Capstone Details for Batch 1

We have three tasks for the captsone.

1. IPL Analytics: Exploratory Data Analytics and Win Predictor
2. Stock price data analysis and price prediction
3. Case Study: Learning to see in the Dark

1. IPL Analytics

Ball-by-Ball data of 637 IPL matches are given (through 2017 season).
Data source is credited to http://cricsheet.org/

There are 5 different files (column indices are given below):

1. ball_by_ball.csv: contains detailed ball by ball information
2. matches.csv: list of matches played, teams, toss winner, match winner, winning margin
3. player_match_list: Information between whether a player played a match. If yes, role details such as captain/ wicket-keeper/ man-of-match award and mapping between match identifier and player identifier
4. teams.csv : Mapping between team name and identifiers used in the dataset
5. players.csv: Player's name/ description and mapping to the identifiers used in the dataset

Exploratory data analysis

a. Venue-wise analysis
The average 1st innings and 2nd innings score across all venues
b. Effect of Toss and home-field advantage on win
c. Clustering of matches: Based on balls used/ wickets lost/ runs scored by the two teams, can you categorize matches?
d. Find the top-5 run-scorers and wicket-takers across all IPL seasons
e. Histogram of dismisal-types for Virat Kohli

Win-predictor network

You will train a win-predictor network for the team batting second. The input variables are first team's score, second team's current score / wickets remaining / balls remaining. The win-predictor network will be used only when atleast 30 balls have been played by the second team.

For training, do not consider matches that were washed out. For simplicity you need not consider data from matches that end with a super-over.

Suggestion for validating your network

Sample about 10 matches and plot the ball-by-ball probability of winning of the second team from the start of the 6th over. Superpose on this plot the wickets/ boundary events.

1. Verify there is a dip / rise in win probability corresponding to the events.
2. Verify that the probability has tailed off to 0% or plateaued to 100% depending on the final outcome.

Evaluation Criteria

Please provide

- A python notebook, or python scripts (with instructions on how to run)
- A report containing:
  1. Description of the solution/network, including your rationale for choices that you made.
  2. Results obtained.
  3. Observations/comments such as what improvements were tried, what worked, what did not work, such as hyperparameter tuning etc..

Submit your project as a zip file containing appropriate folder structures as given below:

ZIP file structure:

<teamIPL.zip>

IPL (name of the folder):

- ---- report.pdf
- ---- A/
- ---- A/solution(s).ipynb
- ---- A/extra.csv (if used)

If your code requires any special files, include them in your submission in the correct folder.

We will use your win-predictor network to forecast winners in the ongoing IPL season. So please submit the trained model weights.

Enhanced Win-predictor network

Are there one or more attributes you can add to improve the accuracy?

Feel free to add your own. Here are some suggestions:
- Venue
- Each team's form (for eg, last 3 matches results can be coded as WLW, LLL etc)
- Toss (captain winning the toss may have opted to bat first for reasons like dew)
- Top 3 batsmen/ bowler form

2. Stock price data analysis and price prediction

Given are 3 datasets:
Daily stock price movement(price_movement),
Revenue(revenue) &
News item(news_articles) for 10 companies.

(Data source : https://www.ischool.berkeley.edu/projects/2017/ml-based-investment-analytical-tool)
Period for which each dataset is available is different.

Exploratory Data Analysis

1. Find the top 3 performing stocks, report any trends/seasonality in the price movement.
2. Explore different ways of extracting feature information from news articles such as sentiment, presence of keywords that might influence the price movement etc.. consider using nltk/textblob etc.
3. For the top 3 performing stocks, observe if there is any correlation between the sentiment conveyed in the news article on a particular day and their price movement over the next 5 days for which the data is available.
4. Report any other observations you come across.

Forecasting

Use the information from the three datasets and build a LSTM model for predicting the stock price on the 30th day.

1. As the dates for which the 3 datasets are available is different, explore & report different ways of dealing with missing entries.You have to match the news articles with the financial data based on company codes from *newsdata.txt*

2. (i) Use PACF to find the optimum number of lag days to consider for determining the dependency of the 30th day stock price on previous days price, revenue and features extracted from *newsitem* and build a time series data set based on this to be fed to the LSTM model.

(ii) Does taking the *newsarticle* information to predict the stock price result in better prediction than using only financial data?

(iii) Build a forecasting model using only the financial data for years 2008 -2010 (inclusive of both) & check if the prediction confirms to the recovery trend observed in that period.

3. Use any word embeddings (glove is provided in shared dataset folder) to represent the news data along with the financial data & check if dimensionality reduction techniques give better results.

Note: Revenue data is reported on quarterly basis whereas the stock price data and news articles are reported only day basis you can consider the propagating the data for all days until the next quarter data is available.

Evaluation Criteria

Please provide

- A python notebook, or python scripts (with instructions on how to run)
- A report containing:
    1. Description of the solution/network, including your rationale for choices you made.
    2. Results obtained.
    3. Observations/comments such as what improvements were tried, what worked, what did not work, such as hyper-parameter tuning etc..

Submit your project as a zip file containing appropriate folder structures such as given below:

ZIP file structure:
<teamLSTM.zip>

Stock price (name of the folder):

- ---- report.pdf
- ---- A/
- ---- A/solution(s).ipynb
- ---- A/extra.csv (if used)

If your code requires any special files, include them in your submission in the correct folder. Do not include weights for pre-built word vectors/embeddings such as Glove, word2vec.

Solution notebook(s) will be used to predict the stocks for the given companies for a set of arbitrarily chosen dates in the duration considered.

3. Learning to see in the Dark

We will look at a neural network that brightens the images taken under short exposure. Please clone the repository given in https://github.com/cchen156/Learning-to-See-in-the-Dark.

There are pre-trained models available for both Sony and Fuji cameras. We will use only Fuji images.

1. Download the weights for the Fuji model.
2. Run the pre-trained model on the dataset provided in input_gt.tar.gz. The zipped file has input images along with ground truth images. This what test_Fuji.py expects. Check the results_Fuji folder to see the output images in png format.
3. The main task is to enhance the five test images given in test_images.tar.gz. Firstly, copy the images to the directory: dataset/Fuji/short. Delete all the files in dataset/Fuji/long/. Modify the code in test_Fuji.py so that the output images of the test set are written to results_Fuji/final.

To verify the quality of the output, compare the png image produced by the model for the image 10001_heart.RAF to the one given in the paper. The output must resemble the image given in fig. 5(c).