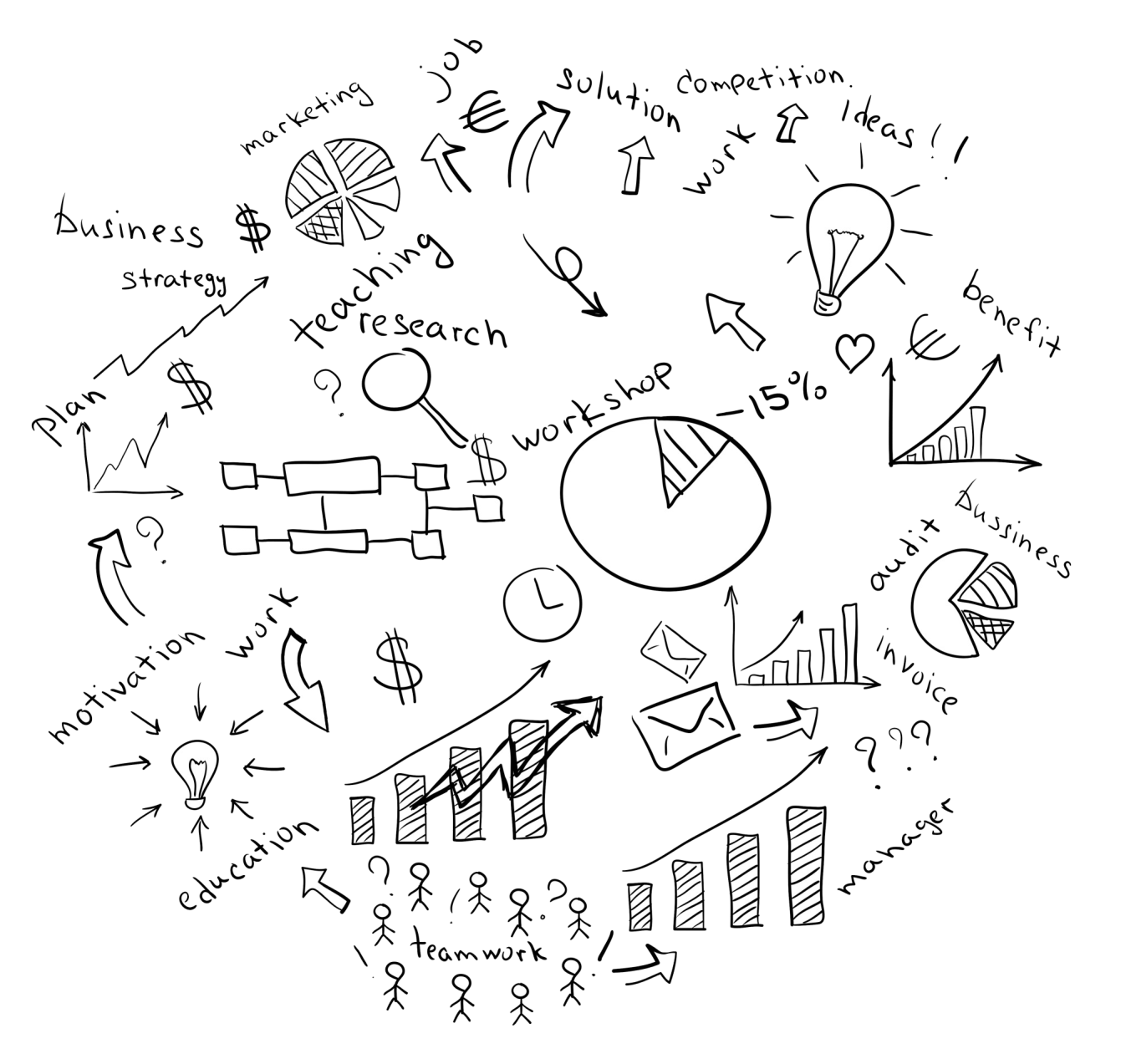
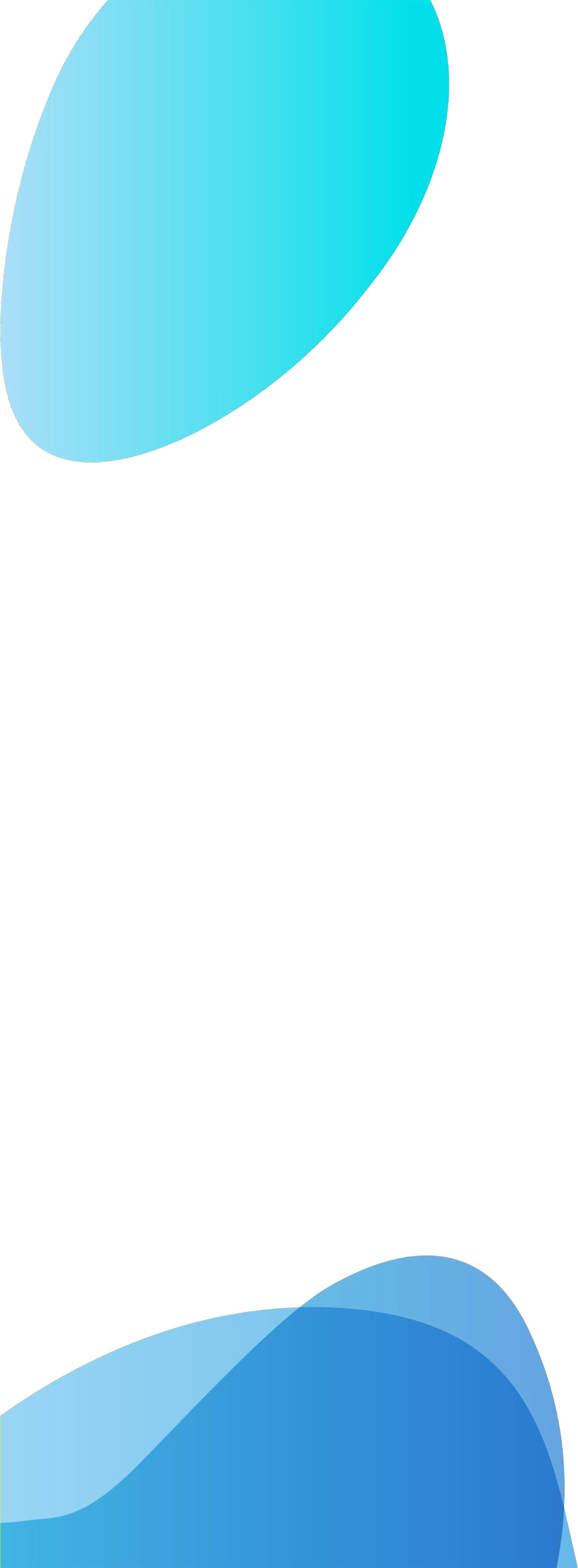
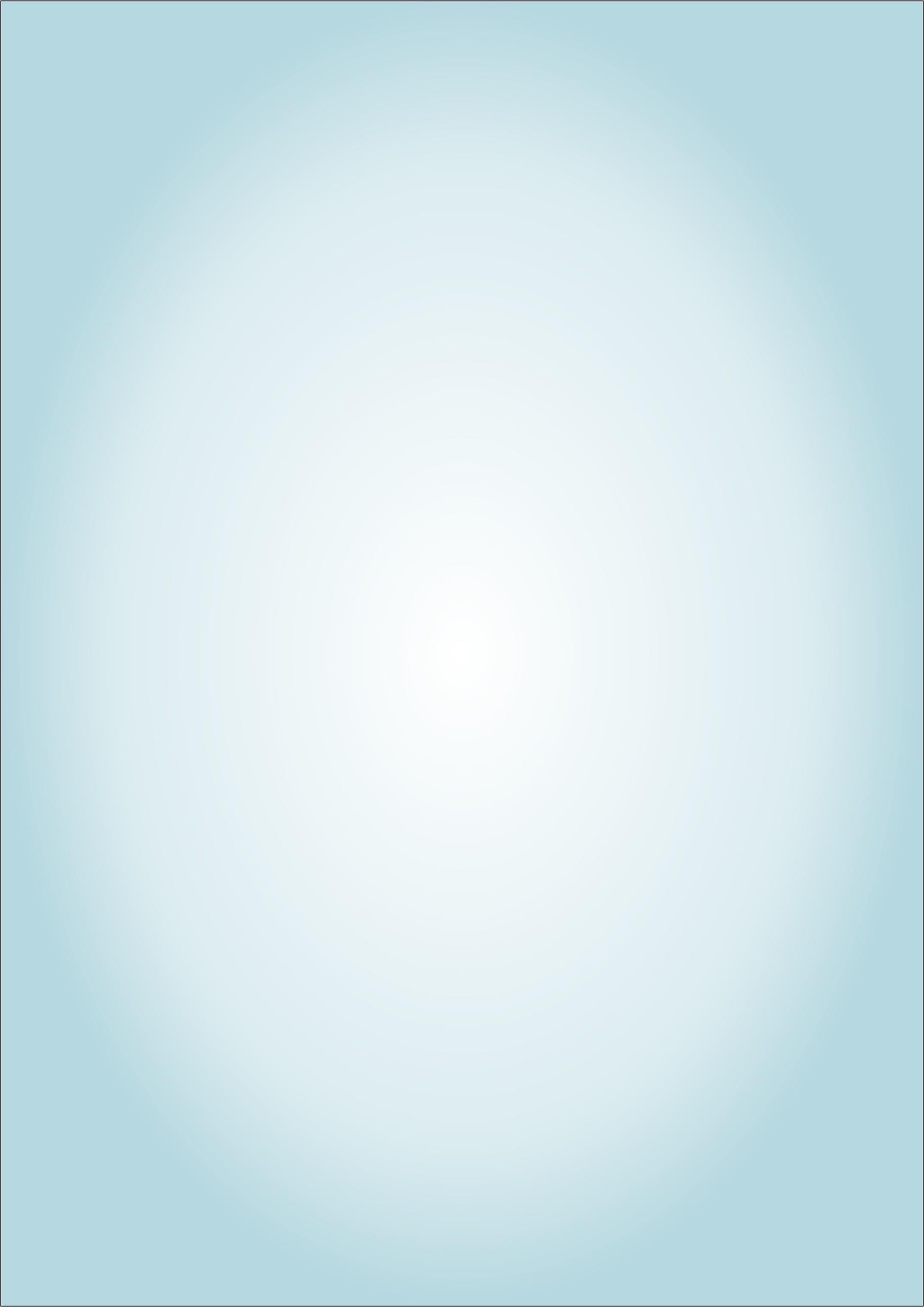
**IPL Winner Prediction**



ISM 6136 Data Mining Group Project

by

**Team 7 - Trail Blazers**

Abhishanth Penta

Lakshman Ganta

Shailendra Singh  
 Sri Priyanka Seelam

**Introduction**

The [**Indian Premier League** (**IPL**)](https://www.iplt20.com/) is a professional men's [**Twenty20 cricket**](https://en.wikipedia.org/wiki/Twenty20_cricket) league, contested by ten teams based out of ten Indian cities. The league was founded by the [**Board of Control for Cricket in India**](https://en.wikipedia.org/wiki/Board_of_Control_for_Cricket_in_India) (BCCI) in 2007. It is usually held between March and May of every year.

Cricket is a sport that captivates audiences and fans around the world. It is played on the international stage and is a global phenomenon. Different formats of the game are in existence today and the most fast paced and most watched format is the T20 format. 3-hour games with 40 overs per game makes it exhilarating to play and watch. After the international schedule concludes, domestic competitions take place and that is what gave birth to one of the most expensive and most watched leagues in the world, the Indian Premier League (IPL) in the year 2008.

The 2022 Indian Premier League (IPL) is set to generate I**₹1,000 crore (US$131,363,200)** in sponsorship revenue, Board of Control for Cricket in India (BCCI) secretary Jah Shah has confirmed.

**Problem Definition and Motivation**

The traditional approach in attempting to win the title of Indian Premier League, using machine learning techniques, is to use systems involving different variables. These systems include the following types of variables:

id, season, city, date, team\_1, team\_2, toss\_winner, toss\_decision, result, dl\_applied, winner, win\_by\_runs, win\_by\_wickets, player\_of\_match, venue, umpire1, umpire2, and umpire3.

Here the **Target Variable is “Winner”**, depending on the teams who played the match, the winner will be present in the winner column. Here the Winner is categorical with all the teams in the Winner data, but we will have only of the 2 teams which played that respective match.

Finding, compiling, maintaining, and updating this data is a massive task for the individual. Unless you have access to a database of such data - where would you even start?

The technology-centric Sports Consultant use their skill to study the players of each team and make a prediction of which team wins the tournament- that they think a particular player will play well at a particular situation in a match. We take those Consultants predictions and put them through a machine learning algorithm (Microsoft Azure) asking it to predict the winner of each match including the finals of the tournament based upon the team’s performance history.

**Data Description**

The data set for this classification problem comes from the Matches sample data set collection on Kaggle.

A general idea of the data can be gained by looking at the columns and their unique values.

* Id: Unique UID for each of the IPL of all Seasons.
* Season: The Edition of IPL in which the match happened.
* City: City where the match happened.
* Date: Date on which match happened.
* Team1, Team2: Teams which participated in the Match.
* Toss Winner: Team who won the Toss of the Match.
* Toss Decision: Team who won the toss have the benefit of selecting batting/bowling first of their choice.
* Result: Whether the Match ended normally, or it is a Tie (Which results in Super over).
* DL Applied: Method which is used to calculate the winner if the match stops (Mostly because of rain).
* Winner: Winner of the Match (Our Target Variable).
* Win by Runs: If the winner bats first, the team will win by Runs.
* Win by Wickets: if the winner bats second, the team will win by Wickets.
* Player of the Match: Award given to the best performer of the Match.
* Venue: Stadium where the match happened.
* Umpire1, Umpire2, Umpire3: The representatives who manages the flow of the Match.

**A picture containing table

Description automatically generated**

**Metrics for Model Evaluation**

Various measures are used to evaluate the performance of the chosen models:

• **Feature weights**: Indicates the model's key features for generating predictions.

• **Confusion matrix**: Displays a grid of true and false predictions versus actual values.

• **Accuracy score**: Indicates the model's overall accuracy for both the training and test sets.

• **ROC Curve**: Shows a model's diagnostic ability by combining true positives rate (TPR) and false positive rate (FPR) for various class prediction thresholds (For example, churn thresholds of 10%, 50%, or 90% result in a prediction of churn)

• **AUC (for ROC)**: Indicates the model's overall separability between classes associated with the ROC curve.

• **Precision-Recall-Curve**: Compares the false positive rate (FPR) and false negative rate (FNR) for different thresholds of class predictions to demonstrate diagnostic competence. It's good for data sets with a lot of class imbalances (negative values overrepresented), because it concentrates on accuracy and recall, which aren't affected by the quantity of genuine negatives, hence it eliminates the problem.

• **F1 Score**: Calculates the harmonic mean of precision and recall and so assesses the trade-off between the two.

• **AUC (for PRC)**: Indicates the model's overall separability between classes as measured by the Precision-Recall curve.

**Predictive Model**

**Diagram

Description automatically generated**

In the beginning we tested out models and measured their performance by several metrics. The models used include: Here, we did split of 70% to train the model and rest 30% to test.

**Two-Class Decision Forest:** Two-Class Decision Forest module to create a machine learning model based on the random decision forests algorithm. Decision forests are fast, supervised ensemble models. This module can be used to predict a target that has two values.

**Neural Network:** Despite the fact that the data set is minimal and that neural networks typically require a large amount of training data to have useful prediction capabilities, a rudimentary neural network is used to compare the two approaches.

**Support Vector Mechanism:** Support vector machines (SVMs) are a well-researched class of supervised learning methods. This implementation is suited to prediction of two possible outcomes, based on either continuous or categorical variables.

**Experiments**

1. **Two-Class Decision Forest:**

**Chart, line chart

Description automatically generated** **Chart

Description automatically generated**

**Graphical user interface, application

Description automatically generated with medium confidence**

1. **Two-Class Neural Network:**

**Chart, line chart

Description automatically generated Chart

Description automatically generated**

**Graphical user interface, text, application, Word

Description automatically generated**

1. **Two-Class Support Vector Machine:**

**Chart, line chart

Description automatically generated Chart, line chart

Description automatically generated**

**Table

Description automatically generated with medium confidence**

**Predictive Model with Factors affecting Team’s Performance.**

**Diagram

Description automatically generated**

**Comparison of Model with all independent variables and the model with no Details regarding Venue.**

**Chart, line chart

Description automatically generated** **Table

Description automatically generated**

**Comparison of Model with all independent variables and the model with no Details regarding Toss.**

**Chart, line chart

Description automatically generated**

**Table

Description automatically generated**

**Comparing Models**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Re Call** | **F1 Score** | **AUC** |
| Two-Class Decision Forest | **0.995** | **0.994** | **1.000** | **0.997** | **1.000** |
| Two-Class Neural Network | **0.956** | **0.974** | **0.980** | **0.977** | **0.971** |
| Two-Class SVM | **0.941** | **0.941** | **1.000** | **0.970** | **0.701** |
| Models with No Venue | **0.956** | **0.974** | **0.980** | **0.977** | **0.971** |
| Models with no Toss | **0.941** | **0.941** | **1.000** | **0.970** | **0.701** |

**Conclusion**

Based on the above findings, the Two Class Decision Forest model has the best accuracy (0.995) on the test set. i.e., with all the independent variables, the accuracy is more when compared to other models including comparison of default model with the model which exclude Venue details and the other model which exclude Toss Details.

This is because of the correlation between the Venue and the Toss, few venues have advantage of bating first and few others have advantage of fielding first. This is affecting the models and the model with all these independent variables is performing better when compared to other models which exclude these details and details related to toss.

**Recommendations**

* Franchise can optimize the team and bid in auction based on the player’s previous performance and recent form for winning the present season of IPL.
* Franchise can estimate the price range of a particular player based on the analysis.
* Stats of the venues and pitch condition can help the teams to decide on how to start the game (whether to bat first or field first after winning the toss).
* This analysis helps the team management to plot things against a particular player or a particular player by knowing their strengths and weaknesses.
* It helps Team Management to give rest to key players when playing against weak team.

**Way Forward:** With the help of the above analysis, we can predict the winner of any game or any tournament between any number of teams with high efficiency.