.Prob_true  >= 0.0) & (prediction_of_actve_df.Prob_true < 0.4)):
      return 'LOW_RISK_PROBABILITY'
   elif ((prediction_of_actve_df.Prob_true  >= 0.4) & (prediction_of_actve_df.Prob_true <= 0.5)):
      return 'MED_RISK_PROBABILITY'
   else:
      return 'HIGH_RISK_PROBABILITY'


risk_active["RISK_BAND"] = risk_active.apply(risktype, axis=1)


risk_active.RISK_BAND.value_counts().plot(kind = "barh")
plt.title("Distribution of Probable RISK firms")

plt.xlabel("Count")
plt.show()
```



(B) For firms with High Risk probabilities will require an immediate call to understand their grievances/complains, and issues. May be a new relationship manager will be helpful to understand their issues in terms of payment, language and complaints. For firms with Medium Risk probabilities may require souvenirs for their time with the service provider. Keep in touch with call and do monitor their grievances and complains for better contribution.